WIBOX - A robust video receiver allowing WIMAX video broadcasting and indoor WIFI rebroadcast

Usman Ali

THÈSE DE DOCTORAT

SPECIALITE : Informatique et Réseaux

Ecole Doctorale « l'école doctorale informatique, télécommunications et électronique»

WIBOX – Une passerelle pour une réception robuste de vidéo diffusée via WIMAX et une rediffusion indoor via WIFI

Présentée par : Usman Ali

Soutenue le 9 Novembre 2010 devant les membres du jury :

Prof. Marco Chiani	WCL/University of Bologna, Italy	Rapporteur
Prof. Jean-Marie Gorce	CITI/INSA-Lyon, France	Rapporteur
Prof. Béatrice Pesquet-Popescu	TSI/Télécom, ParisTech, France	Examinatrice
Prof. Pierre Duhamel	LSS/Supelec, France	Directeur de thèse
Prof. Michel Kieffer	LSS/Supelec, France	Directeur de thèse

Résumé Long

La plupart des outils robustes, comme les décodeurs vidéo utilisant des techniques Décodage Cannel-Source Conjoint (DCSC), ne sont pas compatibles avec le Protocole de Pile Standard (PPS) en raison de plusieurs conditions non-conformes:

1. Le PPS empêche corrompu paquets pour atteindre la couche APL, par exemple, l'actuel protocole WiFi gouttes tous les paquets erronés et WiMAX gouttes tous les paquets avec des en-têtes erronés. Dans le PPS de la technologie WiMAX et les normes Wi-Fi, la politique de contrôle retransmission est utilisé pour réduire la perte de paquets de bout en bout.
2. Ils exigent l'échange d'information souple entre les couches protocolaires. Le PPS du WiMAX et WiFi ne fonctionne pas avec l'information soft et d'entraver l'échange d'informations entre les mous du décodeur de canal à la couche PHY et le décodeur source fiable à la couche Application (APL).
3. L'utilisation classique du mécanisme de retransmission dans le WiMAX et WiFi, où paquet erroné est tombé et est retransmis, n'est pas compatible avec ces décodeurs robuste.

Les solutions des exigences ci-dessus sont normalement apaiser par les modifications suivantes à l'Accord PPS:

1. La pile de protocole est généralement rendue perméable aux erreurs de transmission en le rendant capable de transmettre des paquets erronés à l'APL couche [1]. Ainsi, l'objectif principal des couches inférieures est de relayer nombre maximum de paquets à la couche d'APL. Ce problème de la PPS en développement est partiellement abordée pour la couche UDP [2; 3] et pour la couche MAC dans [4; 5; 6], où les idées de détection des erreurs sélectifs sont présentés. L'adoption de la détection des erreurs sélective, où le paquet de rejet est d'éviter autant que les erreurs ne portent pas atteinte bits important (par exemple, en-tête) d'un paquet, peut produire plusieurs effets positifs sur le réseau. Néanmoins, le paquet sera en baisse en raison de têtes erronée peut encore devenir importante, en particulier à des taux élevés date [7] ou dans des situations où le ratio charge utile-à-tête est faible. Cette lacune peut être adressée en partie par le récepteur régimes fondés qu'en plus d'ignorer les erreurs de charge utile peut estimer les champs d'en-tête corrompue [8; 9; 10; 11], donc même les paquets avec des en-têtes erronés peuvent être transmis aux couches supérieures.

2. Les informations doux à la couche PHY [12] peuvent atteindre la couche d'APL, par exemple, en utilisant les mécanismes couche transparente proposée dans [13; 14], et peut donc être utilisé par le décodeur vidéo robuste. Cette modification entraîne des changements dans le récepteur, et n'est donc pas une solution applicable: il est compatible avec le signal qui est transmis.
3. Un paquet reçu par erreur ne doit pas être retransmis sans le décodeur vidéo robuste à couche APL ne peut pas récupérer l'erreur.

Cependant, ces solutions pourraient être difficiles à utiliser dans plusieurs circonstances. Les deux complications importantes, on peut faire face tout en développant un récepteur vidéo robuste, sont étudiés dans cette thèse et ne sont mentionnés ci-dessous.

Tout d'abord, souvent à une couche donnée de la PPS de la technologie WiMAX et les normes Wi-Fi, les petits paquets sont regroupés en paquets plus volumineux ou des éclats, qui sont ensuite transmis aux couches inférieures du protocole à l'émetteur. Cette agrégation de paquets est utile dans les situations où chaque unité de transport peut avoir des frais généraux importants (préambules, en-têtes, etc) ou lorsque la taille de paquet attendu est faible par rapport à la quantité maximale d'informations pouvant être transmises. Ainsi, il améliore le ratio charge utile-à-tête et augmente le débit de ces réseaux sans fil. De même, tel que proposé dans [15; 16], le multiplexeur peut combiner plusieurs paquets en un seul paquet multiplexés et le point d'accès multidiffusions le paquet multiplexés à canal sans fil de réduire la capacité des frais généraux et augmenter. Toutefois, lorsque la transmission sur un canal bruité est considéré, FS, c'est à dire, récupérer les paquets agrégées à partir d'un éclatement ou un paquet multiplexés, peut devenir difficile.

Au niveau du récepteur, sans FS robustes, des techniques d'estimation en-tête mentionnés dans la solution (1) ci-dessus demeurent infructueuses, quant à eux trouver les en-têtes et de leurs limites est obligatoire. En outre, les échecs FS à un protocole de couche inférieure peut entraîner la perte de plusieurs paquets consécutifs, ce qui pourrait autrement être transmise à la couche d'APL pour DCSC. Bien que, en cas, par exemple, la dégradation de WiMAX-lien, l'utilisateur peut demander le changement de modulation et de codage à une plus robuste, il peut ne pas être possible dans le scénario de diffusion. En outre, certains des FS sur l'état de l'art des techniques [17; 18] ne sont pas conformes à la solution (2), car ils travaillent sur des bouts durs. Bien que, plusieurs techniques de FS, généralement déployés à la couche PHY, peuvent travailler sur la sortie souple du canal [19; 20; 21; 22; 23], ils ont besoin d'insertion des marqueurs de synchronisation (les codes de démarrage), qui nécessitent des modifications l'émetteur et une charge supplémentaire, ne peut donc pas être autorisée dans les piles de protocoles des normes WiMAX/WiFi.

Deuxièmement, pour améliorer la performance, ces réseaux sans fil, en particulier WiFi, ont tendance à inclure un FEC algorithme [24] pour éviter les retransmissions de couche de liaison, dans les situations où les paquets consécutifs sont susceptibles d'être infectées par erreur en rafale, et dans la diffusion des applications où retransmission est très difficile, sinon interdit. Ainsi, il faut fournir une protection supplémentaire pour les paquets, afin qu'un paquet perdu peut être récupéré à la fin du récepteur. Au niveau des paquets FEC décodage pour récupérer un paquet perdu est souvent pratiquée sur les bits dur, ce qui nécessite la perte de l'information soft de la soft-décodeurs de sortie fonctionne à la couche PHY. Par conséquent, il entrave la circulation de l'information doux pour les couches supérieures, ce qui empêche l'utilisation de la solution (2). Par exemple, le paquet-niveau du protocole FEC RFC 5109 [25] décrit la méthode pour récupérer un paquet à partir des paquets non corrompue dur. En outre, généralement au niveau des paquets FEC décodage nécessite paquets sans erreur pour bien récupérer les paquets perdus ou abandonnés [26],

ce qui rend inutile DCSC en ne permettant pas les paquets erronés pour atteindre la couche APL, donc en conflit avec la solution (1).

L'induction de contention systèmes d'accès multiple dans IEEE 802.11e [27] et le développement des techniques robustes de récupération DCSC à différentes couches de protocole ont presque éliminé les pertes de paquets dans le canal et dans les couches de protocole, respectivement. Par conséquent, au lieu de la solution (3), le développement d'une technique de décodage, capable d'effectuer le décodage FEC sur le soft-évalués paquets partiellement endommagé, est la nécessité de l'époque. Ainsi, pour les services de radiodiffusion de la retransmission peut être complètement désactivé et remplacé par le paquet au niveau des FEC.

Propositions

Cette thèse étudie un certain nombre d'outils (rassemblés dans la WiBox) nécessaires pour une réception fiable de vidéo diffusée sur WiMAX, puis rediffusée sur Wifi. Il s'agit de fournir des services WiMAX à des utilisateurs WiFi, avec une qualité de réception vidéo raisonnable, même avec un très faible signal WiMAX. Pour cela, des techniques de décodage conjoint de paquets erronés sont indispensables afin de limiter les retards liés aux retransmissions.

Cette thèse se propose d'étudier ce qui précède deux circonstances d'élaborer des propositions pour le Protocole de Pile Soft-perméables (PPSP) afin de permettre la réception robuste de la diffusion vidéo sur WiMAX / WiFi-Link. Souvent, l'ensemble du cadre est ignoré, ce qui est la principale raison de la sous-utilisation des outils d'atténuation d'erreur. L'objectif de cette thèse est d'explorer le cadre global d'un scénario inter-réseau des réseaux WiMAX et WiFi, où la diffusion de la vidéo à partir du réseau WiMAX à l'utilisateur du terminal dans le réseau WiFi est analysé. Un outil robuste nécessaire à ces fins devraient être en mesure de répondre à certaines exigences, qui facilitent le déploiement des solutions mentionnées ci-dessus, à savoir, (1) et (2). Le FS et des outils au niveau des paquets de décodage FEC qui seraient étudiées dans cette thèse devrait prendre en compte les exigences suivantes:

1. Afin de propager l'information soft aux couches supérieures, malgré le protocole d'autres fonctions pile, la FS devrait également être en mesure de travailler sur les valeurs mous, il faut être capable de transmettre l'information soft aux couches supérieures. De même, pour la diffusion fiable le décodeur au niveau des paquets FEC ne devrait pas entraver la circulation de l'information soft pour la couche d'APL.
2. Les outils robustes devraient essayer de transmettre nombre maximum de paquets à la couche d'APL, même les paquets ayant des charges utiles erronée. FS peut aider à segmenter les paquets solidement regroupés même si les têtes des petits paquets à l'intérieur de la salve sont erronées. Cela permet de paquets erronés, qui sont par ailleurs chuté, pour atteindre la couche APL où ils peuvent être bien compris par les décodeurs JSC. La charge utile corrompu transmis à la couche d'APL pourrait, par exemple, dépasse la limite tolérable de décodeur vidéo résistante aux erreurs. Par conséquent, pour la PPSP, il faut réduire davantage les pertes de paquets à l'aide au niveau des paquets FEC. Tout au niveau des paquets FEC devrait, avec les couches inférieures perméable, être capable de passer nombre maximum de paquets à la couche d'APL.
3. Plusieurs licenciements présents dans la couche de protocole (champs connus en-têtes,

la présence de CRC, Header Check Sequence (HCS), ou des sommes de contrôle, etc) doit être utilisée pour effectuer FS robuste et de réduire au minimum la possibilité d'abandonner un paquet. De même, quelques licenciements explicites présents dans l'en-tête de paquet doit être exploité pour améliorer les performances du décodeur robuste FEC au niveau des paquets.

4. Un outil robuste ne devrait pas modifier les fonctionnalités émetteur et doit rester compatible avec PPS de l'émetteur.

En plus des exigences ci-dessus, le cadre devrait permettre l'utilisation conjointe d'informations entre les couches de la PPSP et l'échange d'avantages de ses outils constitutifs. D'un côté, par exemple, un décodeur robuste JSC déployés à couche APL peuvent bénéficier de la reprise d'en-tête robuste, FS, et au niveau des paquets FEC décodeur dans les couches inférieures, car ils augmentent le nombre de paquets transmis à elle. De l'autre côté, les FS et au niveau des paquets FEC décodeurs peuvent bénéficier de l'information soft fourni par la couche PHY et de la capacité de détection des erreurs de plusieurs décodeurs source fiable à la couche APL.

Partie I: Synchronisation châssis robuste

Dans la première partie de cette thèse, nous considérons le problème de la délinéation de paquets agrégés en macro-paquets. Cette opération d'agrégation est réalisée dans de nombreux protocoles afin d'améliorer le rapport en-tête/charge utile des systèmes de communication. Plusieurs méthodes de délinéation sont proposées. Elles exploitent d'une part les informations souples provenant des couches protocolaires basses ainsi que la redondance présente dans les paquets devant être séparés. L'ensemble des successions possibles de paquets au sein d'un macro-paquet est décrit à l'aide d'un trellis. Le problème de délinéation est transformé en un problème d'estimation de l'état d'une variable aléatoire Markovienne, pour lequel de nombreux algorithmes (BCJR, Viterbi) sont disponibles. Cette technique est très efficace mais complexe. De plus, elle nécessite la réception de l'ensemble du macro-paquet, ce qui peut induire des latences importantes. Dans une deuxième étape, nous proposons une technque où le décodage se fait sur une fenêtre glissante contenant une partie du macro-paquet. Un treillis glissant est considéré cette fois. La taille de la fenêtre permet d'ajuster un comproimis entre complexité et efficacité de décodage. Enfin, une méthode de décodage à la volée exploitant un automate à 3 état et des tests d'hypothèses Bayésiens permet de réaliser une délinéation moins efficace, mais sans latence. Ces méthodes sont comparées sur un problème de délinéation de paquets MAC dans des macro-paquets PHY dans WiMAX.

Sans perdre de vue mentionnés ci-dessus exigences, dans la première partie de cette thèse, nous proposons plusieurs approches Protocole-Channel Decodage Conjoint (PCDC) pour FS. Ils exploitent toutes les informations disponibles: l'information soft à la sortie du canal (ou décodeur de canal) ainsi que la structure des couches de protocole pour estimer les limites des paquets de petite et le contenu de leurs têtes.

Tout d'abord, une technique à base de treillis pour la SF est proposé, où l'agrégation de paquets est modélisé par un processus de Markov, qui permettent de représenter toutes les successions possibles de paquets dans un élan par un treillis inspirée de celle de [28]. Une modification de BCJR algorithme [29] est appliqué sur ce treillis pour obtenir les limites de paquet. Deuxièmement, un faible délai et la version à complexité réduite sous-optimale de l'algorithme de treillis à base est proposé. Il utilise l'approche d'un treillis de glisse (ST)-base inspiré de [30], où une variante à faible latence de l'algorithme BCJR

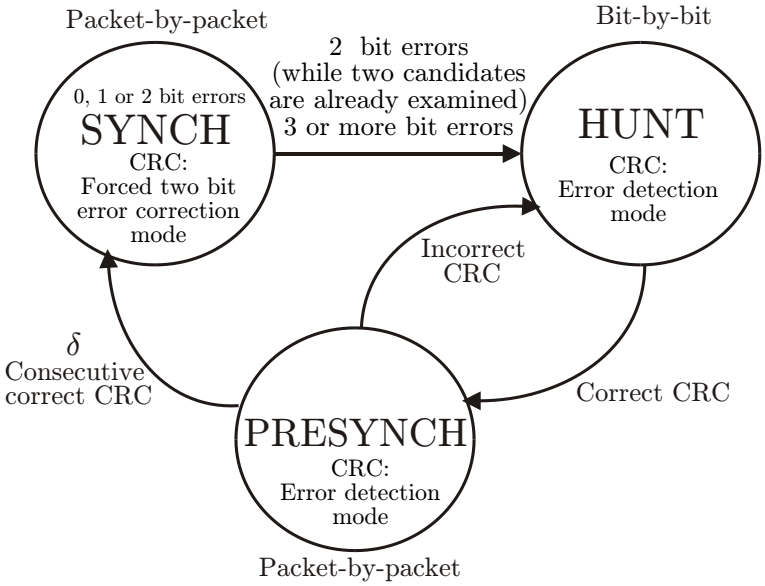

Figure 1: 3S automate utilisé en [17]

a été présenté pour le décodage des pays candidats. Ce sont des hold-et-sync (hronize) techniques, qui nécessitent l'ensemble (pour treillis à base de) ou partie (pour ST-base) de la salve d'effectuer FS.

Enfin, une technique à la volée, il est proposé, qui combine de solides technique de récupération en-tête inspirée de [8] avec des tests d'hypothèses bayésiens inspirée de [19; 20; 21; 22] pour localiser les limites de paquet via une recherche échantillon par échantillon . Nous utilisons un automate robuste 3S, dérivée de celle de [17], mais au lieu de CRC correction dur, une technique d'en-tête récupération douce [8] pour corriger les en-têtes endommagées (en exploitant toutes connues intra et inter-couche de licenciements) est exploité pour estimer le champ longueur de l'en-tête. En outre, le test d'hypothèse bayésienne, utilisé pour rechercher le SF correcte, offre des performances améliorées en raison de l'utilisation des informations du canal douce combinée avec une information a priori en raison de la redondance présente à la tête d'un paquet.

Description détaillée des états de l'automate 3S en [17] est fourni ci-dessous.

Considérons un éclat de L bits comprenant des paquets N agrégées. Cette salve contient soit paquets $N-1$ de données et un paquet contenant un rembourrage supplémentaire bits de remplissage uniquement, ou des paquets de données et pas de N bits de remplissage. Supposons que chacun de ces paquets, sauf le paquet de remplissage, contient un en-tête et une charge utile et suit la même syntaxe, comme décrit dans la section.

En supposant que L est fixé avant l'agrégation de paquets, et que N n'est pas déterminé a priori, la longueur cumulée ℓ en bits des n premiers paquets agrégées peuvent être décrits par un processus de Markov, qui est désigné par l'etat S_n. Un état transition priori probabilités $p\left(S_n = \ell \,|\, S_{n-1} = \ell'\right)$ peut être déduite facilement.

Si $\ell < L$, alors

$$p\left(S_n = \ell \,|\, S_{n-1} = \ell'\right) = \begin{cases} \pi_{\ell-\ell'} & \text{if } \ell_{\min} \leqslant \ell - \ell' \leqslant \ell_{\max} \\ 0 & \text{else,} \end{cases} \tag{1}$$

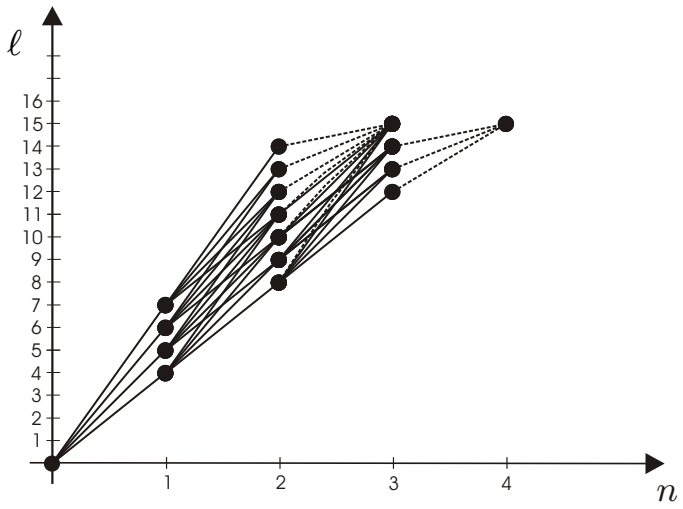

Figure 2: Trellis représentant la longueur totale autorisée en bits L vs le nombre de paquets n dans un élan de $L = 15$ bits avec $\ell_{\min} = 4$ bits et $\ell_{\max} = 7$ bits. Les lignes en pointillés correspondent à des bits de bourrage.

et si$\ell = L$, alors

$$p\left(S_N = L \,|\, S_{N-1} = \ell'\right) \quad = \quad \begin{cases} 0, \text{ if } L - \ell' > \ell_{\max} \\ 1, \text{ if } 0 < L - \ell' < \ell_{\min} \\ \sum\limits_{k=L-\ell'}^{\ell_{\max}} \pi_k, \text{ else.} \end{cases} \tag{2}$$

Dans le treillis telle que celle de la figure 2, ℓ identifie un état S_n à horloge-paquets $n = 0, \ldots, N_{\max}$. Si, au horloge-paquets n, $L - \ell \geq \ell_{\max}$, puis le paquet suivant est nécessairement un paquet de données, sinon le paquet de remplissage peuvent également être présents. Les transitions en pointillés correspondent à des paquets de matelassage et de la plaine transitions correspondent à des paquets de données. Pour le dernier paquet (quand $S_n = L$), en pointillés et des transitions lisses peuvent être parallèles.

Pour $n = 1, ..., N_{\max}$,, BCJR classique avant et arrière récurrences peuvent être effectuées, avec

$$\alpha_n(\ell) \quad = \quad \sum_{\ell'} \alpha_{n-1}(\ell')\gamma_n(\ell', \ell), \tag{3}$$

$$\beta_n(\ell) \quad = \quad \sum_{\ell'} \beta_{n+1}(\ell')\gamma_{n+1}(\ell, \ell'), \tag{4}$$

et

$$\gamma_n\left(\ell', \ell\right) \quad = \quad P(S_n = \ell, \mathbf{y}_{\ell'+1}^{\ell} | S_{n-1} = \ell'). \tag{5}$$

La complexité de l'algorithme de FS simplifie à

$$\mathcal{N}_{\mathrm{n}} = \frac{L^2}{2}\left(\frac{\ell_{\max} - \ell_{\min}}{\ell_{\max}\ell_{\min}}\right) = \mathcal{O}\left(L^2\right). \tag{6}$$

Pour le cas de la technologie WiMAX, nous avons montré que toutes les successions de paquets MAC dans un éclat peut être représenté par un treillis sur lequel une variante de l'algorithme BCJR a été mis au travail pour robuste identifier les limites de chaque paquet MAC. Pour l'erreur de localisation MAC similaires par paquets, sur un canal AWGN, un gain d'environ 8 dB de SNR par rapport à la technique classique à base de HD est observée. La technique de treillis à base fournit les meilleures performances comme prévu, mais au prix d'un délai équivalent à la longueur d'une rafale. Ainsi, dans le chapitre suivant, nous proposer des modifications nécessaires pour le mettre au point un faible retard et réduit la complexité technique de FS.

La technique de treillis à base de FS présentés est une technique de hold-et-sync FS, ce qui nécessite l'information soft à être reçus avant le FS de la scène. En outre, il nécessite la connaissance de :

1. le début et
2. la durée de la salve d'effectuer FS.

Ces deux hypothèses nécessitent un décodage sans erreur des en-têtes des couches inférieures du protocole, qui contiennent cette information. Cela peut être fait en utilisant des méthodes présentées dans [8], qui permettent à la couche inférieure de l'avant l'éclatement de la couche où il est traité.

Contrairement à ce que le treillis en [29], le treillis considéré comme la figure 2 a un nombre variable d'états pour chaque valeur du paquet d'horloge n. On peut appliquer directement les idées SW-BCJR, mais en raison de l'augmentation de la taille du treillis (du moins pour les petites valeurs de n), ce serait encore besoin de treillis gros pour être manipulé, avec un temps de calcul augmente. Ici, une approche fondée sur ST est introduit: un treillis de taille réduite est pris en considération dans chaque fenêtre de décodage. Comme dans [31], un certain chevauchement entre les fenêtres est envisagée pour permettre une meilleure réutilisation des quantités déjà calculées pour atteindre une complexité efficacité compromis.

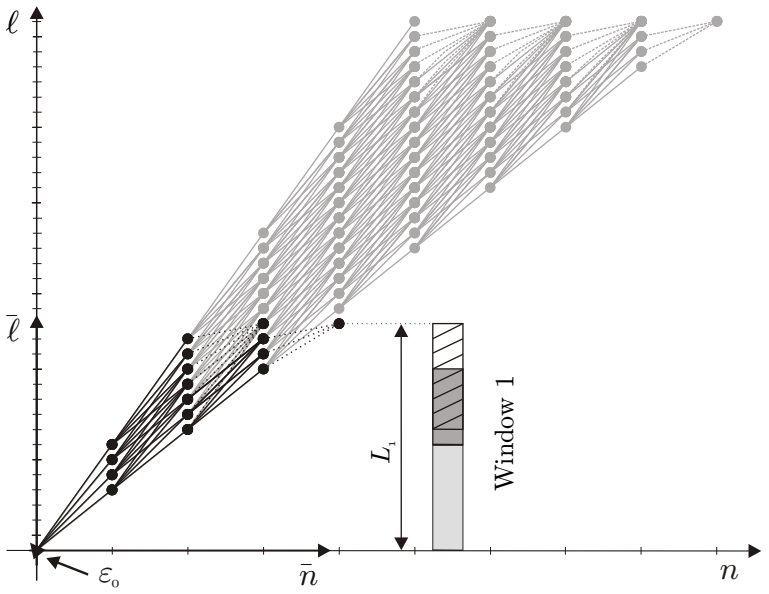

Figure 3: ST pour la fenêtre de décodage d'abord, le treillis d'origine est en gris

Maintenant, nous allons comparer la complexité de la technique ST-fondé FS à celle de la technique de treillis à base de FS présenté dans le chapitre précédent. Pour la technique ST-base FS, considérer les fenêtres M d'environ la même taille L^w, qui se chevauchent en moyenne sur $L^o = \ell_h + 1.5\ell_{\max}$ bits. Pour suffisamment grand L, on aura à traiter

$$M \approx \frac{L}{L^w - L^o}$$

fenêtres qui se chevauchent avec autant de tribus nombreuses, chacune avec

$$\mathcal{N}_{\mathrm{n}}^w \approx \frac{(L^w)^2}{2}\left(\frac{\ell_{\max} - \ell_{\min}}{\ell_{\max}\ell_{\min}}\right)$$

n¡uds. Le nombre total de n¡uds à traiter est alors

$$M\mathcal{N}_{\mathrm{n}}^w \approx \frac{L}{L^w - L^o}\frac{(L^w)^2}{2}\left(\frac{\ell_{\max} - \ell_{\min}}{\ell_{\max}\ell_{\min}}\right). \tag{7}$$

Le nombre total de n¡uds est bien inférieure à celle de la technique de treillis à base de FS. Une comparaison pour différentes valeurs de taille de rafale L est fourni dans le Tableau 1 pour taille de la fenêtre $L^w = 480 + L^o$, dans le Tableau 2 de taille de la fenêtre $L^w = 600 + L^o$, et dans le tableau 3 pour taille de la fenêtre $L^w = 900 + L^o$.. Par exemple, lorsque $L = 12960$ octets et $L^w = 480 + L^o$, la technique ST-fondé de réduire le nombre de n¡uds d'un facteur 10 par rapport à la technique de treillis à base de. Figure [fig: No_of_Nodes] compare le nombre de n¡uds dans la technique de treillis à base de la technique ST-base pour différentes valeurs de taille de rafale L et taille de la fenêtre L^w.

L (octets) ST-base (nombre de n¡uds) Complexité Gain

L (octets)	1800	4000	8000	12960	16000	24000
Treillis à base (# de n¡uds)	24300	120000	480000	1259712	1920000	4320000
ST-base (# de n¡uds)	17400	38600	77200	125100	154450	231700
Complexité Gain	1.4	3.1	6.2	10.0	12.5	18.6

Table 1: Comparaison entre la complexité du treillis à base de ST et à base de ($L^w = 480 + L^o$) techniques FS.

L (octets)	1800	4000	8000	12960	16000	24000
Treillis à base (# de n¡uds)	24300	120000	480000	1259712	1920000	4320000
ST-base (# de n¡uds)	18500	41000	82000	133000	164200	246200
Complexité Gain	1.3	2.9	5.8	9.5	11.7	17.5

Table 2: Comparaison entre la complexité du treillis à base de ST et à base de ($L^w = 600 + L^o$) techniques FS.

Un peu de retard, à complexité réduite et plus efficace, technique ST-fondé FS est présenté, qui effectue FS en divisant la salve reçue dans les fenêtres superposées. Dans la méthode proposée WiMAX ST-base est déployée pour exécuter FS des rafales à la couche MAC, pour lesquels les simulations sont également réalisées. Par rapport à la technique de treillis à base de FS, la FS délai et la complexité de calcul est considérablement réduit. Le prix à payer est légère dégradation des performances. Un gain significatif en performance

L (octets)	1800	4000	8000	12960	16000	24000
Treillis à base (# de n¡uds)	24300	120000	480000	1259712	1920000	4320000
ST-base (# de n¡uds)	21800	48500	96900	157000	193900	290900
Complexité Gain	1.1	2.5	4.9	8.0	9.9	14.9

Table 3: Comparaison entre la complexité du treillis à base de ST et à base de ($L^w = 900 + L^o$) techniques FS.

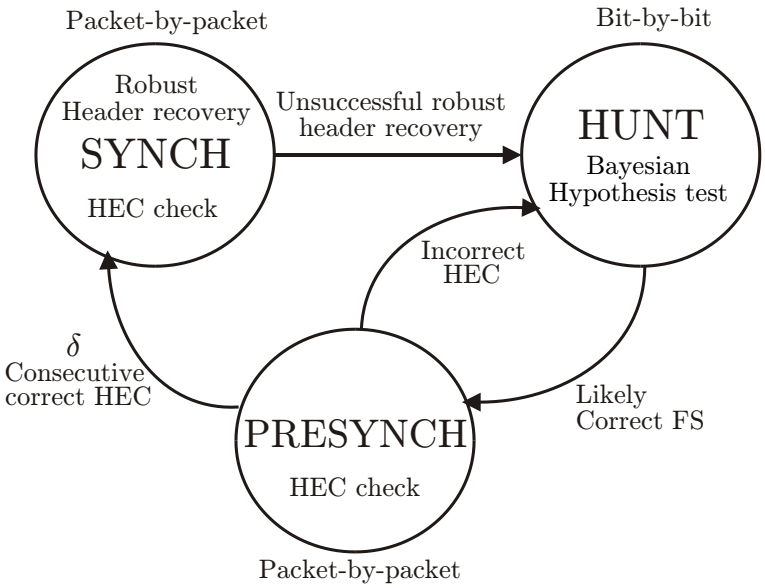

Figure 4: Robust 3S FS automaton

est obtenue pour les deux canaux AWGN et canal à évanouissements de Rayleigh par rapport aux techniques FS traditional.

Proposé robuste 3S automproposé robuste 3S automate FS est illustré à la figure 4. En tle proposé FS trois états, nous avons proposé plusieurs améliorations à la méthode à base de HEC Ueda à développer un automate robuste 3S, voir la figure 4. Tout d'abord, au lieu d'effectuer la correction d'erreurs dans l'état SYNC, l'estimation robuste en-tête présenté dans [Marin2008] est utilisée pour estimer le champ de longueur d'un paquet. En cas d'échec de vérifier HEC de tête avec le champ longueur estimée (HEC vérification est effectuée après le remplacement du champ bruité reçu longuement avec le champ de longueur estimée), l'automate passe à l'état HUNT, où des tests d'hypothèses bayésiens est réalisée pour rechercher les FS correct. L'opération réalisée dans l'état PRESYNC reste inchangé.

Les résultats de simulation pour la transmission sur un canal AWGN et Rayleigh sont présentées. La méthode proposée 3S FS donne de meilleurs résultats que la méthode de MU, en raison de l'utilisation des tests d'hypothèse bayésienne en fonctionnement la chasse et à la technique de récupération en-tête robuste. La méthode MU fonctionne mal, surtout à faible SNR, en raison du fait qu'il utilise la détection dur HEC / correction et la taille du HCS (8 bits) est faible par rapport à la taille de l'en-tête (48 bits), donc plus de 10 candidats pour deux syndrome erreur sur les bits doivent être considérés. Le projet 3S FS automate fonctionne mieux au SNR faible en raison de l'efficacité des tests d'hypothèse

bayésienne dans l'état HUNT, qui peut récupérer rapidement FS. La différence entre 3S FS et diminue MU au rapport signal / bruit, car même si la reprise d'en-tête fonctionne bien, FS erronées par des tests d'hypothèses bayésiens dégrade les performances, car même dans un état bon canal, on peut à tort FS correcte en raison de la simulation de tête par hasard données. La technique de treillis à base fournit les meilleures performances comme prévu, mais au prix d'un délai équivalent à la longueur d'un paquet global (burst). La proposition technique 3S FS FS prévoit clairement améliorée par rapport au compromis l'état des algorithmes de pointe, offrant ainsi raisonnable entre la performance et de latence. La méthode donne des HSA FS FS étage d'erreur au rapport signal / bruit, comme prévu, en raison de la persistance inévitable et les fausses alarmes, que les données de charge utile peut simuler l'en-tête. Depuis, la méthode HSA utilise plus de redondance de l'en-tête, il donne un sol beaucoup plus faible par rapport à la méthode NP FS. Performance de la méthode MU est très touchée par canal de Rayleigh, alors que les résultats pour le projet 3S automate montrent clairement son efficacité dans ce contexte.

Les techniques présentées ici FS ne nécessitent pas de frais généraux de signalisation, c'est à dire, pas de marques de synchronisation sont ajoutés et que les informations disponibles dans la couche de protocole est utilisé. En outre, ces techniques sont assez générales, elles sont illustrées par la synchronisation des paquets MAC WiMAX agrégées en rafales, qui sont transmis à la couche PHY [32], mais ils peuvent être facilement étendus à d'autres protocoles où l'agrégation de paquets est effectué.

Partie II: Robuste décodage FEC au niveau des paquets

Dans la deuxième partie de la thèse, nous proposons de réaliser un décodage souple des codes en blocs utilisés dans certaines couches de piles protocolaires pour le multimédia. Cdes sorties souples sont générées afin de permettre un décodage conjoint des en-têtes et de la charge utile au niveau des couches supérieures. Nous avons en particulier étudié des outils de décodage souple ldans le cas de la norme RTP FEC, et avons comparé les performances du décodeur proposé avec des approches classiques de décodage.

Pour diffuser les paquets multimédia sur WiMAX / WiFi-Link, un paquet de niveau FEC régime est analysé afin de surmonter les retards de retransmission. Tenant compte des exigences mentionnées plus tôt, au lieu d'effectuer le décodage des bits dur, l'information soft transmis par les couches inférieures est utilisé pour récupérer les paquets erronés. Le décodeur au niveau des paquets FEC est déployé à la couche RTP, où il est supposé que les paquets RTP atteindre le décodeur FEC sont doux-évalués et peuvent avoir des erreurs. L'idée même de PCDC telle qu'elle est déployée dans les techniques de FS se met au travail pour développer un paquet niveau Maximum A Posteriori (MAP) pour estimer le décodeur paquets erronés, en utilisant les licenciements en-tête RTP, la redondance introduite grâce à l'utilisation de la FEC, et la probabilité à partir du canal. Plus particulièrement, il ne provoque aucune entrave à la circulation de l'information soft des couches inférieures, à travers la couche RTP, aux couches supérieures. En outre, le décodeur robuste présenté ici a besoin d'aucune information côté et reste totalement compatible avec la RFC 5109.

Bien que l'utilisation de paquets FEC redondants diminuerait le débit utile du système, mais étant donné que la retransmission est désactivé, on peut utiliser cette goodput épargné pour la transmission des paquets FEC redondantes. Sur la base de l'état du canal et la nature des services (vidéo en temps réel, données, voix, etc), l'émetteur peut décider pour chaque flux de l'application ou du service, soit d'utiliser ou de retransmission régime FEC. Transmetteur commutateurs au régime FEC si la condition canal tombe en dessous d'un

certain niveau ou dans des situations où la retransmission entraîne une perte de débit utile ou retard inacceptable.

Juste avant le décodeur de source à la couche APL. Par exemple, dans [1] un récepteur perméable-couche et un schéma de FEC (à couche APL), capable d'exploiter traditionnellement inutile des informations erronées, est proposé. Dans ce schéma FEC, tous les segments (même erronée) du paquet IP sont transmis à la couche d'APL, où le décodeur FEC récupère les paquets IP, mais erronée segments restent inutiles. De même, dans [11; 9], les auteurs ont proposé de modifier le récepteur et ont identifié que seuls quelques domaines importants présents dans l'en-tête UDP sont suffisantes pour identifier de manière unique chaque session active multimédia avec peu de fausses alarmes, ainsi même les paquets avec des en-têtes erronés peut être transmis à la couche supérieure. RS est alors utilisé comme un code FEC, pour décoder les erreurs (en paquets corrompus) et ratures (de détections manquées) dans la charge utile UDP, en même temps. Néanmoins, les fausses alarmes peuvent désynchroniser la vidéo et / ou décodeur FEC, ce qui exige SN fiables pour fournir des emplacements d'effacement.

Pour la connaissance de l'auteur, il n'existe pas au niveau des paquets schéma de décodage FEC qui utilisent des licenciements inter-paquet intra et présenté par la structure de l'en-tête, dont le bénéfice a déjà été prouvée dans les approches FS, présenté plus tôt dans cette thèse. Nous soutenons que l'utilisation de l'idée de PCDC, on peut utiliser conjointement les

1. l'information soft du canal,
2. licenciements plusieurs présents dans l'en-tête de paquet, et
3. les licenciés FEC

pour améliorer l'efficacité de récupération en-tête du décodeur FEC. En outre, une libre circulation de l'information soft de la couche PHY pour la couche d'APL, par l'intermédiaire du décodeur FEC, est essentiel pour les techniques de DCSC plusieurs afin d'augmenter l'efficacité du décodeur vidéo. Bien que, la proposition de protocole de couche perméable [8], en l'absence de niveau des paquets FEC, peut permettre l'échange d'information souple et partiellement erronée entre les couches de protocole, il n'y a pas disponibles au niveau des paquets souples de sortie FEC décodeur qui peut permettre la circulation simultanée de l'information soft et partiellement erronée de la couche d'APL.

La protection de la FEC et le décodeur PCDC à base FEC présenté dans cette thèse sont bien optimisé pour la couche RTP, mais ils peuvent être étendus aux autres couches de la PPSP. En outre, ils restent compatibles avec les scénarios de diffuser d'autres, par exemple, la réception de la TV Mobile en DVB-H (Digital Video Broadcasting - Handheld) [33; 34], puis rediffusion sur le réseau WiFi.

De nombreux outils sont présentés dans cette thèse, de réduire considérablement le nombre de paquets qui doivent être supprimés et permettre des flux de soft-paquets (paquets-s), qui peuvent contenir des erreurs, aux couches supérieures et d'améliorer la performance des outils robustes fonctionnement à les couches supérieures. Ils peuvent ensuite être transmis à la couche d'APL en utilisant les techniques présentées dans Protocole de Pile Perméables (PPP) [13; 14; 8], et robuste décodées en utilisant des techniques DCSC [35; 36].

Prenons le même schéma simple au niveau des paquets FEC comme nous le verrons dans le chapitre précédent, où deux MedPs sont codés pour donner un FeCp utilisant la RFC 5109 et l'opération XOR sur les paquets de papier sans erreur peut être utilisé pour reconstruire le paquet perdu. Toutefois, dans plusieurs situations, il n'est pas nécessaire pour recevoir des paquets sans erreur, en particulier dans le cas où les couches inférieures

sont perméables, c'est à dire, capable de transmettre des paquets erronés aux couches supérieures (requis par exemple pour un décodage robuste de la vidéo paquets), au lieu de les laisser tomber. C'est exactement le cas dans le PPSP a examiné en détail au chapitre . En raison de l'utilisation de la technique d'accès multiple (HCCA) prévu dans 802.11e [37], on peut raisonnablement supposer qu'aucun paquet est complètement perdu dans le canal, mais en fait, il peut être erroné. Ainsi, nous avons toujours tous les paquets (MedPs & FeCp) d'un bloc MedPs-FEC atteindre la couche RTP-FEC. En outre, la PPSP exige que le débit de s-paquets à la couche APL, ce qui nécessite le développement de soft-sorties au niveau des paquets FEC décodeur.

Considérons un s-paquet $\mathbf{x}_n$, où $n = \{1, 2\}$ correspondent aux MedPs et $n = 3$ correspond à une FeCp généré en utilisant le savoir MedPs,

$$\mathbf{x}_3 = \mathbf{fec}\left(\mathbf{x}_1, \mathbf{x}_2\right).$$

Lorsque, **fec** est une fonction d'encodage FEC, qui peut être aussi simple que, par exemple, une opération XOR. Un bloc MedPs-FEC, c'est à dire, un groupe de paquets $\mathbf{x}_1$, $\mathbf{x}_2$, et $\mathbf{x}_3$, est transmis sur le canal sans mémoire. Soit le vecteur $\mathbf{y}_n$ contient la sortie du canal du paquet \ mathbf $\mathbf{x}_n$. Au niveau du récepteur, il est supposé que tous les trois paquets sont disponibles, mais ils peuvent être erronées.

Maintenant, notre objectif est d'obtenir des estimations conjointes PAM $\overline{\mathbf{x}}_1$ et $\overline{\mathbf{x}}_2$ of $\mathbf{x}_1$ et $\mathbf{x}_2$, respectivement, en utilisant la connaissance de $\mathbf{y}_1$, $\mathbf{y}_2$, et $\mathbf{y}_3$, c'est à dire,

$$\left(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2\right) = \arg\max_{\mathbf{x}_1, \mathbf{x}_2} p\left(\mathbf{x}_1, \mathbf{x}_2 \,|\, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right). \tag{8}$$

Un décodeur RTP niveau FEC est presente est présent chapitre, qui utilise (i) l'introduction FeCp redondants, (ii) le présent redondance explicite dans l'en-tête RTP, et (iii) l'information risque de la chaîne. Il faut profiter pleinement de la PPSP en exploitant les informations transmises à doux, ce qui le rend efficace même avec des paquets erronés. En outre, il montre plus de robustesse à l'erreur sur les bits en raison de l'utilisation des licenciements structurels.

La comparaison avec le régime de retransmission traditionnels ont montré que le décodage MAP serait capable de tolérer la dégradation de plusieurs canaux, puis le régime de retransmission. Il réduit le PER et donne plus de marge de compensation au cas où le canal se dégrade, tout en ne provoquant pas de retard et entrave à la circulation de l'information soft, à travers la couche RTP, pour la couche d'APL. Le décodeur MAP couplé avec des détecteurs robustes erreur et décodeurs vidéo (à couche APL) est un outil efficace pour atténuer les erreurs laissées par les couches inférieures de la PPSP.

Bien que, seul un cas simple d'EEP est présenté, le décodeur peut être facilement adaptée pour UEP. Le décodeur proposé peut également être prolongé pour des codes FEC plus complexes et robustes, tels que les codes RS. En outre, il peut être déployé au niveau des couches inférieures, par exemple, au niveau des couches MAC ou UDP, où l'on peut également exploiter la redondance disponible en raison de sommes de contrôle / CRC.

En résumé, les techniques de décodage conjoint proposées permettent de réduire le nombre de paquets perdus, d'augmenter le nombre de paquets transmis vers les couches applicatives où des décodeurs source-canal conjoints peuvent être utilisés pour améliorer la qualité de la vidéo reçue.

Aperçu de thèse

cette thèse est organisé comme suit:

Chapitre I: Ce chapitre explique d'abord un scénario basé PPS inter-réseau des réseaux WiMAX et WiFi. Il détaille ensuite les modifications requises nécessaires au développement de PPSP, qui peut permettre des flux de doux et d'information corrompu à couches supérieures de protocole. Le cadre global présenté motivera la nécessité pour les propositions présentées dans les chapitres à venir. Le scénario inter-réseaux avec les S-PSP, présentées dans ce chapitre, est considéré tout au long de cette thèse.

Partie I: Synchronisation châssis robuste

Chapitre II: Une introduction à la FS est présentée dans ce chapitre. Nous présentons brièvement les travaux connexes sur les techniques de FS de la littérature. Nous expliquons les techniques de longueur fixe FS [38; 39] et la longueur variable sur FS [19; 20; 21; 40; 22; 17].

Chapitre III: Ce chapitre présente une technique à base de treillis pour FS, ce qui est compatible avec le PPSP.

Chapitre IV: Dans ce chapitre, nous montrons comment la technique de FS du chapitre III peut être modifié pour développer un faible délai et une variante du sous-optimale à complexité réduite.

Chapitre V: Dans ce chapitre, nous prenons le meilleur parti de FS présentées dans le chapitre II et affiner leur proposer automate 3S pour FS. Cette extension vise à fournir une technique de FS qui peuvent travailler sur la volée, contrairement aux techniques FS treillis.

Partie II: Robuste décodage FEC au niveau des paquets

Chapitre VI: Nous présentons une introduction à la FEC au niveau des paquets, puis fournir une évaluation de la faisabilité de la retransmission et les régimes au niveau des paquets FEC. Il illustre en outre pourquoi au niveau des paquets FEC est important et intéressant pour le scénario de la radiodiffusion.

Chapitre VII: Le projet au niveau des paquets FEC décodeur est présentée dans ce chapitre. Ce décodeur au niveau des paquets FEC est appliquée à la couche RTP, où les paquets FEC sont générés par la RFC 5109 protocole.

Chapitre VIII: Nous discutons les conclusions de notre thèse dans ce chapitre et fournissent également des applications possibles des techniques de FS et le niveau des paquets FEC décodeur présenté dans cette thèse.

Cinq annexes ont été jointes à cette thèse. Annexes A.1 et A.2 sont des références pour la structure de la trame PHY et MAC de paquets dans le WiMAX, respectivement. Annexe A.3 détaille les domaines du en-tête UDP, annexe A.4 détaille les domaines en-tête de le UDP-lite, et annexe A.5 détaille les domaines en-tête du RTP.

– Say: Truly, my prayer and my service of sacrifice, my life and my death, are [all] for Allah, the Cherisher of the Worlds No partner hath He: this am I commanded, and I am the first of those who bow to His will. *(Quran : Surat Al-anaam, 162-163)*

– Education is a progressive discovery of our own ignorance. *(Will Durant, 1885 - 1981)*

To my family and friends
&
PAKISTAN

Acknowledgments

After Acknowledging the Allah Almighty's mercy and blessing, I must admit that this PhD thesis is not the fruit of my effort alone since many people have helped me throughout three years. I would like to show special thanks to my supervisors, Prof. Pierre Duhamel and Prof. Michel Kieffer, for their excellent academic advice and continues support . They always encouraged me to define the research problem and helped me to identify and refine research ideas. Without their invaluable advice, help, suggestions, and planning, this thesis would not have been possible. Their precise and critical questions have made me realize that the research is nothing but making things clear. Their pointing of my mistakes during the coarse of the research process helped me to refine my method of approaching towards the problem. I would also like to thank honorable reviewers/jury members Prof. Marco Chiani, Prof. Jean-Marie Gorce, and Prof. Béatrice Pesquet-Popescu for their constructive comments to improve the manuscript.

I would like to express my highest gratitude to my parents M. Khalid and Zahida Begium. This thesis is dedicated to them. I want to give my greatest wishes to my daugther Sibgha Noor Ali - may she has a wounder and successful life.

I must also acknowledge the Higher Education Commission of Pakistan and their financial support during my M.S. studies. I thank Microsoft Research for funding this thesis through the European PhD Scholarship Programme, which has allowed me to do my studies in the best conditions possible.

There are many others including my colleagues at LSS/Supelec to whom I would like to broadcast this acknowledgment ... THANK YOU!

List of Acronyms

This list contains the main abbreviations and acronyms used throughout this thesis.

Acronyms	**Description**
ACK	Acknowledgment
APP	A Posteriori Probability
ATM	Asynchronous Transfer Mode
AWGN	Additive White Gaussian Noise
APL	Application (a protocol stack layer)
BER	Bit Error Rate
BS	Base Station
CC	Convolutional Code
CRC	Cyclic Redundancy Check
CID	Connection IDentifier. A 16-bit identification of a WiMAX MAC connection
CI	Coarse Initialization
CD	Cell Delineation. A process of synchronizing fixed-length ATM packets (cells)
DL	Down-link
EPLR	Erroneous Packet Location Rate
EDCA	Enhanced Distributed Channel Access. A prioritized but contention based channel access scheme
EEP	Equal Error Protection. A constant protection that is independent of the importance of data
FA	False Alarm. A false detection of the start of SW/header
FCH	Frame Control Header. It provides the configuration information of the WiMAX PHY frame
(A)FEC	(Adaptive) Forward Error Correction
FECP	Forward Error Correction Packet
FS	Frame Synchronization
HC	Hybrid Coordinator. A centralized channel access controller residing in the QAP
HEC	Header Error Control. A header field to protect header.
HCS	Header Check Sequence. A WiMAC MAC header field to protect MAC header.
HCCA	Hybrid Controlled Channel Access. A channel access scheme using a polling mechanism
HD-based	Hard Decision-based. A simple FS technique that uses packet length field to perform FS

Acronyms	**Description**
HSA	HUNT State Alone. A simple FS automaton that contains only a single state (*i.e.,* HUNT)
(S-)IP	(Soft-)Internet Protocol (a protocol stack layer)
JSC(D)	Joint Source Channel (Decoding)
JPC(D)	Joint Protocol Channel (Decoding)
LR(T)	Likelihood Ratio (Test)
MedP	Media Packet
MAN	Metropolitan Area Network
(S-)MAC	(Soft-)Medium Access Control (a protocol stack layer)
MAP	Maximum A Posteriori
MTU	Maximum Transmission Unit. The size (in bytes) of the largest protocol data unit
NP	Neyman-Pearson. A criterion used to choose the threshold for LRT
NALU	Network Adaptation Layer Unit. A network friendly data unit generated by a video encoder
PER	Packet Error Rate
(S-)PHY	(Soft-)Physical (a protocol stack layer)
PI	Precise Initialization
QAP	QoS-Access Point. It controls the channel accessed by the WiFi SSs.
QSTA	QoS enabled 802.11e Station
QoS	Quality of Service
RTP	Real-Time Transport Protocol (a protocol stack layer)
RS	Reed-Solomon
SN	Sequence Number
SPS	Standard Protocol Stack. A stack defined by the WiFi/WiMAX standard
S-PPS	Soft-Permeable Protocol Stack. A stack that allow exchange of soft and erroneous information between layers.
s-packet	soft-packet. It contains bit APPs or soft information from the channel
ST	Sliding Trellis
SW-BCJR	Sliding Window variant of the BCJR algorithm
SNR	Signal to Noise Ratio
SW	Synchronization Word. A fixed bit-pattern inserted to facilitate FS at the receiver
SS	Subscriber Station
TS	Traffic Streams
(S-)UDP	(Soft-)User Datagram Protocol (a protocol stack layer)
UL	Up-link
UEP	Unequal Error Protection. A variable protection that depends on the importance of data
WLAN	Wireless Local Area Network
3S automaton	Three-State automaton. A FS technique having three states, each performing distinct function

Summary

This PhD study intends to investigate the tools necessary to implement a device (the *WiBOX*), which can robustly receive video broadcast over WiMAX and then rebroadcast it over WiFi. WiBOX should not only provide WiMAX services access to a WiFi user, but it should also achieve reasonable video quality even with a very weak WiMAX signal, and at the same time for WiFi rebroadcast, it should utilize alternative recovery techniques and avoid delays caused by the conventional retransmissions. This would help to improve WiFi user quality and to remain consistent with the broadcast scenario. To achieve the said objectives one has to consider several robust tools, which are often deployed to solve problems, like packet loss, synchronization failures, high delay, throughput etc., encountered while receiving video through a WiMAX/WiFi-link. These robust tools can be deployed at several protocol layers, among them few notable are, *e.g.*, *Joint Source Channel Decoding* (JSCD) techniques deployed at the application (APL) layer, iterative decoding techniques deployed at the physical (PHY) layer, and header recovery, estimation, or synchronization tools deployed at various layers.

For an efficient performance of these robust tools some cross-layer approach to enable exchange of useful information between the protocol layers and the complete analysis of the protocol stack is required. Some of these tools have requirements that are not compliant with the Standard Protocol Stack (SPS) and require Soft-Permeable Protocol Stack (S-PPS), which can allow flow of erroneous packets, containing the *soft* information*, e.g.*, *A Posteriori* Probabilities (APP) or likelihood ratios, to the higher layers. More importantly, for performance enhancement these tools should mutually benefit and reinforce each other instead of undoing each other's advantage.

To increase the throughput, in both WiMAX and WiFi communication standards, packet aggregation is used; several packets are aggregated at a given layer of the protocol stack in the same *burst* to be transmitted. One can deploy *Frame Synchronization* (FS), *i.e.*, to synchronize and recover the aggregated packets, however, when transmission over a noisy channel is considered, FS can cause loss of several error-free or partially error-free packets, which could otherwise be beneficial for other tools, *e.g.*, JSCD and header recovery tools, functioning at higher layers of the S-PPS. Rebroadcasting video over WiFi can significantly increase packet loss rate as the retransmission is omitted, which can be overcome by the *packet-level* Forward Error Correction (FEC) techniques. The FS and packet-level FEC decoder for S-PPS should not only allow flow of soft information from the PHY layer but should also mutually benefit from the JSC decoders deployed at the APL layer. In this thesis, we propose several *Joint Protocol-Channel Decoding* (JPCD) techniques for FS and packet-level FEC decoders operating at S-PPS.

In the first part of this thesis, we propose several robust FS methods for S-PPS based on the implicit redundancies present in protocol and the soft information from the soft decoders at PHY layer. First, we propose a *trellis*-based algorithm that provides the APPs of packet boundaries. The possible successions of packets forming an aggregated

packet are described by a trellis. The resulting algorithm is very efficient (optimal in some sense), but requires the knowledge of the whole aggregated packet beforehand, which might not be possible in latency-constrained situations. Thus in a second step, we propose a low-delay and reduced-complexity *Sliding Trellis* (ST)-based FS technique, where each burst is divided into overlapping windows in which FS is performed. Finally, we propose an *on-the-fly* three-state (3S) automaton, where packet length is estimated utilizing implicit redundancies and Bayesian hypothesis testing is performed to retrieve the correct FS. These methods are illustrated for the WiMAX Medium Access Control (MAC) layer and do not need any supplementary framing information. Practically, these improvements will result in increasing the amount of packets that can reach the JSC decoders.

In the second part, we propose robust packet-level FEC decoder for S-PPS, which in addition to utilizing the introduced redundant FEC packets, uses the soft information (instead of *hard* bits, *i.e.*, bit-stream of '1's and '0's) provided by the PHY layer along with the protocol redundancies, in order to provide robustness against bit error. Though, it does not impede the flow of soft information as required for S-PPS, it needs support from the header recovery techniques at the lower layers to forward erroneous packets and from the JSC decoders at the APL layer to detect and remove remaining errors. We have investigated the standard RTP-level FEC, and compared the performance of the proposed FEC decoder with alternative approaches.

The proposed FS and packet-level FEC techniques would reduce the amount of packets dropped, increase the number of packets relayed to the video decoder functioning at APL layer, and improve the received video quality.

Publications

Conferences

- U. Ali, M. Kieffer, and P. Duhamel, Joint Protocol-Channel Decoding for Robust Aggregated Packet Recovery at WiMAX MAC Layer, in the proceedings of IEEE 10th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), 2009. Perugia, Italy.
- U. Ali, M. Kieffer, and P. Duhamel, Frame Synchronization based on Robust Header Recovery and Bayesian Testing, in the proceedings of IEEE the 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) , 26 - 29 September 2010, Istanbul, Turkey.
- U. Ali, M. Kieffer, and P. Duhamel, Sliding Trellis Based Joint Protocol-Channel Decoding for Frame Synchronization, **submitted** to IEEE International Conference on Communications (ICC), 2011, Kyoto, Japan.

Journals

- U. Ali, M. Kieffer, and P. Duhamel, Frame Synchronization using Joint Protocol-Channel Decoding, **to be submitted** to IEEE Transactions.
- U. Ali and P. Duhamel, Packet-level FEC using Joint Protocol-Channel Decoding, **under construction**.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 General Introduction

Internet access has undergone a fundamental change in recent years, users are expanding their demand from web browsing and email to multimedia services, including Voice-over-IP (VoIP) and media streaming. IP-based television (IPTV), Video-on-Demand (VoD), and multimedia education/training applications over Internet are generating new market opportunities for the streaming industry. Streaming media especially over wireless networks is gaining popularity at an unprecedented rate, at home, at work, and in public hot spot locations. WiMAX [32] and WiFi [41] have emerged as promising broadband access solutions for the latest generation of Wireless Metropolitan Area Networks (WMANs) and Wireless Local Area Networks (WLANs), respectively. WiMAX allows various services, such as data, voice, video, and mobility, to be available within a single network and can support popular multimedia applications over Internet [42]. Likewise, WiFi can now provide Quality of Service (QoS) support, where the new IEEE 802.11e [27] has expanded the WiFi application domain by enabling applications such as voice and video services. Their complementary features enable the use of WiMAX as a backhaul service to connect multiple dispersed WiFi hotspots to the Internet. However, these technological advances have diverse challenges, which are imposed by, *e.g.*, bandwidth-intense, loss-tolerant, and delay-sensitive characteristics of the video streaming. For several applications, such as mobile WiMAX IPTV [43], transmission delays and packet losses are particularly critical [44]. Thus, the video reception over WiMAX and WiFi links seems to be very difficult.

Low packet loss rate is essential to provide good quality for multimedia services. Packet loss can have a destructive effect on the reconstructed video by making the presentation displeasing to human eyes. Packet losses are mainly due to the wireless nature of the transmission channel, in which transmission errors are almost unavoidable. Because of the error propagation properties of inter-frame video coding a single lost packet can cause quality degradations that can last in the order of a second depending on the encoding parameters. To overcome high error and to efficiently receive the broadcasted video, several tools, like robust video decoding, channel decoding, error recovery methods, etc., had to be employed at the receiver. Demodulators and channel decoders working at PHY layer are considered as the first front to reduce the Bit Error Rate (BER). For example, decoders proposed in [45; 46] provide an improved reception. To mitigate packet loss FEC techniques [47], like Low-Density Parity Check (LDPC), Reed-Solomon (RS), Convolutional Codes (CC), and Turbo codes, have long been widely deployed at the PHY layer. These techniques check and correct bit errors to ensure that the upper protocol layers receive error-free

packets. Unfortunately, when errors are not corrected at the PHY layer, they may lead to the loss of large packets at upper protocol layers.

In order to mitigate the errors left by the PHY layer, some error control schemes are needed to reduce the effective packet loss. Packet loss compensation techniques are divided into error concealment and loss recovery techniques. Error concealment techniques [48; 49] reconstruct the lost packet at the receiver, do not need any additional information from the sender, and can be used by the source decoders at APL layer. They exploit the redundancy (temporal and/or spatial) found in the multimedia data for estimating the missing information. Emerging multimedia standards like H.264 [50] have introduced enhanced error-resilience and concealment features, which enable video decoders to tolerate a certain level of packet losses [51; 52]. Nevertheless, in case of broadcast, *e.g.*, over a lossy indoor environment of WiFi-link, the Packet Error Rate (PER) can increase above the tolerable limit of error-resilient video decoder. In loss recovery techniques, the sender sends an additional information and the receiver then uses this information to recover the erroneous packet, *e.g.*, the authors in [12] have recovered original packet from multiple copies of retransmissions or from cooperating receivers by exploiting the soft information from the PHY layer, while FEC codes are employed in [53; 26] to reduce the PERs for multimedia streaming. Thus avoiding service disruption, which is essential for ensuring optimum application performance across the network.

Recently, alternative robust video decoding techniques based on JSCD [54] have been proposed to recover damaged packets mainly at APL layer. It is a cross-layer approach, where the PHY layer information is passed to the higher layers for performance improvement. Furthermore, in contrast with the standard error correcting codes, the source decoder is used as an error-detection and error-correction code. Combined with a sequential channel decoder, transmission errors in received packets may be detected and corrected, without introducing additional redundancy in the bit-stream.

JSCD techniques use the residual redundancy in the APL layer packets due to the semantic of the source coders [35; 36; 55], and to the fact that compressed data have to be packetized [56]. Residual redundancy may also come from the syntax of variable-length source codes [57; 58; 59]. Furthermore, the protocol stake redundancies are also used by several authors to develop header and packet recovery methods, *e.g.*, robust header estimation technique presented in [8] uses intra and inter-layer redundancy (known fields in headers, presence of Cyclic Redundancy Check (CRC), *etc.*) in the protocol stack to recover the information in the header of a packet corrupted by transmission errors. It is further adapted in [60] for improved decoding of critical data generated by the MPEG-4 AAC audio coder and in [61] for the upper protocol layers (IP and UDP-lite) protected by checksums.

The wireless PHY and MAC layers in the SPSs of WiFi and WiMAX are designed to be as reliable as possible, where one bit error in a packet causes drop of the whole packet. However, due to the error-resilience features of many state-of-the-art multimedia video decoders and the utilization of JPCD strategies at APL layer, packets with errors can still be useful for multimedia applications. Furthermore, for an error-resilient application, distortion in multimedia quality can be decreased by reducing the amount of data loss at a wireless receiver, *i.e.*, by relaying maximum number of error-free and erroneous packets to the multimedia application. It is then up to the APL layer to drop or retain the erroneous packets. Therefore, mechanisms are needed to efficiently support multimedia data transmission over wireless networks by reducing the packet drop and forwarding maximum packets (even partially erroneous) to multimedia applications, where one can deploy robust

decoding and recovery techniques.

1.2 Problem addressed in this thesis

Most of the robust tools, like video decoders using JSCD techniques, are not compatible with the SPS due to several non-compliant requirements:

1. The SPS prevents corrupted packets to reach the APL layer, *e.g.*, current WiFi protocol drops all erroneous packets and WiMAX drops all packets with erroneous headers. In the SPSs of both WiMAX and WiFi standards, the retransmission control policy is used to reduce end-to-end packet loss.
2. They require the exchange of soft information between the protocol layers. The SPSs of both WiMAX and WiFi do not function with the soft information and hamper the exchange of soft information between the channel decoder at PHY layer and the robust source decoder at APL layer.
3. The conventional use of retransmission mechanism in WiMAX and WiFi, where erroneous packet is dropped and is retransmitted, is not compatible with these robust decoders.

The solutions of the above-mentioned requirements are normally sooth through the following modifications to the SPS:

1. The protocol stack is usually made permeable to transmission errors by making it capable of forwarding erroneous packets to the APL layer [1]. Thus, the main objective of lower layers is to relay maximum number of packets to the APL layer. This problem of developing PPS is partially addressed for UDP layer in [2; 3] and for MAC layer in [4; 5; 6], where the ideas of selective error detection are presented. The adoption of selective error detection, where the packet discard is avoided as long as errors do not affect important bits (*e.g.*, header) of a packet, can produce several positive effects on the network. Nevertheless, the packet drops due to erroneous headers can still become significant, especially at high date rates [7] or in situations where payload-to-header ratio is low. This shortcoming can be addressed partially by the receiver based schemes that in addition to ignoring the payload errors can estimate the corrupted header fields [8; 9; 10; 11], thus even packets with erroneous headers can be forwarded to the higher layers.
2. The soft information at PHY layer [12] can reach the APL layer, *e.g.*, by using the transparent layer mechanisms proposed in [13; 14], and thus can be utilized by the robust video decoder. This modification requires changes in the receiver only, and is therefore an applicable solution: it is compliant with the signal which is actually transmitted.
3. A packet received in error needs not be retransmitted unless the robust video decoder at APL layer cannot recover the error.

However, these solutions might be difficult to use under several circumstances. The two important complications, one can face while developing a robust video receiver, are studied in this thesis and are mentioned below.

First, often at a given layer of the SPSs of WiMAX and WiFi standards, small packets are aggregated into larger packets or bursts, which are then forwarded to lower protocol

layers at the transmitter. This packet aggregation is useful in situations where each transmission unit may have significant overhead (preambles, headers, etc) or where the expected packet size is small compared to the maximum amount of information that can be transmitted. Thus, it improves the payload-to-header ratio and boosts the throughput of these wireless networks. Similarly, as proposed in [15; 16], the multiplexer can combine multiple packets into a single multiplexed packet and the access point multicasts the multiplexed packet to the wireless end stations to reduce overhead and increase capacity. However, when transmission over a noisy channel is considered, FS, *i.e.*, recovering the aggregated packets from a burst or multiplexed packet, may become difficult.

At the receiver, without a robust FS, header estimation techniques mentioned in the solution (1) above remain fruitless, as for them locating the headers and their boundaries is compulsory. Furthermore, FS failures at a lower protocol layer can cause loss of several consecutive packets, which could otherwise be relayed to the APL layer for JSCD. Though, in case of *e.g.*, WiMAX-link degradation, the user can request the change of modulation and coding scheme to more robust one, it may not be possible in broadcast scenario. Furthermore, some of the state-of-the-art FS techniques [17; 18] are not compliant with the solution (2), as they work on hard bits. Though, several FS techniques, usually deployed at the PHY layer, can work on the soft output of the channel [19; 20; 21; 22; 23], they require insertion of the synchronization markers (start codes), which require modifications to the transmitter and an additional overhead, thus cannot be allowed in the protocol stacks of WiMAX/WiFi standards.

Second, for improving the performance, these wireless networks, especially WiFi, tend to include a FEC algorithm [24] to avoid link-layer retransmissions, in situations where the consecutive packets are likely to be infected with bursty error, and in broadcasting applications where retransmission is very difficult if not forbidden. Thus, one needs to provide an additional protection to the packets, so that a lost packet can be recovered at the receiver end. Packet-level FEC decoding to recover a lost packet is often performed on the hard bits, thus necessitating the loss of soft information from the soft-output decoders functioning at PHY layer. Therefore, it hinders the flow of soft information to the higher layers, thus preventing the use of the solution (2). For example, the packet-level FEC protocol RFC 5109 [25] describes method to recover a packet from the uncorrupted hard packets. Furthermore, generally packet-level FEC decoding requires error-free packets to fully retrieve the lost or dropped packets [26], which makes JSCD useless by not allowing erroneous packets to reach the APL layer, thus conflicting with the solution (1).

The induction of contention-free multiple access schemes in IEEE 802.11e [27] and the development of the robust JSCD recovery techniques at various protocol layers have almost eliminated the packet drops in the channel and at the protocol layers, respectively. Therefore, instead of the solution (3), the development of a decoding technique, capable of performing FEC decoding on the soft-valued partially corrupted packets, is the need of the time. Thus, for the broadcast services the retransmissions can be disabled completely and replaced with the packet-level FEC.

1.3 Proposals

This thesis intends to study the above two circumstances to develop proposals for S-PPS in order to enable robust reception of the video broadcast over WiMAX/WiFi-link. Often the whole framework is ignored, which is the main reason for the under utilization of the error mitigation tools. The objective of this PhD is to explore the global framework of

an inter-networking scenario of WiMAX and WiFi networks, where broadcast of the video from WiMAX network to the terminal user in WiFi network is analyzed. Any robust tool necessary for the said purpose should be able to meet certain requirements, which facilitate the deployment of the above mentioned solutions, *i.e.*, (1) and (2). The FS and packet-level FEC decoding tools that would be studied in this thesis should take into account the following requirements:

1. In order to propagate soft information to the higher layers, despite other protocol stack functions, the FS should also be able to work on the soft values and should be capable of forwarding soft information to the upper layers. Similarly, for reliable broadcast the packet-level FEC decoder should not hinder the flow of soft information to the APL layer.
2. The robust tools should try to forward maximum number of packets to the APL layer, even packets having erroneous payloads. FS can help to robustly segment the aggregated packets even if the headers of the small packets inside the burst are erroneous. This allows erroneous packets, which are otherwise dropped, to reach the APL layer where they may be correctly understood by JSC decoders. The corrupted payload relayed to the APL layer could *e.g.*, exceeds the tolerable limit of error-resilient video decoder. Therefore, for the S-PPS, one must further reduce the packet loss by using packet-level FEC. Any packet-level FEC should, along with the lower permeable layers, be able to pass maximum number of packets to the APL layer.
3. Several redundancies present in the protocol layer (known fields in headers, presence of CRC, Header Check Sequence (HCS), or checksums, etc.) should be utilized to perform robust FS and to minimize the possibility of dropping a packet. Similarly, few explicit redundancies present inside the packet header should be exploited to improve the performance of the robust packet-level FEC decoder.
4. Any robust tool should not modify the transmitter functionality and should remain compatible with the transmitter's SPS.

In addition to the above-mentioned requirements, the framework should enable the joint use of information between the layers of the S-PPS and the exchange of benefits of its constituent tools. On one side, *e.g.*, a robust JSC decoder deployed at APL layer can benefit from the robust header recovery, FS, and packet-level FEC decoder at lower layers as they increase the number of packets relayed to it. On the other side, the FS and packet-level FEC decoders can benefit from the soft information provided by the PHY layer and from the error-detection capability of several robust source decoders at APL layer.

1.3.1 Part I: Robust Frame Synchronization

Keeping in view the above-mentioned requirements, in the first part of this thesis, we propose several JPCD approaches for FS. They exploit all available information: soft information at the output of the channel (or channel decoder) as well as the structure of the protocol layers to estimate the boundaries of the small packets and the content of their headers.

First, a trellis-based technique for FS is proposed, where the packet aggregation is modeled by a Markov process, which allow representing all possible successions of packets in a burst by a trellis inspired from that of [28]. A modified BCJR algorithm [29] is applied on this trellis to obtain the packet boundaries. Second, a low-delay and reduced-complexity suboptimal version of the trellis-based algorithm is proposed. It uses a *Sliding*

Trellis (ST)-based approach inspired from [30], where a low-latency variant of the BCJR algorithm was presented for the decoding of the CCs. These are *hold-and-sync(hronize)* techniques, which require the whole (for trellis-based) or part (for ST-based) of the burst to perform FS.

Finally, an on-the-fly technique is proposed, which combines robust header recovery technique inspired from [8] with Bayesian hypothesis testing inspired from [19; 20; 21; 22] to localize packet boundaries via a sample-by-sample search. We use a robust 3S automaton, derived from that of [17], but instead of hard CRC correction, a soft header recovery technique [8] for correcting the damaged headers (exploiting all known intra and inter-layer redundancies) is exploited to estimate the length field of the header. Moreover, the Bayesian hypothesis testing, used to search for the correct FS, provides improved performance due to the use of soft channel information combined with *a priori* information due to the redundancy present at the header of a packet.

The FS techniques presented here do not require any signaling overhead, *i.e.*, no synchronization markers are added and only the available information in the protocol layer is utilized. Furthermore, these techniques are quite general, they are illustrated with the synchronization of WiMAX MAC packets aggregated in bursts, which are transmitted to the PHY layer [32], but they are easily extendable to other protocols where packet aggregation is performed.

1.3.2 Part II: Robust Packet-Level FEC Decoding

To broadcast the multimedia packets over WiMAX/WiFi-link, a packet-level FEC scheme is analyzed to overcome the retransmission delays. Taking into account the requirements mentioned earlier, instead of performing decoding on hard bits, the soft information forwarded by lower layers is used to recover erroneous packets. The packet-level FEC decoder is deployed at the RTP layer, where it is assumed that the RTP packets reaching the FEC decoder are soft-valued and can have errors. The same idea of JPCD as deployed in FS techniques is put at work to develop a packet-level Maximum *A Posteriori* (MAP) decoder to estimate the erroneous packets, utilizing the RTP header redundancies, the redundancy introduced due to use of FEC, and likelihood from the channel. More notably, it causes no hindrance to the flow of soft information from the lower layers, through the RTP layer, to the higher layers. Moreover, the robust decoder presented here needs no side information and remains completely compatible with RFC 5109.

Though the use of FEC redundant packets would decrease the system goodput, but given that the retransmission is disabled, one can utilize this spared goodput for the transmission of the FEC redundant packets. Based on the channel condition and service nature (real-time video, data, voice, etc.), transmitter can decide for each application or service flow, either to use retransmission or FEC scheme. Transmitter switches to FEC scheme if the channel condition falls below a certain level or in situations when retransmission is causing goodput loss or unacceptable delay.

The FEC protection and the JPCD-based FEC decoder presented in this thesis are well tuned for RTP layer, but they can be extended to the other layers of the S-PPS. Furthermore, they remain compatible with other broadcast scenarios, *e.g.*, of receiving the Mobile TV over DVB-H (Digital Video Broadcasting - Handheld) [33; 34] and then rebroadcasting it over the WiFi network.

Robust tools presented in this thesis, significantly reduce the amount of packets that need to be dropped and enable flow of *soft-packets* (s-packets), which may contain errors,

to the upper layers and enhance the performance of the robust tools functioning at higher layers. They can then be forwarded to the APL layer using the PPS techniques presented in [13; 14; 8], and robustly decoded using JSCD techniques [35; 36].

1.4 Thesis Outline

The remainder of this thesis is organized as follows:

Chapter 2: This chapter first explains an SPS based inter-networking scenario of WiMAX and WiFi networks. It then details the required modifications needed to develop S-PPS, which can enable flow of soft and corrupted information to higher protocol layers. The global framework presented will further motivate the need for the proposals presented in the coming chapters. The inter-networking scenario with S-PPSs, presented in this chapter, is considered throughout this thesis.

1.4.1 Part I: Robust Frame Synchronization

Chapter 3: An introduction to the FS is presented in this chapter. We briefly present the related work on the FS techniques from the literature. We explain the fixed-length state-of-the-art FS techniques [38; 39] and the variable-length state-of-the-art FS techniques [19; 20; 21; 40; 22; 17].

Chapter 4: This chapter presents a trellis-based technique for FS, which is compatible with the S-PPS.

Chapter 5: In this chapter, we show how the FS technique of Chapter 4 can be modified to develop a low-delay and reduced-complexity suboptimal variant. We study its impact on the performance and complexity of the FS.

Chapter 6: In this chapter, we take the best out of the state-of-the-art FS techniques presented in Chapter 3 and fine-tune them to propose the 3S automaton for FS. This extension aims to provide a FS technique that can work on-the-fly, unlike the trellis-based and ST-based FS techniques.

1.4.2 Part II: Robust Packet-Level FEC Decoding

Chapter 7: We present an introduction to the packet-level FEC and then provide an evaluation of the feasibility of the retransmission and the packet-level FEC schemes. It further illustrates why packet-level FEC is important and interesting for the broadcasting scenario.

Chapter 8: The proposed packet-level FEC decoder is presented in this chapter. This packet-level FEC decoder is applied to the RTP layer, where FEC packets are generated using the RFC 5109 protocol.

Chapter 9: We discuss the conclusions of our thesis in this chapter and also provide some possible applications of the FS techniques and the packet-level FEC decoder presented in this thesis.

Five appendices have been attached to this thesis. Appendices A.1 and A.2 are references for the structure of the PHY frame and MAC packet in WiMAX, respectively. Appendix A.3 details the fields of the UDP header, Appendix A.4 details the fields of the UDP-lite header, and Appendix A.5 details the fields of the RTP header.

Chapter 2

Scenario Under Investigation

WLANs and WMANs have experienced a tremendous surge in popularity in recent years. The WiFi has established itself as the worldwide standard for WLANs, while WiMAX is slowly gaining its place as a long distance service providing WMAN standard. Integrating WiMAX and WiFi promises convenient and affordable broadband connectivity, where both networking technologies mutually benefit from each other. The said integration brings new deployment models for the service providers.

WiFi hotspots are connected to the Internet through a wired connection (*e.g.*, Ethernet), and therefore have high deployment costs, particularly in remote rural or suburban areas with low population densities. Therefore, it is necessary to develop new schemes capable of providing sufficient bandwidth to meet the enormous access requirements of WiFi Subscriber Stations (SSs) while simultaneously reducing the backhaul cost. It has been suggested that WiMAX represents a promising solution for providing WLAN hotspots with backhaul support [62].

WiMAX extends the benefits of WiFi networks to deliver the next-generation mobile services, but unfortunately, it shares the same issues as 3G and 4G cellular, *i.e.*, limitation of data capacity of each cell site and poor indoor coverage. Considering the Non-Line-Of-Sight (NLOS) features of WiMAX brought by high frequencies, most operators select the frequency band of 2-6 GHz to realize wireless broadband access. However, higher frequencies mean poorer ability to diffract or bend around obstacles, and greater penetration loss, thus WiMAX suffers from poor in-door coverage. The poor coverage affects the quality of high-speed data services inside a building leading to a poor user experience. Very recently, WiMAX femtocells [63] offer several compelling benefits including better indoor coverage, but often most users connect to Internet through the WiFi technology, thus WiFi users accessing Internet through WiMAX network is an ideal solution for the Internet service providers.

2.1 Introduction

The combination of WiFi and WiMAX may be an attractive solution to wireless broadband access, which enables two protocols to inter-work with each other in many aspects. One of the most significant factors to motivate such integration is supporting "personal" hotspot services. For example, consider the scenario where a group of users are currently within a building, but will shortly leave the premises and travel to another destination with low population density, *e.g.*, a rural or a suburban area. Imagine also that the users are

currently using the Internet services and wish to remain connected to the network when they reach to their destination. To support this personal hotspot requirement, each user can utilize a terminal device equipped with an upgraded WLAN card (*i.e.*, WiFi Receiver) to access a nearby hotspot device (called WiBOX device). When the users leave the building, they can simply take the WiBOX with them. The WiBOX will continue to provide the Internet services through the backhaul service provided by the mobile WiMAX network. Note that the demand for personal hotspot services such as that described in this scenario is expected to grow significantly in the near future to accommodate the requirement for ubiquitous network environments.

The principal advantage of the WiBOX solution investigated in this study is that WiFi SSs located within the coverage area of a WiBOX can access the backhaul service provided by the local WiMAX network. In other words, no additional WiMAX interface is required for the SSs with traditional WLAN cards to access the services provided by the WiMAX network, and thus the hardware and Internet access costs are significantly reduced.

This chapter aims to provide the detailed review of the inter-networking of WiFi and WiMAX, where we consider the scenario of providing video broadcast services through Internet, *e.g.*, IPTV, to the end users. Under the considered scenario, WiBOX is able to receive video broadcast over WiMAX-link, and is capable to rebroadcast the video to the indoor users through WiFi-link. WiBOX receives the video packets transmitted by the WiMAX Base Station (BS) using WiMAX protocol [32] and relays them to WiFi SSs using WiFi protocol [41], by taking into account the QoS requirements of the different streams.

The stress is put on the down-link (DL) rather than the up-link (UL) since this thesis is meant to study the broadcast to the end users. Therefore, with respect to the motivations of this thesis, the inter-networking scenario investigated is intended to:

1. provide realistic analysis of how the deployment of JSCD is possible at the SS,
2. study the packet loss through the whole protocol stack by keeping in view the global framework and to find means to reduce it, and
3. select the best possible tool necessary to facilitate the robust multimedia communication, while remaining standard compatible at the receiver.

The inter-networking scenario presented in this chapter would be considered throughout this thesis. The remainder of this chapter is organized as follows: An introduction to WiMAX and WiFi is presented in Section 2.2 and 2.3, respectively, before explaining the scenario under consideration in Section 2.4, where the descriptions of the SPSs of WiMAX BS, WiBOX, and WiFi SSs are provided. Section 2.5 suggests the required modifications needed in the SPSs to meet the requirements mentioned in the introduction of this thesis (see Chapter 1).

2.2 Introduction to WiMAX

In this section we discuss features of WiMAX standard [32; 64; 65; 66] essential for the exposition of the ideas in this thesis. WiMAX provide a robust, reliable, and cost-effective means to deliver broadband services in metropolitan and rural areas. It is also known as IEEE 802.16 and is intended for last-mile wireless broadband services. In a WMAN, WiMAX can provide Broadband Wireless Access (BWA) up to 30 miles (50 km) for fixed stations, and 3 - 10 miles (5 - 15 km) for mobile stations. This is in contrast to the WiFi WLAN standard, which is limited in most cases to only 100 - 300 feet (30 - 100m). The

most outstanding advantage of WiMAX is its low cost for installation and maintenance compared with the traditional wire or fiber network access, especially for those areas that are too remote or difficult to reach. Networks could be created in a short time by deploying a small number of BSs on buildings or poles to create high-capacity wireless access systems.

WiMAX supports very robust data throughput. The technology theoretically could support approximately 75 Mbps per channel. The service across a single channel could be shared by multiple customers. One of the main strength of WiMAX is to provide several QoS in a deliberate fashion to offer different bandwidth capabilities to customers with different needs (and different budgets). Mobile WiMAX capabilities on a per customer basis will be lower in practical terms, but much better than competing 3G technologies. In practical terms, WiMAX industry intends to deliver service at 2 Mbps to 4 Mbps to its customers with Mobile WiMAX.

The WiMAX standard defines the technical features of the communications protocol. It defines two possible network topologies, *i.e.*,

1. PMP (Point-to-Multipoint) topology
2. Mesh topology

The main difference between the two modes is the following: in the PMP mode, traffic may take place only between a BS and its SSs, while in the Mesh mode the traffic can be routed through other SSs until it reach BS and communication can even take place between SSs. PMP is a centralized topology where the BS is the center of the system, while in Mesh topology it is not. In the considered broadcast scenario, PMP mode is of interest.

The WiMAX standard includes the following two main duplexing techniques

1. Time Division Duplexing (TDD)
2. Frequency Division Duplexing (FDD)

The choice of one duplexing technique or the other may affect certain PHY parameters as well as the features that can be supported.

In a FDD system, the UL and DL channels are located on separate frequencies. A fixed duration frame is used for both UL and DL transmissions. It allows simultaneous use of both full-duplex SSs, which can transmit and receive simultaneously and half-duplex SSs, which cannot. A full-duplex SS is capable of continuously listening to the DL channel, while a half-duplex SS can listen to the DL channel only when it is not transmitting on the UL channel.

In the case of TDD, the UL and DL transmissions share the same frequency but they take place at different times. A TDD frame has a fixed duration and contains one DL and one UL sub-frame. The frame is not necessarily divided into two equal parts. The TDD framing is adaptive, *i.e.*, the bandwidth allocated to the DL versus the UL can change. The split between the UL and DL is a system parameter and the WiMAX standard states that it is controlled at the MAC layer of the BS. TDD mode is the most frequently used mode of duplexing due to its simplicity and is therefore considered in this thesis.

The WiMAX standard comprises of multiple PHY layers and a single MAC layer. It supports multiple PHY layer specifications due to its modular nature. The first version of the standard only supported Single Carrier (SC) modulation. Since that time, OFDM and scalable OFDMA have been included to operate in NLOS environment and to provide mobility. However, the PHY and MAC definitions are quite closely coupled and consequently, it is difficult to make a clear separation between them in the model. Nevertheless, we will now briefly explain each one of them.

2.2.1 WiMAX MAC Layer

A network that utilizes a shared medium shall provide an efficient sharing mechanism. Here the medium is the wireless channel, and the MAC layer provides the core functionality of system access, bandwidth allocation, connection establishment, and connection maintenance. QoS is insured to the transmission and the scheduling of data over the PHY frame is performed.

In the WiMAX standard, PHY frame is divided in to DL and UL sub-frames for TDD mode. DL sub-frame is separated from UL sub-frame by Transmit Transition Gap (TTG) and the gap between the UL sub-frame and the subsequent DL sub-frame is called Receive Transition Gap (RTG), as shown in the figure 2.1.

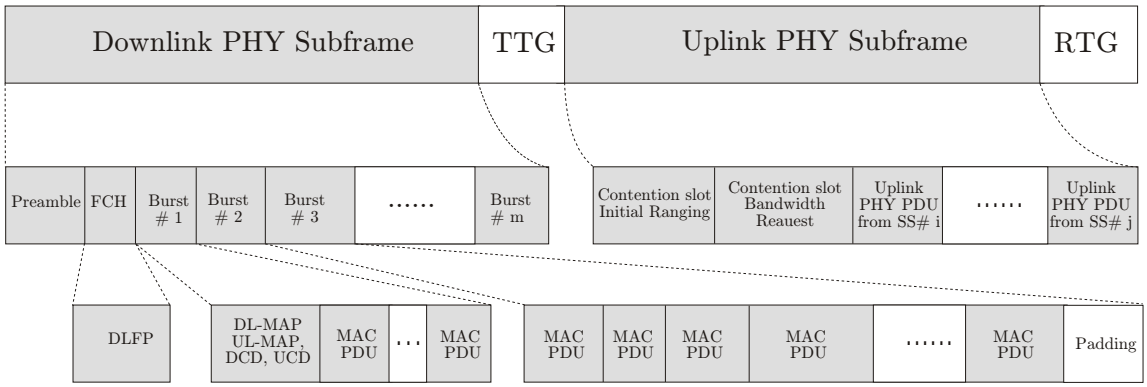

Figure 2.1: WiMAX PHY Frame

The scheduler manages the DL and UL sub-frames and controls how much resources are allocated in each direction. DL sub-frames begin with a frame control section that contains Frame Control Header (FCH) and the DL map (DL-MAP) for the current DL sub-frame as well as the UL map (UL-MAP) for a frame in future, see appendix A.1 for other fields of DL and UL sub-frames. The FCH contains, among other fields, the BS ID (4 bits), the frame number (4 bits), as well as an 8-bit CRC, which is used to detect errors in the FCH. The DL-MAP informs all SSs of which part of the current DL sub-frame they should listen to. The UL-MAP informs SSs of their transmission opportunities as a response to their dynamic bandwidth requests, or on the basis of prior service agreements. These are then followed by the transmission of the DL sub-frame and the UL sub-frame.

The DL sub-frame is divided into Time Division Multiplex (TDM) portions, which are so-called *burst* profiles: each burst profile is characterized by the use of a particular modulation and coding scheme and can contain multiple concatenated fixed-length or variable-length packets or fragments of packets received from the higher layers. Each burst profile has an associated FCFS (First Come First Served) queue; each queue is checked to see if there are packets to transmit for that burst profile. Each burst is filled with several MAC packets in order, until there is not enough space left. Padding bytes (0xFF) are added [32] at the end of the burst. Lengths of bursts are assigned by the scheduler and are communicated to users in the DL-MAP. Each burst is modulated and coded at the PHY layer before transmission in the DL PHY sub-frame. In what follows, we consider only the DL scenario and thus only the DL sub-frame.

At the receiver side, SS do not have to decode all bursts, but only the relevant one and as the burst profile has already been negotiated while making connection between BS

and SS, SS knows which burst it has to decode. Burst transmission enables SS to save power by putting receiver in sleep mode during off-burst interval. Thus, each SS reads its relevant burst, performs the PHY layer operation, and retrieves its MAC packets after reading their headers. Errors left by the PHY layer can affect the isolation of the MAC packets within a burst, thus an effective FS is required to handle the residual errors.

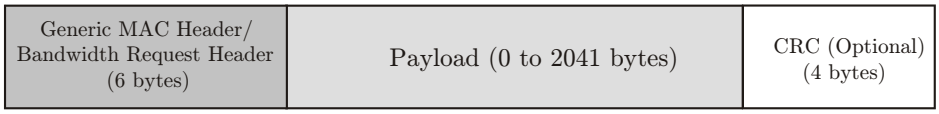

Figure 2.2: Typical WiMAX MAC PDU

The structure of a typical WiMAX MAC Protocol Data Unit (PDU) is illustrated in Figure 2.2. Each MAC PDU begins with a fixed-length header of $\ell_h = 48$ bits, followed by a variable-length payload and an optional CRC. The header can be of two types, one is the Generic MAC Header (GMH) containing either MAC Management messages or higher layer data and the other is the Bandwidth Request Header (BRH) used to request additional bandwidth (see appendix A.2). As we are considering only the DL case, where the connection is already established, MAC PDUs belonging to a burst contain only CS data, so only the GMH is possible inside a burst. GMH is shown in Figure 2.3.

See appendix A.2.1, for the detail description of each field of the GMH. For the sake of simplicity it is assumed that

- No CRC is used (the CRC is mandatory only for MAC management messages in OFDM and OFDMA).
- No ARQ is used.
- No packing and fragmentation is used. The packing sub-header allows the transmission of multiple Service Data Unit (SDU) fragments in a single MAC PDU.
- No Encryption is used for payload.
- No extended sub-header is present.

Some fields are already fixed in a MAC header, but with the considered assumptions, fields such as Header Type (HT), Encryption Control (EC), sub-headers, and special payload types (Type), Reserved (Rsv), CRC Indicator (CI), and Encryption Key Sequence (EKS) remain constant. The LEN field, representing the length in bytes of the MAC PDU including the MAC header, has a variable content. Connection IDentifier (CID) represents the identifier of the connection of the receiver and is a variable field. The HCS, an 8-bit CRC, is used to detect errors in the header and is also a variable field. It is a CRC-8 with a generator polynomial $D^8 + D^2 + D + 1$ and protects the content of all header fields.

2.2.2 WiMAX PHY Layer

Following five PHY interfaces are defined in the WiMAX standard.

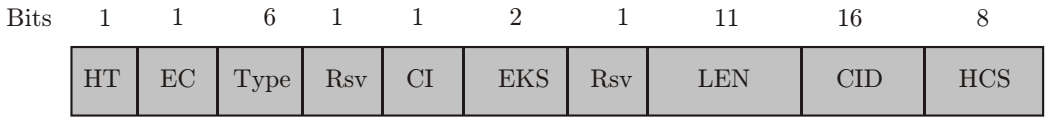

Figure 2.3: GMH as specified in IEEE 802.16-2004[32]

1. WirelessMAN-SC (using SC)
2. WirelessMAN-SC (using SC)
3. WirelessMAN-OFDM (using OFDM transmission)
4. WirelessMAN-OFDMA (using OFDM transmission, and OFDMA)
5. WirelessHUMAN

The use of OFDM increases the data capacity and, consequently, the bandwidth efficiency with respect to classical SC transmission. This is done by having carriers very close to each other but still avoiding interference because of the orthogonal nature of these carriers. In OFDMA, the subcarriers are divided into subsets of subcarriers, each subset representing a subchannel. The standard indicates that the OFDMA symbol is divided into logical subchannels to support scalability, multiple access, and advanced antenna array processing capabilities. In the DL, a subchannel may be intended for different receivers or groups of receivers, while in the UL, a transmitter may be assigned one or more subchannels. The subcarriers forming one subchannel may be adjacent or not.

The basic idea being exploited in OFDM and OFDMA is to divide the symbol stream into sub-streams to be transmitted over a set of orthogonal subcarriers where resulted symbol period in each sub-stream is much larger than the multipath channel delay spread to efficiently mitigate the intersymbol interference. In practice, OFDM can be efficiently implemented using the well-known FFT and IFFT functions. The supported FFT sizes are 2048, 1024, 512 and 128.

The OFDM PHY layer model would be considered hereafter due to its frequent use and implementation simplicity. An OFDM symbol is made up of carriers; the amount of carriers determines the FFT size used. There are several carrier types:

- Data carriers for data transmission.
- Pilot carriers for different estimation purposes.
- Null carriers for guard bands and DC carrier.

The WiMAX transmitter consists of the following blocks: randomizer, FEC encoder, interleaver, and modulator. They are applied in this order at transmission. The corresponding operations at the receiver are applied in reverse order.

Randomizer is used prior to FEC encoding. Randomization decorrelates the bits to be encoded and avoids long runs of zeros or ones to be forwarded to the encoder. In the OFDM PHY layer, the randomizer uses the Pseudo Random Binary Sequence (PRBS) generator $g(x) = 1 + x^{14} + x^{15}$.

The WirelessMAN-OFDM PHY layer supports two mandatory FEC schemes: RS only and RS concatenated with block CC. The RS code is a systematic code generated from Galois Field, GF(256) with information block length variable from 6 to 255 bytes. Its error correction capability varies from 0 to 16 error bytes. The concatenated coding scheme uses CC as an inner code. The standard also supports turbo block codes as an optional FEC scheme. FEC is essential for OFDM systems since it compensates for the bit errors that are inevitable in times of deep fade in the channel.

After FEC encoding all encoded data bits shall be interleaved by a block interleaver, after which modulation is carried out. The standard supports four different modulation schemes. It supports higher order 16-QAM and 64-QAM schemes to maximize link throughput and also supports BPSK and QPSK for robustness and reliability.

Finally, a 256 point FFT OFDM is done. By using 256 point FFT OFDM, the symbol duration is prolonged therefore it is very tolerant of the long multipath delays that occur in long-range NLOS operation.

2.3 Introduction to WiFi

User demands to access multimedia applications through wireless medium has made WiFi, also called IEEE 802.11, as their ultimate choice. A WLAN based on WiFi shares the medium at all times between UL and DL flows and is inherently a distributed environment. It also allows different operational transmission rates for each station.

WiFi MAC features two mode of operations: Distributed Coordinating Function (DCF) and Point Coordinating Function (PCF). DCF is CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) access protocol that uses random back-off to avoid collision. PCF provide centralized scheduled access to channel. It comprises of the chain of Contention Free Period (CFP) and Contention Period (CP). DCF rules are followed in the CP. In the CFP Point Coordinator (PC) polls the node one by one and grant access to channel. New stations that need to get enrolled in poll list, send request in CP.

High collision rate and frequent retransmission in wireless channel will cause unpredictable delays and jitters, which degrade the quality of real-time voice and video transmission. IEEE 802.11e is an improved amendment to the IEEE 802.11 standard that defines a set of QoS enhancements for WLAN applications through modifications to the MAC layer in order to support QOS requirements of time-sensitive applications such as voice and video by introducing priority mechanisms.

The first improvement is the introduction of enhanced channel access mechanisms, namely, Enhanced Distributed Channel Access (EDCA), which is a contention-based channel access, enriching the existing DCF, and the Hybrid Coordination Function Controlled Channel Access (HCCA), a controlled channel access, which improves upon PCF. These two entities are managed by a centralized controller called Hybrid Coordinator (HC), which resides in the QoS-Access Point (QAP). Each of them would now be explained.

2.3.1 EDCA

EDCA provides a mechanism whereby traffic can be prioritized but it remains a contention based system and therefore it cannot guarantee QoS. With EDCA, high priority traffic has a higher chance of being sent than low priority traffic: a station with high priority traffic waits a little less before it sends its packet, on average, than a station with low priority traffic. This is accomplished by using a shorter contention window (CW) and shorter arbitration inter-frame space (AIFS) for higher priority packets.

2.3.2 HCCA

The HCCA adopts a different technique by using a polling mechanism. Accordingly it can provide guarantees about the level of service it can provide, and thereby providing a true QoS level. HCCA is a centralized access mechanism controlled by the HC. Data packets belonging to different data, voice, and video streams having common traffic characteristics are grouped in classes of traffic called Traffic Streams (TSs). The provision of QoS is done by defining the traffic characteristics. Each QoS enabled 802.11e station (QSTA) may establish up to eight HCCA TSs. TSs provide a parameterized QoS access to the medium and can be either uni-directional (*i.e.*, UL or DL) or bi-directional. The QSTA is able to gain access to a radio channel for a given number of packets. The channel is release only after these given number of packets have been sent.

HCCA functions as follows: HC takes the control of the channel. Once it has taken control it polls all the QSTAs or transmitters in the network. To do this it broadcasts

particular frame indicating the start of polling, and it will poll each QSTA in turn to determine the highest priority. It will then enable the transmitter with the highest priority data to transmit, although it will result in longer delays for traffic that has a lower priority. The HC maintains a centralized schedule that is based on the QoS requirements of all of its registered QSTAs. Then, the QAP notifies each of the QSTAs about the time it will have access to the wireless medium. Since this process is managed from a central location, it is guaranteed that the access will be contention-free.

The following transmission parameters are sent to the QSTA at the polling time:

- Service Interval (SI): the time interval between two successive polls of the SS.
- Transmission Opportunity (TXOP): the SS transmission duration based on the mean application data rates of its TSs.

SI and TXOP are the basic parameters used by the scheduling algorithms to manage the access to the medium and their choice is the key in the QoS provisioning. During TXOPs, the medium is accessed by only one QSTA. More specifically, a DL TXOP consists of a burst of QoS Data (data, for short) packets transmitted from the QAP to a QSTA. As polls or TXOPs are addressed to a QSTA and not to an individual stream, the QSTA must ensure fairness in serving individual streams emanating from it.

EDCA function helps to reduce the probability of collision by varying the CW. Although the EDCA function may provide satisfactory service differentiation in low-load environments, its contention based nature results in impaired performance, low channel efficiency, and lack of QoS guarantee in the presence of heavy traffic load [67]. On the other hand, the HCCA function is designed to always meet the negotiated QoS requirements of the admitted TSs. For example, in [68] it is demonstrated that the scheduled access scheme, *i.e.*, HCCA, outperforms the contention-based prioritized services, *i.e.*, EDCA, while the problem of video transmission over HCCA is considered in [69]. Therefore, due to the guaranteed collision-free access to the medium for the different TSs, HCCA is considered in this thesis, as to make sure that almost no packet is dropped in the channel.

2.4 WiMAX-WiFi Inter-networking Scenario

In the inter-networking under investigation in this study, the WiMAX system provides broadband wireless access to the WiBOX devices in a PMP topology, as a result, the WiMAX network provides a backhaul service connecting the dispersed WiFi hotspots to the WiMAX BS through WiBOX devices. The topology consists of a WiMAX BS, multiple WiMAX SSs, multiple WiBOX devices as QAPs, and multiple WiFi SSs, as shown in Figure 2.4.

In the WiFi-link each WiBOX device has multiple WiFi SSs, communicating through the same wireless channel, while in the WiMAX-link a single WiMAX BS serves both WiMAX SSs and WiBOX devices within its coverage area. The WiBOX device serves as a relay node as well as a QAP for the WiFi SSs, which are 802.11e QSTAs. The connection between the BS and the WiMAX SS is dedicated to a single user. However, the connection between the BS and each WiBOX is shared amongst all the WiFi SSs within the WiFi hotspot served by the WiBOX.

The MAC layer, as defined in WiMAX standard, enables the WiMAX BS to assign available bandwidth to both WiMAX SSs and WiBOX devices, thus enabling the coexistence and inter-working of WiMAX and WiFi technologies within a single integrated network. Now, we will provide detail description of the SPSs of WiMAX BS, WiBOX, and WiFi SS.

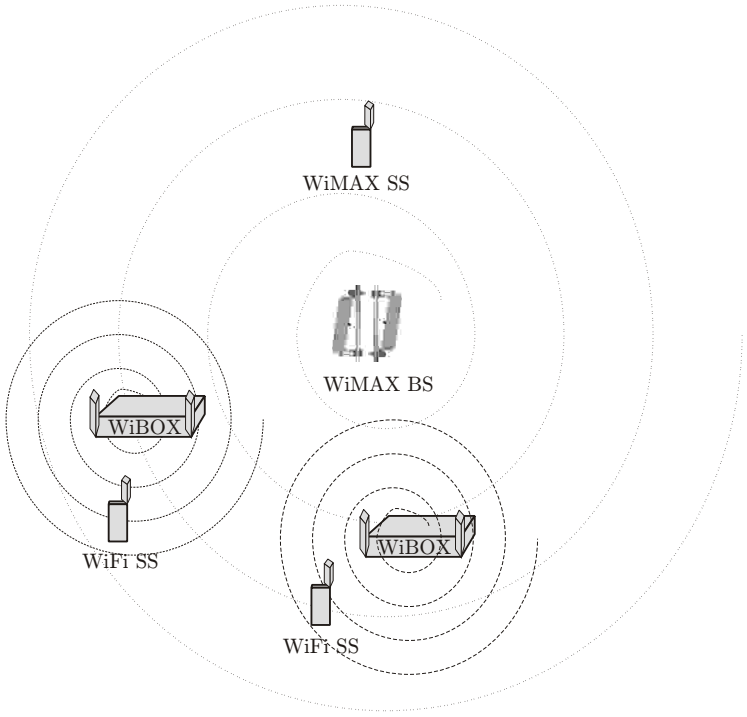

Figure 2.4: System Architecture: WiMAX-WiFi Inter-networking Scenario

2.4.1 SPS of WiMAX BS

In WiMAX, BS has the responsibility to centrally control the SSs, meaning that the SSs have to communicate with each other through BS. An application generates the APL layer packets consisting of data payload and application header. Whenever a packet is passed down to the next protocol layer, a header associated with that layer is added, as shown in Figure 2.5. The Real-Time Transport Protocol (RTP) [70] packet is encapsulated in the User Datagram Protocol (UDP) [71] packet, which in turn is encapsulated in the Internet Protocol (IP) [72] packet.

The IP packet reach at the WiMAX BS through wired link, and the MAC packet header is appended to the IP packet. BS then aggregates several MAC packets belonging to the same burst profile into the corresponding burst of the DL PHY sub-frame. We assume that, before aggregation, the size of each MAC packet is independent from the previous one.

In this stack, RTP protocol provides end-to-end network transport functions suitable for applications transmitting real-time data, such as audio and video. UDP and IP provide source IP addresses, destination IP addresses, and port numbers of the communication pair to ensure correct delivery. Packets are dropped at the IP layer due to congestion or route disruption. On the other hand, the MAC/PHY layers defined by WiMAX standard have to deal with bit errors. Any bit error within a packet could result in the whole packet being dropped, even though the errors could be corrected at the APL layer.

A scheduling decision in the WiMAX BS MAC layer is done once per PHY frame. A scheduling decision assigns time slots to WiMAX user. Once the decision has been made for the entire PHY frame, it is broadcasted by the WiMAX BS to all the users.

2.4.1.1 Video over WiMAX

In the envisaged scenario, the video contents available from the video server are passed to the WiMAX BS through Internet backbone, where WiMAX BS carries out all scheduling and resource allocation in order to efficiently use the network. BS is directly connected with the video server using a wired link, whereas SSs are connected with the BS through a wireless link. Let us consider an H.264 encoder that consists of a Video Coding Layer (VCL), which performs all classic signal processing tasks and generates bit-strings containing coded macro-blocks, and a Network Adaptation Layer (NAL), which adapts those bit-strings in a network friendly way. The basic element for decoding is called NAL Unit (NALU) [50]. RTP is the most often used transportation protocol for the real-time multimedia applications, it provide support to audio and video streams, *e.g.*, see RTP format for H.264 [73]. The NALUs are placed in the payloads of the RTP packets.

To determine the size of each MAC packet, assume that H.264 video NALUs are RTP, UDP, and IP packetized before they reach the WiMAX BS MAC layer. The headers of each of these layers have following sizes: 12 bytes for RTP, 8 bytes for UDP, and 20 bytes for IP. Hence, the total length of RTP/UDP/IP headers is 40 bytes. Thus, the minimum possible size of the MAC packet is 47 bytes, which includes the MAC header of 6 bytes and at least 1 data byte. For WiMAX though a maximum MAC PDU size is 2047 bytes [32], but the default Maximum Transmission Unit (MTU), *i.e.*, the size (in bytes) of the largest protocol data, for IPv6 packet over the WiMAX-link should be 1500 bytes (Ethernet MTU size) [74]. While, the WiFi MTU for MAC layer is 2304 bytes [75]. Thus, in order to avoid split/recombination on the network interface the maximum MAC payload size is kept at 1500 bytes.

One can deploy two strategies to avoid fragmentation and aggregation at IP/MAC layers. First, the fixed-length NALUs can be generated by the encoder at the cost of performance reduction and overhead depending on the NALU size. One NALU is placed per RTP packet. The packet size after adding RTP/UDP/IP headers should not exceed 1500 bytes limit. Second, keeping the MTU limit in view, the variable-length NALUs are aggregated or fragmented into a single or multiple RTP packets, respectively. For example, in H264/AVC, if after RTP/UDP/IP headers overhead the packet exceed 1500 bytes limit, NALUs generated by NAL can be fragmented into several RTP packets [73]. Similarly, if NALUs are very small, several NALUs can be aggregated in a single RTP packet (see ref [73] for more detail on fragmentation and aggregation of NALUs in RTP payload).

2.4.2 SPS of WiBOX

The SPS of WiBOX consists of the layers from WiMAX and WiFi standards and is shown in Figure 2.6. The operation of each layer is explained as follows.

2.4.2.1 802.16 PHY Layer

Conventionally, the hard-output demodulator, interleaver, channel decoder, and derandomizer are deployed at the PHY layer. Hard decoding provides an estimation of the binary sequence that was sent by the BS. The CRC present in the FCH of the DL sub-frame protects the contents of FCH. CRC decoder is required at the receiver to check whether the content of the decoded FCH is correct or not. Received PHY frame with damaged FCH is discarded. After DL-MAP decoding, the receiver can decide which portion (or which burst) of the DL PHY sub-frame it should read.

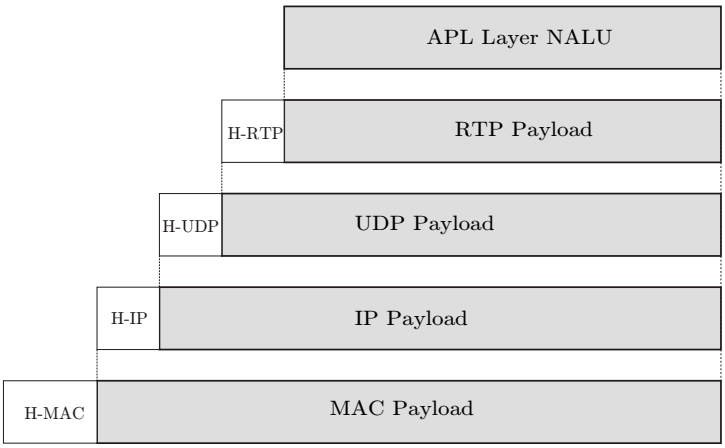

Figure 2.5: Protocol stack for video transmission

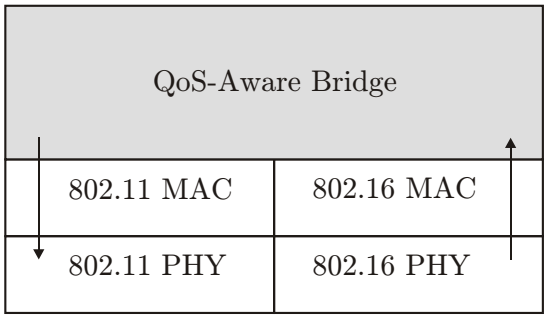

Figure 2.6: SPS of WiBOX

2.4.2.2 802.16 MAC Layer

After reading the DL-MAP of WiMAX PHY frame, the relevant burst is passed to the MAC layer, which extracts the MAC packets corresponding to the WiFi SS using the CID field present in the MAC header. The length field present in the MAC header is used to extract the MAC packet from the burst. Furthermore, HCS field present in the MAC header covers only the header part of a packet and not the payload. Packets with erroneous headers are thus dropped. If the PHY layer channel decoding is unable to remove all errors, the residual errors can cause loss of several MAC packets. A single packet drop can cause misalignments, which can in turn cause loss of the remaining burst, which is intolerable for the multimedia applications.

WiMAX supports automatic retransmission requests (ARQ) at the MAC layer. ARQ-enabled connections require each transmitted packet to be acknowledged by the receiver; unacknowledged packets are assumed to be lost and are retransmitted. WiMAX also optionally supports hybrid-ARQ, which is an effective hybrid between FEC and ARQ.

2.4.2.3 QoS-Aware Bridge

Generally speaking, when constructing integrated WiMAX/WiFi networks, one of the most challenging issues facing network designers is that of designing efficient MAC layer

protocols to optimize the QoS between the WiMAX and the WiFi components of the architecture. Obviously, the overall operational principles of WiMAX and WiFi are quite different, particularly their bandwidth access and QoS provisioning mechanisms. WiMAX systems generally utilize bandwidth more finely than WiFi systems. Furthermore, connection-oriented bandwidth allocation approaches tend to provide a more predictable QoS. Therefore, it is reasonable to expect WiMAX technologies to provide a better QoS than their WiFi counterparts. Since WiFi and WiMAX use different operational protocols in their bandwidth access mechanisms, it is necessary to embed additional layer, which can map the WiMAX QoS requirements to corresponding WiFi QoS priorities. QoS-Aware Bridge layer lies on the top of the MAC Layer and functions as a bridge between WiFi and WiMAX networks.

Several researchers have recently proposed QoS provisioning mechanisms for integrated WiMAX/WiFi systems [76; 77]. For example, in [77], the authors have proposed a QoS framework for 802.16/802.11 inter-networking applications allowing to map the QoS requirements of an application originating in WiFi network to WiMAX network. However, the mechanisms required to satisfy the QoS requirements (*e.g.*, bandwidth assignment, scheduling, and admission control) were not considered. Similarly, in [76], a QoS control protocol is presented to support an integrated QoS, but an implementation of the proposed QoS provisioning mechanism requires a major rework of the WiMAX and WiFi control protocols.

In [78] two bridging solutions for a WiFi/WiMAX interconnection are proposed, by taking into account two main goals: traffic priority and implementation issues. The first solution is more based on the concept of maintaining a certain end-to-end QoS level. The second solution is more devoted to the reduction of the implementation complexity at the cost of no QoS assurance and limiting the protocol adaptation between the two stacks.

A technique of embedding a WiMAX PDU over WiFi has been investigated in [79], while going a step ahead, the work in [80] proposes an efficient and unified connection-oriented architecture for integrating WiMAX and WiFi technologies in broadband wireless networks. A modified convergence MAC layer of WiFi interfaces is designed by embedding the WiMAX subscriber MAC function within the original WiFi MAC. A seamless interoperability between WiFi and WiMAX is presented in [81], where two scenarios, *i.e.*, (i) applications originating in the WiFi domain and terminating in the WiMAX domain and (ii) applications originating in the WiMAX domain and terminating in the WiFi domain, are considered.

2.4.2.4 IEEE 802.11 MAC Layer

The QoS provisioning mechanisms provided by 802.11e can be used to provide multiple access scheme for WiFi-link. Access to the medium is controlled by the QAP, in our case the WiBOX. EDCA bandwidth contention protocol for accessing the channel has been used in [80]. When EDCA is applied, the number of contentions between traffic flows within the same priority class increases, and hence the average end-to-end delay also increases. Furthermore, as the MAC layer in WiMAX is TDMA based, for closeness of channel access schemes, the HCCA mechanism is the preferred strategy for the interoperability between 802.11e and WiMAX [81].

2.4.3 SPS of WiFi SS

Each layer of the SPS of WiFi SS has distinct operations to perform, we will now provide the functionality of each layer.

2.4.3.1 IEEE 802.11 PHY Layer

At the PHY layer, a CRC protects the header fields. Received packet with damaged header is discarded. The payload of the PHY packet, having error-free header, is forwarded to the MAC layer.

2.4.3.2 IEEE 802.11 MAC Layer

At the MAC layer, a CRC protects both the header and the payload of a packet. Current WiFi MAC implementation forces a receiver station to discard every erroneous packet and a simple retransmission control policy is used to reduce end-to-end packet loss. The retransmission increases as the channel conditions worsen. If many retransmissions are employed to reduce losses, the positive effects of receiving more packets are undone by the consequences of higher end-to-end delays that reduce the communication interactivity.

2.4.3.3 UDP Layer

UDP is a real-time video and audio streaming protocol and is designed to handle occasional lost packets, so only slight degradation in quality occurs, rather than large delays if lost packets were retransmitted. At the UDP layer, a checksum protects the header and the payload of the UDP packet. The UDP layer provides no guarantees to the upper layer protocol for message delivery, as after UDP checksum calculation the erroneous packet is dropped.

Many network applications may be running on the same user. The distinct applications are distinguished through the use of the UDP port number field present inside the UDP header (see appendix A.3).

2.4.3.4 APL Layer

The error-detection mechanisms provided by the CRCs (at MAC layer) and checksums (at UDP layer) combined with the retransmission mechanism at the lower layer of SPS, allow the APL layer to receive only the error-free packets. The price to be paid is a reduced throughput due to MAC level retransmissions. Error-resilient features of the standard H.264 decoder can be utilized at APL layer, to recover the lost and dropped packets.

2.5 Scenario Considered

The scenario considered in this thesis comprises of the same standard topology presented in the previous section. JSCD techniques mentioned in the previous chapter are not compliant with the SPSs of WiBOX and WiFi SS, since they require flow of soft information from the PHY layer to the APL layer. Furthermore, error-detection mechanisms included in the SPSs may prevent corrupted packets to reach the APL layer.

The difficulty comes from the fact that due to the high error-rates of wireless media, many errors are not corrected by the PHY layer. These errors cause CRC or checksum

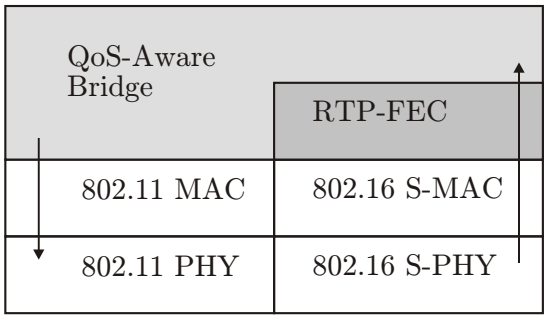

Figure 2.7: S-PPS of WiBOX

failures at higher layers of the SPS, consequently leading to a significant number of packet drops. Given the fact that one is using error-resilient video decoder at the APL layer, the main objective of the lower layers should be to relay maximum number of (error-free and erroneous) packets to a wireless receiver. To achieve the required task, WiBOX and WiFi SS should consist of an improved soft-permeable layer mechanism or S-PPS, having the ability to exchange soft and corrupted information between the protocol layers. This would enable the use of the robust error-mitigation tools, like JSCD, at APL layer.

To address this problem, a UDP-lite protocol was proposed in [2] to provide partial protection by a checksum that only covers the header, while no checksum is provided for the payload. Similarly, to forward erroneous payloads to the APL layer, in [82] the CRC mechanism is modified to only check error in the header part along with an addition of the bit-level FEC for the header part. But, support for a partial checksum requires modifications to multimedia transmitters/receivers and thus is not preferred due to incompatibility with several receivers.

To develop S-PPS, the bottleneck is that the information inside the header is very critical and the packet with erroneous header cannot be interpreted. However, if the header can be reliably estimated, then the soft information, provided by the channel decoder at PHY layer, can reach the APL layer. This problem has been partly addressed in [8] for header recovery at PHY/MAC layer and adapted in [61] to the higher protocol layers (IP and UDP-lite) protected by checksums. The payload of a packet may then more reliably be forwarded to the higher protocol layers.

Thus, to develop such S-PPS, one needs minor modifications at the receiver, while the transmitter functionality remains unchanged (an essential requirement). Now we will provide the features of S-PPSs of WiBOX and WiFi SS.

2.5.1 S-PPS of WiBOX

The S-PPS of WiBOX is shown in Figure 2.7 and functionality of each layer is provided below.

2.5.1.1 802.16 Soft-PHY Layer

For the deployment of a robust video decoder at the APL layer, we are more interested in the soft decoding, therefore the PHY layer is capable of forwarding the soft burst to the MAC layer. Soft-output demodulator, interleaver, channel decoder, and derandomizer

could be deployed at the PHY layer to enable propagation of the soft information to higher layers. Soft decoding provides for each bit of the transmitted binary sequence the probability that it is equal to 0 (or to 1) known as APP.

2.5.1.2 802.16 Soft-MAC Layer

After decoding the DL-MAP of WiMAX PHY frame, APPs of the required soft burst are passed to the MAC layer, which extracts the MAC packets corresponding to the WiFi SS. The method of reading CID and length field from the MAC header is no more applicable due to the soft information. Thus, the robust segmentation of soft burst into its constituent MAC packets using the soft information is required.

Robust FS is a critical tool used at this layer to mitigate the effect of a single bit error on the complete burst and to robustly segment the burst into its constituent packets. FS is explained in Chapter 3. A robust FS technique, proposed in chapter 4 with its low-delay and reduced-complexity variant presented in chapter 5, is deployed to fragment the soft burst into MAC layer s-packets. These robust FS techniques aim to estimate the location and the content of the headers (the length fields) of the packets forming the burst, so that the MAC s-packets even with erroneous payloads can be forwarded to the upper protocol layer (*i.e.*, RTP-FEC layer). Moreover, an on-the-fly FS technique, presented in chapter 6, can also be used to provide an efficient trade-off between performance and complexity/latency.

The fact that the HCS field covers only the header poses no problem even in the SPS to forward the corrupted payload to the upper layer. But, s-packets obtained after segmentation using FS cannot use the standard method of HCS verification and differentiation using CID. Fortunately, they can be forwarded to the upper layers using the recently proposed header estimation techniques [8; 9; 61] and robustly decoded at APL layer using JSCD techniques.

2.5.1.3 RTP-FEC Layer

WiBOX receives the packets from the WiMAX-link and forward them to the WiFi-link. There are two main reasons to add this layer. First, due to the lossy indoor environment for WiFi-link transmission, one needs to provide additional protection to the packets. Second, due to the omission of retransmissions at the MAC layer, one needs an alternative FEC based lossy recovery technique to reduce errors left by the PHY layer. This layer would help to decrease the PER at the WiFi SSs.

This layer receives the RTP s-packets, which are to be rebroadcasted over WiFi-link, therefore these s-packets should be thresholded to generate hard RTP Media Packets (MedPs). The packet-level FEC is then applied to generate redundant RTP FEC Packets (FECPs) from the these MedPs. For simplicity, we propose to use an XOR operation to generate FECPs as presented in RFC 5109 and consider a simple scenario, where two MedPs, let say MedP1 and MedP2, received from the WiMAX-link are encoded to give one FECP. The packet-level FEC encoding using RFC 5109 is performed, which is explained in Chapter 7.

Note that we have not shown IP and UDP layers, which would de-multiplex packets belonging to different service flows or streams. Moreover, RTP-FEC layer is active only for the delay-constrained streams.

2.5.1.4 IEEE 802.11 MAC Layer

The intent is to remain compatible with the standard as much as possible, thus we choose HCCA, provided in 802.11e, to be used as a multiple access scheme for the WiFi-link. DL packets are available in the WiBOX buffers and can be directly scheduled. This layer handles the following issues to insure QoS.

- Maintaining the list of streams and the list of corresponding WiFi SSs.
- Combining the task of scheduling the FECPs and the MedPs into a central scheduler.
- Scheduling of packets belonging to different TSs using the reference scheduler defined in the WiFi standard [75].

2.5.2 S-PPS of WiFi SSs

It is assumed that WiBOX performs FEC encoding on each stream independently, while at the WiFi SS, the UDP protocol differentiate between different streams, and the RTP header/payload errors are subsequently corrected using FEC decoder at the RTP-FEC layer. Taking all of this into account minor protocol stack modifications are needed at the WiFi SS. The proposed framework is shown in Figure 2.8. It is assumed that the receiver uses the S-PPS, which is capable of forwarding even erroneous s-packets to the higher layers. Functionality of each layer of the S-PPS of the WiFi SS is now provided in detail.

2.5.2.1 IEEE 802.11 Soft-PHY Layer

It employs soft decoding to achieve the goal of providing s-packets to the MAC layer.

2.5.2.2 IEEE 802.11 Soft-MAC Layer

The idea of MAC-lite (CRC covers only the header) has been presented in [4; 5; 6], where the receiver would only request retransmission of packet having erroneous MAC header. But MAC-lite is not an attractive solution, as it would not only be incompatible with several WiFi SSs but would also require modifications to the transmitter. Thus, on the contrary, in this thesis the ARQ is completely disabled.

Using the MAC layer CRC, one can identify the error-free s-packets by calculating CRC over the corresponding hard packet (*i.e.*, hard version of the s-packet under consideration). Otherwise, the packet is corrupted and s-packet is marked as corrupted, necessitating recovery at higher layers. Thus, the MAC layer is forwarding even the erroneous s-packet to the upper layer. Given the fact that the MAC s-packet is erroneous, it is equally possible that the MAC destination address (address of the receiver) is corrupted and it is wrongly mapped to the receiver. Though, in such case, a robust estimation algorithm can be employed to estimate the MAC destination address and to find out whether it has the receiver's MAC address. But, thanks to HCCA, the receiver only reads MAC s-packets intended for it from the assigned polling slot, thus avoiding an additional estimation requirement.

2.5.2.3 Soft-IP Layer

At WiFi SS, the lower layers *i.e.*, PHY and MAC layers have soft decoding capabilities, thus they forward even partially erroneous s-packets to the Soft (S)-IP layer. By using the

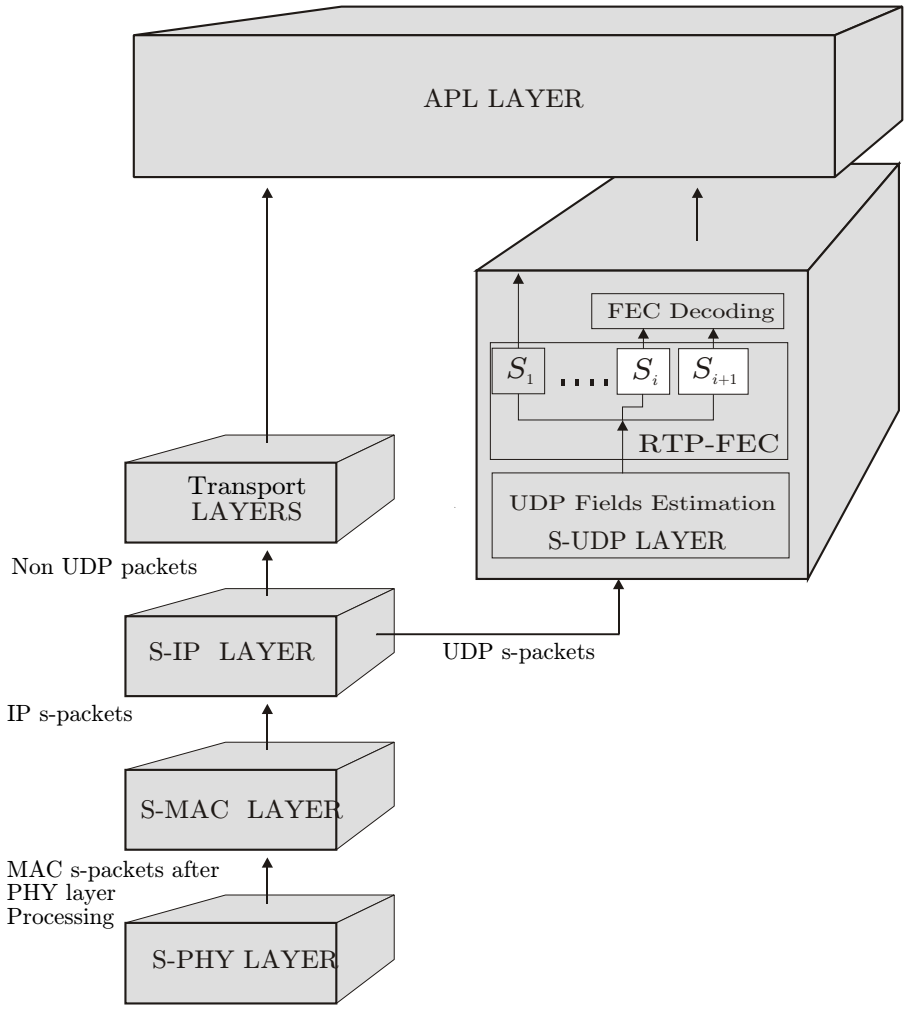

Figure 2.8: S-PPS framework of WiFi SS

header estimation techniques, if the payload of the IP s-packet is a UDP packet, then it would be forwarded to the UDP layer, else it would be passed to other transport layers.

2.5.2.4 Soft-UDP Layer

The use of UDP-lite [2; 3] (see appendix A.4 for detail on the UDP-lite header) can reduce the packet drops by employing a partial checksum which only covers the sensitive-part of a packet, while errors in the insensitive-part are ignored. But, UDP-lite incurs packet drops due to errors in sensitive-part, especially at high data rates. Furthermore, support of partial checksum for UDP-lite requires changes to the standard protocols at the multimedia transmitter. Thus, we propose to stick to UDP as to remain compatible with all users and not to drop UDP packets with erroneous UDP checksum, instead these corrupted packets would be recovered at the RTP-FEC layer.

It should be noted that we are forwarding s-packets even with erroneous headers to Soft (S)-UDP layer. Given the fact that the UDP packet is erroneous, it is equally possible that the UDP destination port is corrupted and, if thresholded, can be mapped to the

wrong stream, thus there are possibilities of false alarms. Furthermore, as one is dealing with s-packet thus the estimation of the destination port field is required, and it can be performed with minor false alarms using the header estimation techniques presented in [11] or in [8]. This estimation would identify the stream S_i to which the s-packet under investigation belongs and whether the packet-level FEC is active for it or not. If the FEC scheme is active for the estimated destination port, then the s-packet will be passed to the FEC decoder.

2.5.2.5 RTP-FEC Layer

At this layer the s-packets are demultiplexed by estimating the port number field of the UDP header. Now, for the FEC active streams, the main question is: what should be the method of packet recovery from the received corrupted s-packets (MedPs and FECPs)? The received s-packet at the RTP-FEC layer has APPs or the soft confidence or probability values for each bit of being '1' or '0'. However, most FEC decoders employed at the higher layers (IP/RTP) deal only with the hard and error-free packets. Consider the FECP generation from MedP 1 and MedP 2, as presented earlier. Now, if MedP1 is lost in WiFi-link it can be recovered from an uncorrupted MedP2 and FECP. But, in several situations it would be difficult to receive fully uncorrupted packets, especially in the case when the channel conditions are not favorable and the lower layers are forwarding erroneous packets (required for example for the robust video decoding).

In the considered scenario, as the lower layers are capable of forwarding erroneous s-packets, instead of dropping them. Thus, soft-input soft-output decoding of the s-packets to recover or reconstruct the corrupted packet is needed to forward s-packets to the APL layer for video decoding, without any hindrance.

Two scenarios exist when a s-packet is received:

1. S-packet is error-free: No need to perform recovery, but can be used for the recovery of other packets.
2. S-packet is erroneous: Recovery is required.

To recover erroneous MedPs, the robust packet-level FEC decoding is performed, which is presented in Chapter 8.

2.5.2.6 APL Layer

The robust video decoders using JSCD techniques can now be deployed at the APL layer, given the fact that due to the S-PPS, the APL layer can now receive erroneous s-packets. Considering the fact that RTP-FEC layer does not guarantee 100 % recovery, the erroneous s-packets need to be detected at the APL layer using *e.g.*, error detection methods presented in [83]. Erroneous NALU inside the RTP packet can be detected if the estimation error has caused H.264 encoder syntax error, as presented in [55; 84].

PART I: Frame Synchronization

Chapter 3

Introduction to Frame Synchronization

In several communication systems including WiMAX and WiFi, packet aggregation techniques at intermediate protocol layers of the protocol stack have been proposed, where small packets are aggregated into larger packets or bursts in order to reduce the overhead due to headers and increase the throughput. These aggregated packets are then forwarded by the transmitter to lower protocol layers. As an example, packet aggregation techniques at intermediate protocol layers have been studied recently [85] in the context of WiFi standard. It is also called frame aggregation if it is deployed at the lower layers of the protocol stack, *e.g.*, at PHY layer. In this thesis, without distinguishing where the aggregation process is performed, be it at the PHY, MAC, IP, or at higher layer, it is called packet aggregation and the single aggregated packet is named a burst.

Aggregation process has similar functionality as the multiplexing scheme proposed in [15; 16], where DL packets are multiplexed into a single packet. Gateway combines multiple packets into a single multiplexed packet, and multicasts the multiplexed packet to the wireless end stations. The demultiplexer in each end station extracts its respective data and forwards them to the application. This has been shown to reduce the overhead of multiple packets and to improve capacity.

The price to be paid is a higher sensitivity of the large packets to transmission errors. Thus, aggregation has two effects: on one side, it increases the useful throughput in quite a large amount, but, on another side, any error upon reception of the burst header results in the loss of the complete burst, composed of many individual packets. When the header of burst has been correctly received, efficient packet alignment becomes very important, since if some packets are not correctly delineated, a large amount of bits has to be retransmitted. In such situation, if the delay is constrained (situations where one cannot afford retransmission of large aggregated packets) or when retransmitting bursts in error is not possible (broadcasting), synchronization algorithms are the relevant techniques, which can be used to properly recognize and maintain packet boundaries. In case the network fails to properly recognize the packet boundaries, it is said that the packet network is out of synchronism. In this thesis, we assume that symbol (bit) synchronization has already been performed at PHY layer [86] and explore packet-level synchronization or Frame Synchronization (FS).

This chapter provides an introduction to FS. No new results are reported in this chapter, only the results for the state-of-the-art FS techniques are presented. The robust FS techniques proposed in this thesis are presented in Chapters 4-6.

Figure 3.1: Aggregated packets

3.1 Introduction

FS is an important problem arising at various layers of the protocol stack of several communication systems. Important examples include the sequence synchronization at a spread-spectrum receiver, which is required before initiating any communication between the end points [87]. FS can be very critical in transmission of data through wireless links where, due to the use of *e.g.*, powerful error correcting codes, the receiver is designed to work with very low SNRs. FS is often required at the PHY layer, to recover the payload and the side information (headers, preambles, etc...) of the PHY packets or frames. Therefore, FS has received continuous attention since many years [88; 89; 90; 91; 17; 21]. Initially, FS has mainly been considered at the PHY layer, albeit this problem may also occur at upper layers of the protocol stack. An example is video transmission where information is usually organized in packets, and decoding requires the knowledge of their exact boundaries. This is the case, *e.g.*, in the MPEG-4 video stream [92; 93], where synchronization markers are aperiodically embedded in the bitstream.

The FS may become difficult when the received burst is corrupted by noise. To facilitate the process of FS at the receiver, often a *Synchronization Word* (SW) pattern is injected periodically into the data stream (continuous transmission) or appended at the beginning of each packet (packet transmission) [88; 89; 90]. In the absence of SW, the Header Error Control (HEC) field of the packet header can be employed [17]. Figure 3.1 shows a typical packet aggregation format with SW or header prefixed to the payload, now, the fundamental problem is to achieve FS, *i.e.*, finding the exact starting location of the SW/header.

In several situations, *e.g.*, when the SW/header is erroneous itself, the performance of FS degrades. In general, the performance of FS depends on the probability of *missed detection*, *i.e.*, missing a correct SW/header, and on the probability of *False Alarm* (FA) or emulation, *i.e.*, false detection of the start of SW/header. It is thus important for FS algorithms to minimize these probabilities. The miss detections can occur due to the channel noise in the SW/header, while the FAs are due to the fact that sometimes the random data plus noise can be interpreted as a SW/header. False alarms can occur either in the case when data symbols are coincident with the SW/header pattern or due to the channel noise, even if data symbols are different from the SW/header pattern. A single FA or miss detection can cause loss of two consecutive packets.

In the scenario under investigation in this thesis, as discussed in Chapter 2, our main concern is to perform FS at the MAC layer to localize the MAC packets, which are of variable length. Nevertheless, we will also provide a brief overview of the fixed-length FS techniques from the literature, because of their similitude with the variable-length FS techniques.

The rest of this chapter is organized as follows: The structure of the fixed-length or variable-length packet is provided in Section 3.2. The state-of-the-art FS techniques for the fixed-length packet network and the variable-length packet network are presented in Section 3.3 and Section 3.4, respectively. Finally, the simulation results for the variable-

length FS techniques are presented in Section 3.5, followed by the conclusions in Section 3.6.

3.2 Packet Structure

The two main conceptual packet networks that are common today are the fixed-length networks, such as the ATM network, and the variable-length packet networks, such as the WiMAX/WiFi network. In the fixed-length packet networks, the sizes of the packets are the same as well as their internal structure, where as in the variable-length packet networks, the sizes of the packets are variable. These packets are usually separated from each other by well defined indicator flags, packet headers, synchronization markers or SWs, which are located at the beginning of the payloads.

Let us define some notations on the structure of the variable-length or fixed-length packets. Consider the n-th packet of variable-length packet network (or fixed-length packet network) at a given protocol layer. This packet is assumed to contain $\lambda_n = \ell_h + \ell_{p,n}$ bits, where the leading ℓ_h bits represent the fixed-length packet header or SW represented by $\mathbf{h}_n$ and the remaining $\ell_{p,n}$ bits are the variable-length (or fixed-length) payload, denoted by $\mathbf{p}_n$. The payload is assumed to be generated by a binary symmetric source. The HEC, which can be a CRC or checksum, is assumed to be present in the last ℓ_c bits of the header and covers only the header (*i.e.*, $\ell_h - \ell_c$ bits) without covering the payload.

3.3 Fixed-length State-of-the-art FS techniques

In order to understand the variable-length FS, let us first briefly explain the FS for the fixed-length packets. In the pioneering works [88; 89; 90] on FS to delimit fixed-length packets in the bit stream, regular spaced fixed patterns or SW are assumed to be present, as is the case, *e.g.*, at the PHY layer of Digital Video Broadcasting-Handheld (DVB-H) [94] for MPEG2 transport stream packets. At the receiver, after recovering timing information, sampled received values are typically correlated with a SW and FS is accomplished by examining the correlation values [88; 90]. This type of FS method, which is generally referred to as the correlation rule, has been popular because of its simplicity in implementation and acceptable performance. For example, in [88], FS is performed by maximizing the correlation between the SW and the received data.

FS can also be achieved using optimal rules such as the Maximum-Likelihood (ML) rules and their various modifications [89; 91]. These rules outperform the correlation rules at the expense of additional computation. In [89] the optimal statistic for FS has been proposed for the Additive White Gaussian Noise (AWGN) case, taking into account the presence of data around the SW. Extensions to more sophisticated transmission schemes and code-aided FS techniques can be found in [91; 95; 96; 97]. A performance loss is experienced when the frequency or phase error exists in the channel. Under such circumstances, improved correlation metrics [98; 99] provide robustness to frequency and phase errors. Optimum and low-complexity sub-optimum techniques in the presence of phase offset due to imperfect carrier phase estimation are proposed in [100], where possible extensions to higher order modulations and fading channels are also provided. The above-mentioned FS techniques work on-the-fly, *i.e.*, only a small portion of the burst is processed at each time to perform FS.

However, the most common fixed-length packet network, *i.e.*, ATM network, has an HEC present in the header, and FS is achieved through HEC search instead of SW local-

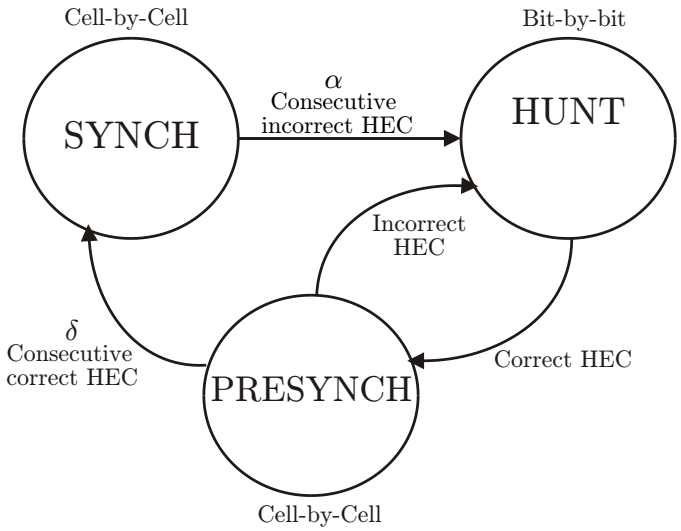

Figure 3.2: 3S automaton for CD in ATM network

ization. FS in ATM network is called *Cell Delineation* (CD) and fixed-length packets are called *cells*. We will now explain CD in detail, as the same basics are followed by some of the variable-length FS techniques, which are of prime interest in this thesis.

3.3.1 Cell Delineation

CD is an HEC-based FS technique, where HEC is evaluated to confirm and to search for FS. According to CD, any ATM cell that is out of synchronism enters a specific state of the 3S automaton. After applying the relevant algorithm the state of the 3S automaton changes, till it gets into the proper state for releasing correct cell back to the network. The 3S automaton for CD, as provided in [39, pp. 7], is shown in Figure 3.2. Detail description of the states of the 3S automaton is provided below.

***HUNT* State:**

This state is reached if α consecutive non-correct HECs have been detected. In this case, the 3S automaton enters into the *HUNT* state. Now, the cell in this state is properly recovered by applying a sequence of bit shifts until a correct HEC is discovered. The delineation process is performed by checking bit-by-bit for the correct HEC for the assumed header field, once such an agreement is found, it is assumed that one header has been found, and the process enters the *PRESYNC* state. The CD process may be performed byte-by-byte. The parameter α determines transition delay and should be reasonably selected.

***PRESYNC* State:**

The delineation process is performed in *PRESYNC* state by checking cell-by-cell for the correct HEC. The process repeats until the correct HEC has been confirmed δ times consecutively, at which point the process moves to the *SYNC* state. If an incorrect HEC is found, the process returns to the *HUNT* state. The total number of consecutive correct HECs required to move from the *HUNT* state to the *SYNC* state is therefore $\delta + 1$. This process of the transition into the *SYNC* state imposes some delay on the overall correction

process. This delay is unavoidable because the detection of the correct HEC could be caused not only by correct FS, but also by FAs.

***SYNC* State:**

This state is reached with the receipt of sequential cells with correct HECs. In the *SYNC* state the FS will be assumed to be lost if an incorrect HEC is obtained α times consecutively.

In the *HUNT* state, a search process is initiated just one bit after the correct FS position. All bit positions in a cell are checked until the correct FS position is reached. The search process checks the bit pattern at each position and shifts to the next position if the pattern does not correspond to a correct HEC. More precisely, the process searches for the correct HEC by calculating HEC over the $\ell_h - \ell_c$ previously received bits and compares it against the current received HEC. It continues until it reaches the correct bit position, provided the HEC is not corrupted by bit errors.

If the correct HEC is simulated, a confirmation process in the *PRESYNC* state is used to detect its falsehood. It checks for δ consecutive ($\delta = 7$ in the ATM standard) valid HECs before declaring correct FS. If the falsehood is detected, the search process is re-initiated, and the next bit position is checked.

The search process moves from the bit position ℓ to $\ell + 1$ in a 1-bit duration, if the correct HEC is not simulated. If random data simulates the HEC, the confirmation process takes at least one cell duration to detect its falsehood. The parameters α and δ are to be chosen to make the CD process as robust as possible. The miss detection or false indication of misalignment can occur due to bit errors in the cell header caused by the channel and the robustness against it depends on α. While, the FA or emulation in the resynchronization process is caused by simulation of a correct HEC by a random data and robustness against it depends on the value of δ.

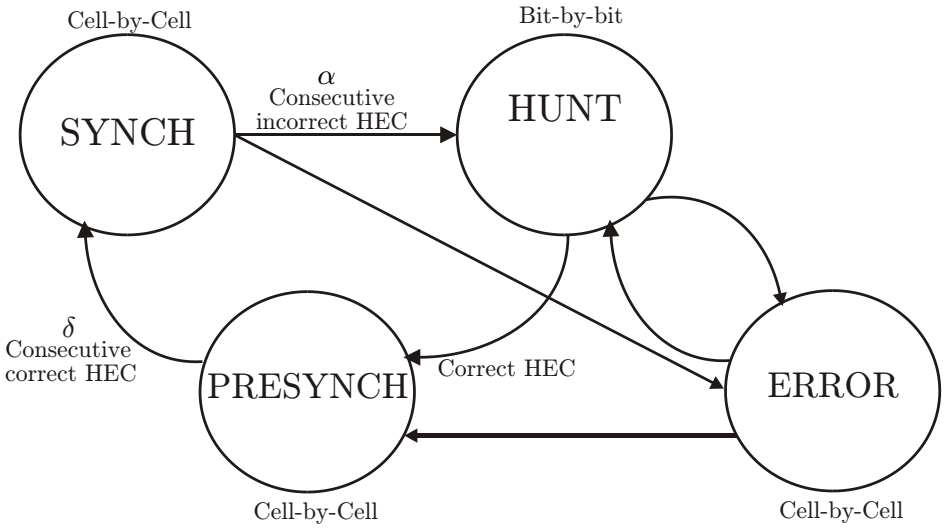

Figure 3.3: Modified automaton for CD in ATM network

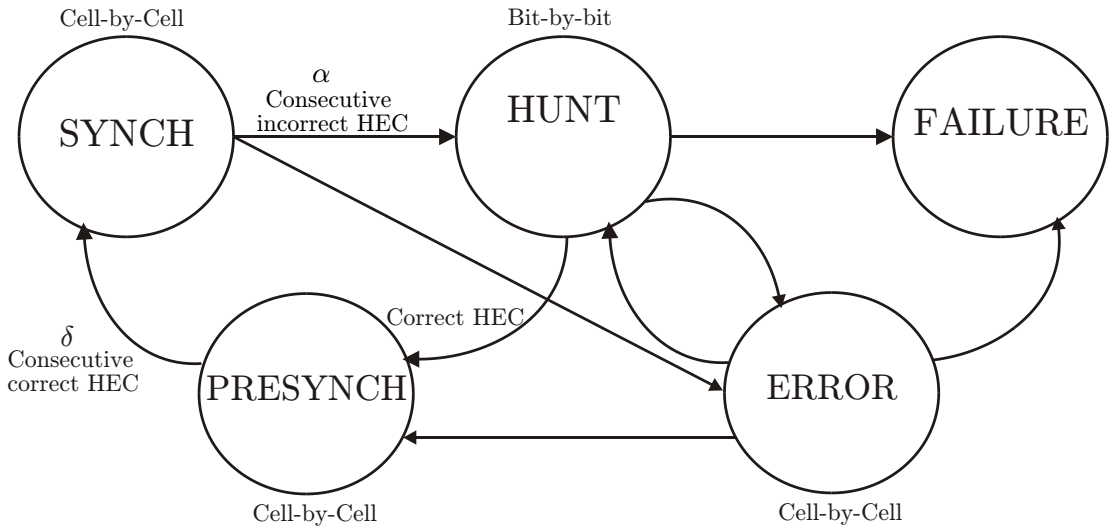

Figure 3.4: Broadened automaton for CD in ATM network

3.3.1.1 Modified CD

For the ATM networks, according to the standard, any bit error or errors detected in the cell header will cause cell to be dropped. However, it is quite dangerous, because the cell boundaries are not clear and any drop may cause additional problems. The origin of this problem is the fact that FS failures and other channel errors are indistinguishably treated in the same way leading to too much cell drop and information transfer loss. In [38], a strategy to differentiate between FS failures and other channel errors is proposed. An additional *ERROR* state is added to the previously described CD protocol in order to distinguish between FS failures and channel errors, and then trying to recover and correct random errors in the fixed-length cell header. This Modified CD (MCD) process is shown by the modified automaton in Figure 3.3.

Newly added *ERROR* state in Figure 3.3 enables the system to distinguish between two incompatible processes, the FS failures and other channel errors. It study cells within a certain time-window T and calculate the number of Black (B) type cells with detectable error (FS failures and channel errors) in the cell header and White (W) type cells with correct cell header. Based on this calculation a variable threshold is defined, which triggers newly added *ERROR* state for error correction and *HUNT* state for FS, mutually exclusively or simultaneously. Furthermore, another state, the *FAILURE* state, is added to represent non-recoverable channel degradation problems and errors. The broadened automaton with this absorptive *FAILURE* state is shown in Figure 3.4.

3.4 Variable-length State-of-the-art FS techniques

In many communication systems, packets are of variable-length, this is the case, *e.g.*, in the 802.11/802.16 standards [101; 32] under investigation in this thesis. Variable-length FS is much more complicated, as a proposition to study the time-window cannot be applied due to the variable nature of packets. The Conventional and simple *Hard Decision* (HD)-based FS using a length field (containing the packet size), assumed present in the header of the variable-length packet, can be used if the noise is moderate. HD-based FS rely on

the correctness of the length field of the previous packet to find the start of the current packet and suffers from limitations if the length field is corrupted with noise.

The most widely used method for providing variable-length FS is to insert a SW or header into the data stream. A simpler and common engineering approach for FS of unknown lengths consists in a sequential detection algorithm: Starting from a given position, the correlation between the received (continuous valued) samples and the SW/header symbols is computed, and compared with some threshold. If the threshold is exceeded the synchronizer declares a SW/header presence, otherwise the search continues [90]. In the presence of SW, in [19; 20; 21; 22], several hypothesis testing techniques have been presented to perform variable-length FS. These techniques are discussed in Section 3.4.1.

Nevertheless, one can use the 3S automaton as presented for the fixed-length packets, but the misframe time (time taken to wrongly signal FS failure) for variable-length packet is too short as $\alpha = 1$, thus even with a single random bit error in the header the system may quickly and falsely switch to the *HUNT* state by signaling FS failure due to missed detection. This problem is addressed in [17], which is briefly reviewed in Section 3.4.2.

3.4.1 NP FS Method

If no *a priori* information on the packet length is assumed, then an acquisition algorithm based on a step-by-step comparison of a proper metric with a threshold is a preferable solution. Several detection metrics for AWGN channel based on the Likelihood Ratio Test (LRT) are considered by Chiani et al [19; 20; 21; 40; 22] to perform FS through correlation, where Neyman-Pearson (NP) criterion is used to choose the threshold. Authors in [40] studied the performance of FS for equiprobable data symbols with the metrics derived in [21; 20], while in [22] an optimal metric is derived for non-equiprobable data symbols. It is further extended in [23] to exploit *a priori* information on the prevalence of ones and zeros in the payload at the price of a small additional computational complexity. This method is denoted as NP FS in what follows. Below a brief description of NP FS method is provided.

Let us consider the case of binary signaling where we have data symbols $d_i \, \epsilon \, \{+1, -1\}$, with probabilities $P(d_i = 1)$ and $P(d_i = -1)$. Similarly, we have known SW or header symbols $h_i \, \epsilon \, \{+1, -1\}$.

Let $b_i \, \epsilon \, \{0, 1\}$ be the i^{th} bit of a burst transmitted using binary antipodal modulation through AWGN channel. At the receiver, we have

$$y_i = (-1)^{b_i} + n_i,$$

where n_i are independent, identically distributed Gaussian random variables (r.v.s), with zero mean and variance σ^2.

The NP FS algorithm works as follows: Starting from a bit index ℓ, the synchronizer observes a vector of ℓ_h subsequent samples; based on the metric evaluated from this vector it decides if the SW/header is at this bit index ℓ; if not, it moves to next bit index $\ell + 1$, repeating the steps until the SW/header is detected. After observing ℓ_h subsequent samples at bit index ℓ, the synchronizer must choose between the following two possible hypotheses, *i.e.*, the data hypothesis

$$H_d \, : \, y_i = d_i + n_i, \; i = \ell, \ell + 1, ..., \ell + \ell_h - 1 \tag{3.1}$$

and SW/header hypothesis

$$H_h \ : \ y_i = h_i + n_i, \ i = \ell, \ell+1, ..., \ell+\ell_h-1. \tag{3.2}$$

Decisions are indicated by D_h and D_d, corresponding to the true hypotheses H_h and H_d, respectively.

Let the received sequence of ℓ_h symbols be denoted by $\mathbf{y} = [y_\ell, y_{\ell+1}, ..., y_{\ell+\ell_h-1}]$, which is composed of either noisy SW/header symbols or noisy data symbols. Let $\mathbf{Y} = [Y_\ell, Y_{\ell+1}, ..., Y_{\ell+\ell_h-1}]$ be the r.v.s corresponding to the vector $\mathbf{y} = [y_\ell, y_{\ell+1}, ..., y_{\ell+\ell_h-1}]$ of received samples, the LRT is represented by

$$\Lambda(\mathbf{y}) = \frac{P_{Y|H_d}(\mathbf{y}|H_d)}{P_{Y|H_h}(\mathbf{y}|H_h)} \underset{D_d}{\overset{D_h}{\lessgtr}} \lambda. \tag{3.3}$$

Where $P_{Y|H_j}(\mathbf{y}|H_j)$ is the probability density function of $\mathbf{Y}$ under hypothesis H_j, $j \, \epsilon \, \{d, \, h\}$, and λ is the selected threshold. Thus, according to the LRT, $\Lambda(\mathbf{y}) < \lambda$ corresponds to the decision D_h, *i.e.*, the presence of a SW or header is detected; otherwise, the decision is D_d.

Now we will provide a brief evaluation of the LR of (3.3) for the AWGN channel. Since the channel is memoryless, we know

$$P_{Y|H_j}(\mathbf{y}|H_j) = \prod_{i=\ell}^{\ell+\ell_h-1} P_{Y_i|H_j}(y_i|H_j).$$

Given H_h, the r.v.s Y_i are Gaussian-distributed with mean h_i and variance σ^2, *i.e.*,

$$P_{Y_i|H_h}(y_i|H_h) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_i-h_i)^2/2\sigma^2}. \tag{3.4}$$

Similarly, given H_d and d_i, the r.v.s Y_i are Gaussian-distributed with mean d_i and variance σ^2, *i.e.*,

$$P_{Y_i|H_d}(y_i|H_d, d_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_i-d_i)^2/2\sigma^2}. \tag{3.5}$$

Equiprobable data symbols

In case of equiprobable data symbols, under hypothesis H_d, d_i take values $+1$ and -1 with equal probability, one gets [21]

$$\begin{aligned} P_{Y_i|H_d}(y_i|H_d) &= \frac{1}{2} P_{Y_i|H_d}(y_i|H_d, d_i = 1) \\ &\quad + \frac{1}{2} P_{Y_i|H_d}(y_i|H_d, d_i = -1), \end{aligned} \tag{3.6}$$

which, using (3.5), becomes

$$\begin{aligned} P_{Y_i|H_d}(y_i|H_d) &= \frac{1}{2} \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_i-1)^2/2\sigma^2} \\ &\quad + \frac{1}{2} \frac{1}{\sqrt{2\pi}\sigma} e^{-(y_i+1)^2/2\sigma^2}. \end{aligned} \tag{3.7}$$

Using (3.7) and (3.4) in the LR of (3.3), one finally has

$$\begin{aligned}\Lambda(\mathbf{y}) &= \frac{1}{2^{\ell_h}} \prod_{i=\ell}^{\ell+\ell_h-1} \frac{e^{-(y_i-1)^2/2\sigma^2} + e^{-(y_i+1)^2/2\sigma^2}}{e^{-(y_i-h_i)^2/2\sigma^2}} \\ &= \frac{1}{2^{\ell_h}} \prod_{i=\ell}^{\ell+\ell_h-1} \left(1 + e^{-2y_ih_i/\sigma^2}\right). \end{aligned} \tag{3.8}$$

Non-equiprobable data symbols

In case of non-equiprobable data symbols, under hypothesis H_d, d_i take values $+1$ and -1 with probabilities p_1 and $1-p_1$ respectively, one gets [22]

$$P_{Y_i|H_d}(y_i|H_d) = \frac{1}{\sqrt{2\pi}\sigma}\left[p_1 e^{-(y_i-1)^2/2\sigma^2} + (1-p_1)\, e^{-(y_i+1)^2/2\sigma^2}\right]. \tag{3.9}$$

Using (3.9) and (3.4) in the LR of (3.3), one finally has

$$\begin{aligned}\Lambda(\mathbf{y}) &= \frac{1}{2^{\ell_h}} \prod_{i=\ell}^{\ell+\ell_h-1} \frac{p_1 e^{-(y_i-1)^2/2\sigma^2} + (1-p_1)\, e^{-(y_i+1)^2/2\sigma^2}}{e^{-(y_i-h_i)^2/2\sigma^2}} \\ &= \frac{1}{2^{\ell_h}} \prod_{i=\ell}^{\ell+\ell_h-1} \left[P(d_i = h_i) + P(d_i \neq h_i) e^{-2y_ih_i/\sigma^2}\right]. \end{aligned} \tag{3.10}$$

NP FS method thus decides D_h (the SW/header is present) if $\Lambda(r) < \lambda$ and D_d otherwise. One can observe that neither the LR nor the threshold depends on the *a priori* probabilities $P(H_j)$, they depend only on the conditional densities $P_{Y_i|H_j}(y_i|H_j)$. Furthermore, the metric for non-equiprobable case is somewhat similar to the one for the equiprobable case, and depends on the channel conditions through σ^2 . Thus, to perform optimum FS using the NP FS method the instantaneous knowledge of the SNR is required.

The threshold is chosen according to the NP criterion, *i.e.*, by fixing the maximum tolerable probability of FA (emulation), p_{em}. The probability of emulation, p_{em}, *i.e.*, the probability of choosing hypothesis H_h when H_d is true, is

$$p_{em} = P(D_h \,|\, H_d),$$

while, the probability of missed detection, p_{md}, *i.e.*, the probability of choosing hypothesis H_d when H_h is true, is

$$p_{md} = P(D_d \,|\, H_h).$$

NP FS method achieves high gain as compared to commonly used correlation-based FS techniques, but in case SW is short *e.g.*, less than 16 bits, the FAs significantly degrade the FS performance. This can be overcome by exploiting the *a priori* information on the packet length in order to adapt the threshold, this problem is addressed in Chapter 6 using Bayesian hypothesis test.

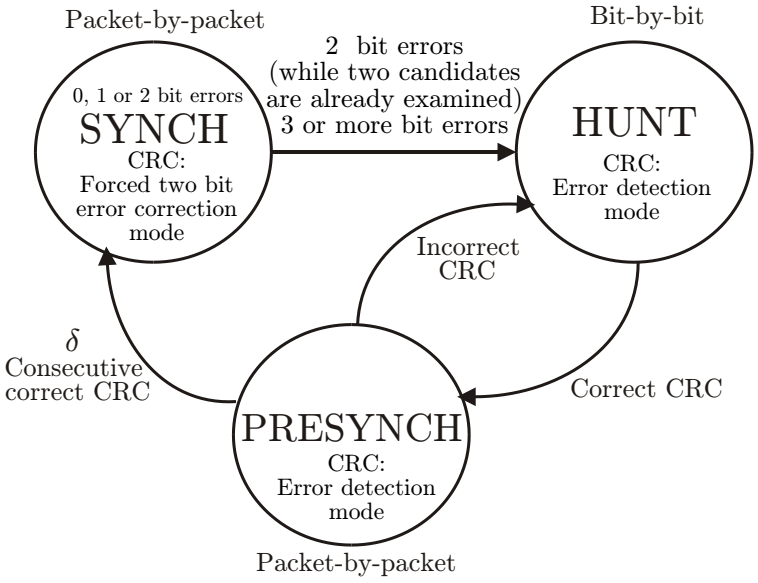

Figure 3.5: 3S automaton used by the Ueda's method

3.4.2 Ueda's Method

In [17], an CRC-based variable-length FS for IP packets is proposed, where a length field is assumed to be present in the packet header. This method uses the 3S automaton, similar to the one used in CD, but instead of distinguishing between FS failures and channel errors, up to two bit error correction (to increase misframe time) is performed before hunting for the correct FS, if required. This method is denoted in what follows as *Ueda's method*.

More precisely, the automaton presented in [17] consists of three states: *SYNC*, *HUNT*, and *PRESYNC*. Assume that the automaton is in the *SYNC* state. It remains in this state as long as no FS failure is detected using CRC (as a HEC). If CRC detects a FS failure, one first tries to correct errors in the header. The 3S automaton used by the Ueda's method is shown in Figure 3.5.

In the *SYNC* state, CRC can correct one-bit error but not two-bit errors because two-bit error syndromes do not necessarily correspond to a single candidate, thus multiple candidates may exist. For two-bit errors, if a single candidate exists then the error can be corrected immediately. But if dual candidates are possible, the best packet length candidate is searched by evaluating the CRC over the next header, at the position indicated by the corresponding candidate. If no candidate is correct, *i.e.*, corresponds to correct CRC at the potential location of the next header, then the candidate that gives one-bit error syndrome at the next header is selected. In case of failure, the automaton switches to the *HUNT* state. As few syndromes of more than two-bit errors are similar to one-bit or two-bit error syndromes, so it is assumed that these error syndromes are originating from one-bit or two-bit errors because they are more often.

In the *HUNT* state, the automaton hunts for the correct FS by searching bit-by-bit for the correct CRC over the assumed header fields. Once an agreement is found, the automaton switches to the *PRESYNC* state, an intermediate state. In this state, a packet-by-packet checking is performed. In the *PRESYNC* state, one makes sure that the FS retrieved in the *HUNT* state is indeed the correct FS by verifying that CRC is correct

for $\delta > 0$ consecutive packets. Once δ consecutive correct CRCs have been obtained, the automaton returns to the *SYNC* state and in case of failure it again switches to the *HUNT* state.

Synchronizer can be implemented in two ways; one is to utilize two synchronizers to select the best suitable candidate out of maximum of two candidates. The second is to sort the two candidates in decreasing order and utilize a single synchronizer to examine the candidates in decreasing order. The second implementation is adopted in Ueda's method.

Ueda's method is well adapted for rather long CRC like CRC-16, because in this case there are no more than two candidates for two-bit error syndromes. For low order or short CRC compared to the size of the header, *e.g.*, CRC-8, many more candidates can be found for syndromes with two bits in error, thus Ueda's method needs some extension to search for the best candidate.

3.4.2.1 Modified Ueda's (MU) method:

We propose to search for the best candidate (among C candidates) by shifting the bit-stream by the potential packet length and then calculating HEC over the next header sequence at the position indicated by corresponding candidate. If no candidate is correct (*i.e.*, corresponds to correct HEC) at the potential location of the next header sequence, then the candidate that gives one-bit error syndrome at the next header sequence is selected as the best candidate. In case of failure, the FS automaton switches to the *HUNT* state. The extended flow diagram for the *SYNC* state is shown in Figure 3.6.

Ueda's and MU methods, in the *HUNT* state, check for no errors in CRC calculations while shifting bit-by-bit or byte-by-byte, a situation impossible to appear at a very low *SNR*, when there would always be few errors in the header, thus making the method of hunting using HEC unreliable at low *SNR*. Furthermore, in case of low order HEC or short HEC meeting a configuration in which random bits emulate the header is more likely. These problems are addressed by jointly analyzing the successive packets in FS techniques presented in Chapters 4 and 5, and by using soft hunt operation in a robust 3S automaton presented in Chapter 6.

3.5 Simulation results

The considered simulator model consists of a burst generator, a BPSK modulator, a channel, and a receiver. Simulations are carried over both AWGN and Rayleigh fading channels. In the case of Rayleigh fading channel, the modulated signal is subject to zero mean and unit variance fast (bit) Rayleigh fading plus zero-mean AWGN noise. For performance analysis, Erroneous Packet Location Rate (EPLR) is evaluated as a function of the channel SNR. It should be noted that a packet is deemed correctly synchronized if both of its ends have been correctly determined.

The state-of-the-art variable-length FS techniques are relevant to the considered scenario of this thesis. Thus, simulation results for the variable-length FS techniques such as Ueda's method [17] and the NP FS method [21] would now be presented as they would serve as reference for the proposed FS techniques in the upcoming chapters. Since the MAC header uses an HEC of 8 bits only, a MU method would be applied. Simulation results for the conventional HD-based FS method, which makes use of the length field to perform FS, would also be provided.

In our simulations, we have chosen $L = 1800$ bytes, as by ignoring the header overhead, 1800 bytes in every WiMAX PHY frame of 20 ms can provide a raw data rate of 703.125 kbps, which, if fully dedicated, is enough for a real-time video service. MAC headers are constructed with the format defined in Section 2.2.1 and the payloads of MAC packets consist of a variable amount of randomly generated bytes. Data packets are randomly generated with a length uniformly distributed between $\ell_{\min} = 50$ bytes and $\ell_{\max} = 200$ bytes. If the generated packet is not insertable in the remaining space of a burst, a padding packet is inserted to fill the burst, which is then BPSK modulated before being sent over the channel.

At the receiver, BPSK demodulator is employed, which provides hard burst to MU and HD-based FS methods. NP FS method with equiprobable data symbols is simulated and the received noisy burst is directly used in the LRT of (3.8). Simulations for MU method are carried out with $\delta = 1$, *i.e.*, a single correct HEC evaluation brings the automaton from the *PRESYNC* state back to the *SYNC* state. Since we want the NP FS method to remain transmitter compliant, thus without inserting any additional SW, several header fields that remain unchanged are merged together to form a SW. Under the assumptions stated in Section 2.2.1, the merger of constant fields constitute a SW of 13 bits.

Simulation results for transmission over a AWGN channel are shown in Figure 3.7. MU method performs poorly at low SNR because of its CRC-based hunt operation, since errors in the header will cause wrong calculation of CRC, and hence miss detections. Furthermore, due to low order CRC, *i.e.*, CRC-8, FAs are also more likely. For the NP FS method, two curves with two probability of FA have been provided. The NP FS method with such a short SW of 13 bits has some FS error floor at high SNR, as expected, due to unavoidable and persistent high FA (emulation), as the payload data can simulate the SW more often due to the short size of the SW.

Simulation results for transmission over Rayleigh channel are shown in Figure 3.8. LRT for only AWGN channels are defined in NP FS method [19; 20; 21; 40; 22], thus simulation results over Rayleigh channel are shown only for HD-based and MU FS methods.

3.6 Conclusions

In this chapter we have introduced several state-of-the-art FS methods. We have explained an HEC-based 3S FS automaton, proposed initially for the FS of the fixed-length ATM cells and adopted in the Ueda's method for the variable-length packets. We have also explained NP FS method, which makes use of the embedded SW inside the bit-stream to perform FS. These FS techniques work on-the-fly, therefore allowing early decisions and introducing almost no latency.

Ueda's and NP FS methods suffer from performance degradation due to several reasons. First, miss detections are more likely at low SNR due to bit errors in HEC or SW. This drawback is more prominent in Ueda's method due to the use of hard HEC verification method. Second, in case of short SW/HEC compared to the header (which is the case in WiMAX MAC header), FAs are more often. Though, FAs can appear both in Ueda's and NP FS methods, but they are more notable in NP FS method due to an ease in emulation of a SW by random data.

In the following chapters, these issues are addressed by exploiting the soft information from the channel along with redundancy present in the header and by analyzing several successive packets before deciding their boundaries.

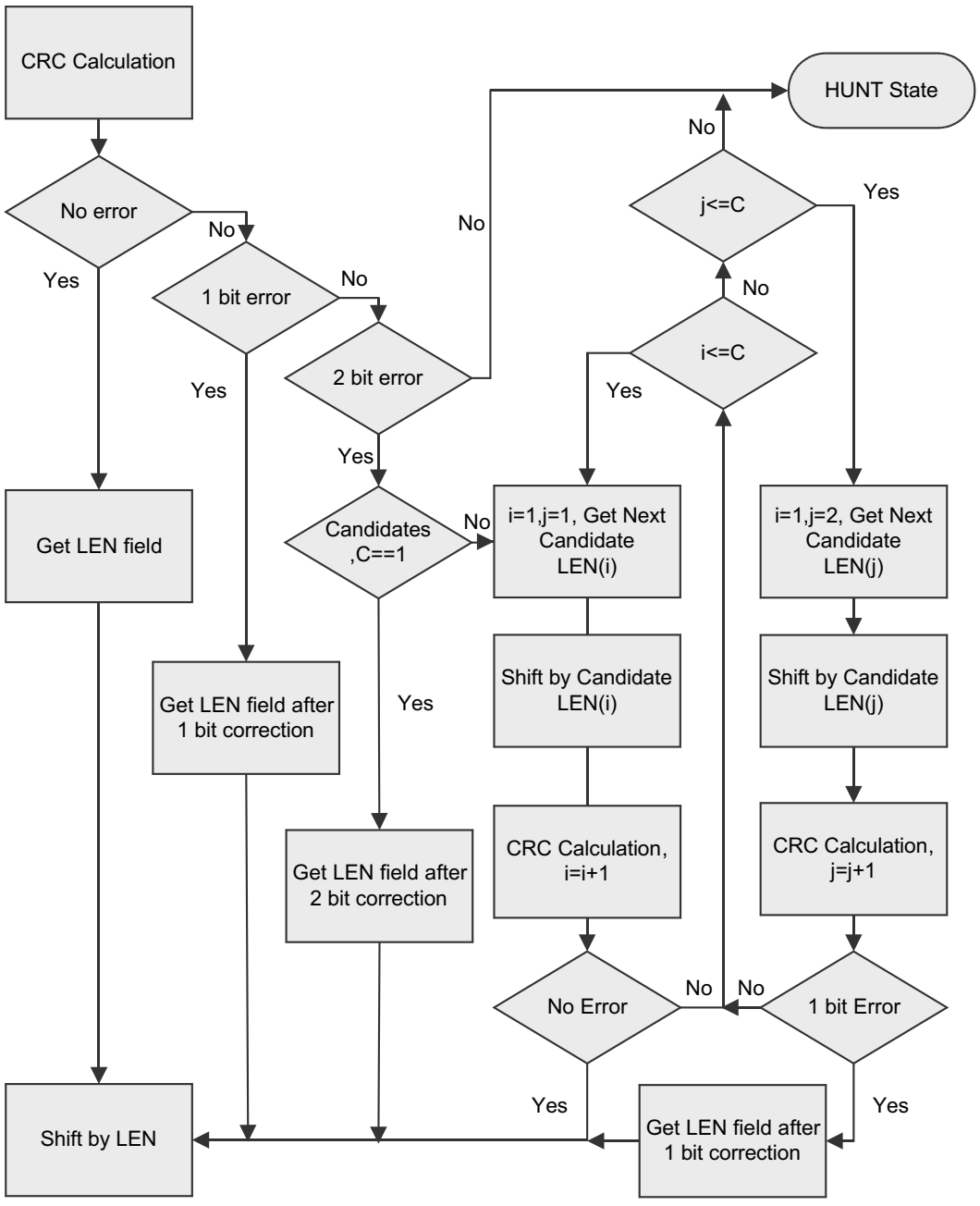

Figure 3.6: Extended flow diagram for *SYNC* state

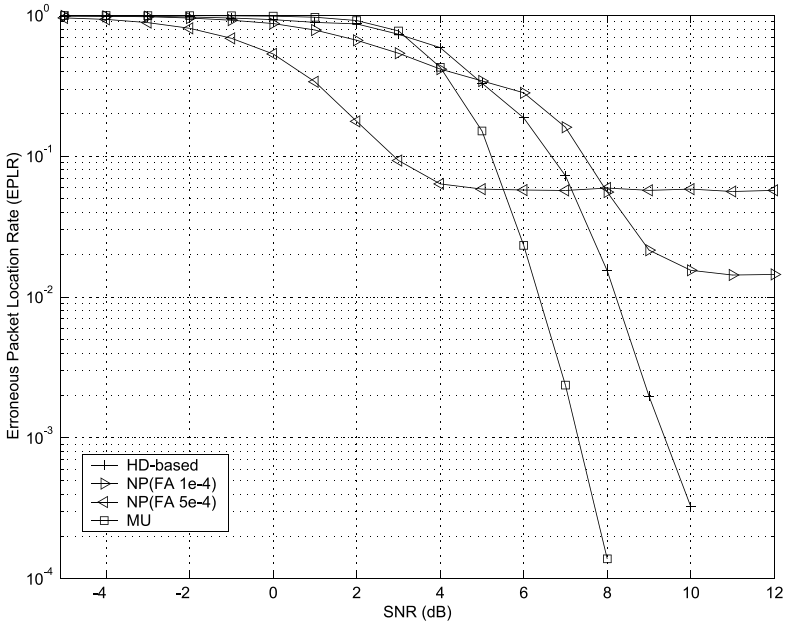

Figure 3.7: State-of-the-art FS methods for bursts transmitted over an AWGN channel

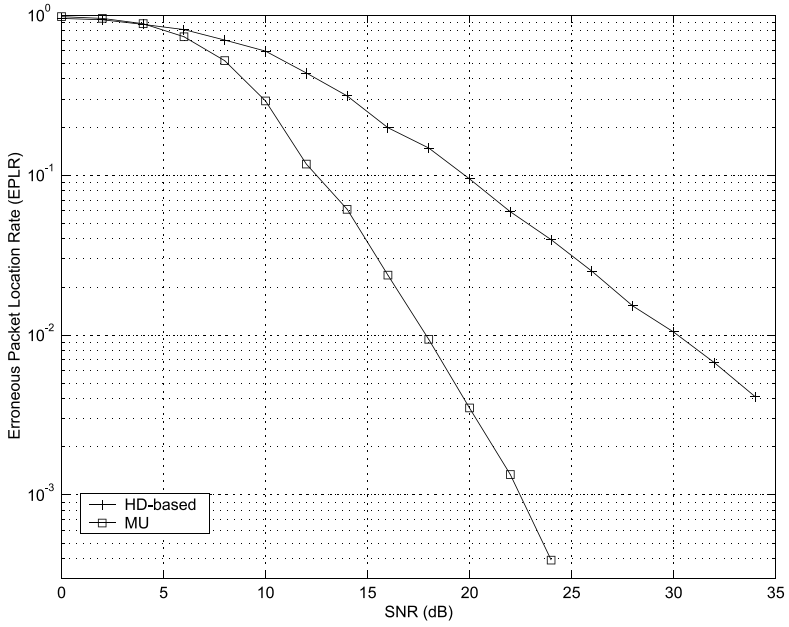

Figure 3.8: State-of-the-art FS methods for bursts transmitted over Rayleigh channel

Chapter 4

Trellis-based FS

The variable-length state-of-the-art FS techniques discussed in the previous chapter are totally non-beneficial for the S-PPS exploited in the WiMAX-WiFi inter-networking scenario under investigation in this thesis. In Ueda's method a single bit error in the packet header can cause a FS failure and loss of otherwise useful packet(s) reaching the higher protocol layers. Furthermore, it is not compatible with the flow of soft information. Though NP FS method utilize the soft information from the channel and after FS can forward s-packets to the higher layer, but can easily suffer from performance degradation due to miss detections and FAs, especially in case the SW is short, causing drop of several consecutive s-packet(s) possibly having error-free payload(s). Furthermore, to increase FS performance NP FS method requires an additional SW to be inserted at the start of each packet, causing additional signaling overhead and compatibility issues.

Due to the above-mentioned drawbacks, the state-of-the-art FS techniques fall short of the stated requirements/needs listed in the introduction of this thesis. This chapter along with the next two chapters intends to overcome these issues.

4.1 Introduction

As discussed in Chapter 1, JSCD techniques [8; 60; 61] exploit the correlation between the headers of successive packets at different protocol layers, in conjunction with soft information provided by the channel decoders at PHY layer, to recover the various headers of the protocol stack. They jointly exploit the structural properties of the protocol stack along with the CRC redundancy. FS can also benefit from these available redundancies in the sequentially aggregated packets to reduce the number of packets dropped in the protocol stack.

In this chapter we present a JPCD technique to perform FS, which utilize the protocol layer redundancies in conjunction with the channel likelihood provided by the channel decoder. It is a trellis-based FS method, where a successive packets to be synchronized are represented as a trellis, on which MAP decoder is applied to estimate the boundaries of the packets.

This chapter is organized as follows: First, some hypotheses on the structure of the header are presented in Section 4.2. The proposed trellis representation of the packet aggregation in a burst is presented in Section 4.3, before presenting the proposed MAP estimation for FS in Section 4.4. The proposed trellis-based FS is explained in Section 4.5. Simulation results are presented in Section 4.6 and conclusions in Section 4.7.

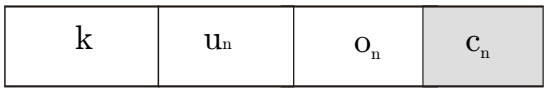

Figure 4.1: Types of Header fields

4.2 Header structure

The header $\mathbf{h}_n$ of the n-th variable packet may be partitioned into four fields, as shown in Figure 4.1, which are

1. The *constant* field $\mathbf{k}$, contains all bits that do not change from one packet to the next. It includes the SW indicating the beginning of a packet, and other bits that remain constant [8] once the communication is established.
2. The header is assumed to contain a *length* field $\mathbf{u}_n$, indicating the size of a packet in bits λ_n, including the header. Our task is to estimate the successive values taken by this quantity in all packets of the burst.
3. The *other* field $\mathbf{o}_n$, gathers all bits of the header that are not used to perform FS.
4. The HEC field $\mathbf{c}_n$ is assumed to cover only the header (*i.e.*, $\ell_h - \ell_c$ "working" bits) without covering the payload, *i.e.*,

$$\mathbf{c}_n = \mathbf{f}\left(\mathbf{k}, \mathbf{u}_n, \mathbf{o}\right), \tag{4.1}$$

where $\mathbf{f}$ is some (CRC or checksum) encoding function. The length λ_n of the n-th packet is assumed to be a realization of a stationary memoryless process Λ characterized by

$$\pi_\lambda = \Pr\left(\Lambda = \lambda\right) \neq 0 \text{ for } \ell_{\min} \leqslant \lambda \leqslant \ell_{\max}, \tag{4.2}$$

where $\ell_{\min}$ and $\ell_{\max}$ are the minimum and maximum length in bits of a packet. In what follows, the length in bits of a vector $\mathbf{z}$ is denoted by $\ell\left(\mathbf{z}\right)$ and its observation (soft information) provided either by a channel, a channel decoder, or a lower protocol layer is denoted by $\mathbf{y}_z$. $\mathbf{z}_a^b$ represents the sub-vector of $\mathbf{z}$ between indexes a and b (in bits).

One can identify several sources of redundancy present in the header of a packet. First, the inter-packet redundancy, available *e.g.*, due to the constant field [8; 102], which is due to the correlation between the headers of the successive packets. Second, the presence of the intra-packet redundancy, *e.g.*, due to CRCs or checksums, which can be exploited as error-correcting codes as in proposed [103; 8; 102].

These kind of redundancies are be used for header suppression and header compression, see *e.g.*, [104] for the suppression and compression of RTP, UDP, and IP headers transmitted over WiMAX. They can also be used to build some a priori information on the erroneous headers for their robust estimation [8; 61]. In this chapter we will utilize them for the robust FS of the aggregated packets.

4.3 Trellis Representation of a burst

Consider a burst of L bits consisting of N aggregated packets. This burst contains either $N-1$ *data* packets and an additional *padding* packet containing only padding bits, see Figure 4.2, or N *data* packets and no padding bits. Assume that each of these packets,

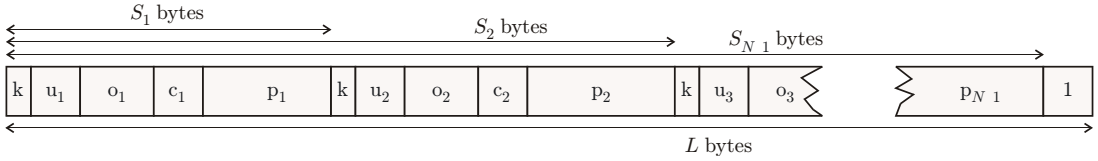

Figure 4.2: Aggregated packets in a WiMAX burst

except the padding packet, contains a header and a payload and follows the same syntax, as described in Section 4.2.

Assuming that L is fixed *before* packet aggregation and that N is not determined *a priori*, the accumulated length ℓ in bits of the n first aggregated packets may be described by a Markov process, which state is denoted by S_n. *A priori* state transition probabilities $p\left(S_n = \ell \,|\, S_{n-1} = \ell'\right)$ can be deduced from (4.2). If $\ell < L$, then

$$p\left(S_n = \ell \,|\, S_{n-1} = \ell'\right) = \begin{cases} \pi_{\ell-\ell'} & \text{if } \ell_{\min} \leqslant \ell - \ell' \leqslant \ell_{\max} \\ 0 & \text{else,} \end{cases} \tag{4.3}$$

and if $\ell = L$, then

$$p\left(S_N = L \,|\, S_{N-1} = \ell'\right) \quad = \quad \begin{cases} 0, \text{ if } L - \ell' > \ell_{\max} \\ 1, \text{ if } 0 < L - \ell' < \ell_{\min} \\ \sum\limits_{k=L-\ell'}^{\ell_{\max}} \pi_k, \text{ else.} \end{cases} \tag{4.4}$$

In (4.4), if $L - \ell' > \ell_{\max}$, there is enough space to put a data packet of any allowed length, thus the n-th packet cannot end the burst.. If $0 < L - \ell' < \ell_{\min}$, there is not enough space in the burst to put a data packet and the n-th packet is thus necessarily the (last) padding packet. In the other cases, only data and padding packets of $L - \ell'$ bits are allowed to finish the burst. This is possible when the n-th data packet is of $L - \ell'$ bits or if the n-th data packet generated by the source has a length strictly larger than $L - \ell'$ bits.

With these representations, the successive values taken by S_n can be described by a trellis such as that of Figure 4.3. This trellis is inspired from the one proposed in [28] for the robust decoding of variable-length encoded data. Each packet may be seen as a symbol encoded with a variable-length code. The main difference comes from the fact that the number of packets N is unknown and is only known to satisfy

$$N_{\min} \leqslant N \leqslant N_{\max}, \tag{4.5}$$

with $N_{\min} = \lceil L/\ell_{\max} \rceil$, $N_{\max} = \lceil L/\ell_{\min} \rceil$, and $\lceil \cdot \rceil$ denoting upward rounding. This is the reason for having non-singular (*i.e.*, $N_{\max} - N_{\min} + 1$) final states $S_n = L$. Furthermore, here only the packet boundaries are of interest and the number of padding bits is unknown at receiver side.

In this trellis, ℓ identifies a particular state S_n at *packet-clock* $n = 0, \ldots, N_{\max}$. If, at packet-clock n, $L - \ell \geq \ell_{\max}$ then the next packet is necessarily a data packet, otherwise the padding packet can also be present. The dashed transitions correspond to padding packets and the plain transitions correspond to data packets. For the last packet (when $S_n = L$), dashed and plain transitions may be parallel.

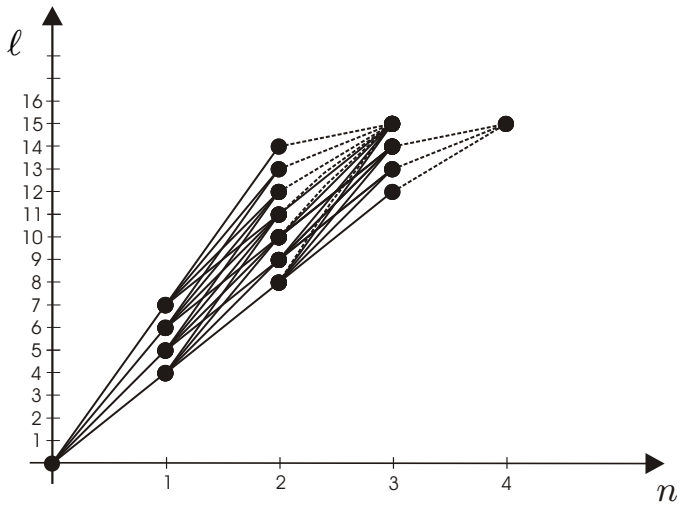

Figure 4.3: Trellis depicting the allowed total length in bits L vs. the number of packets n in a *burst* of $L = 15$ bits with $\ell_{\min} = 4$ bits and $\ell_{\max} = 7$ bits. Dashed lines correspond to padding bits.

4.4 MAP estimation for FS

This section formulates FS of variable-length packets as a MAP estimation problem.

4.4.1 Estimators for the number of packets and their boundaries

Consider a burst $\mathbf{x}_1^L$ of N aggregated packets and some vector $\mathbf{y}_1^L$ containing soft information about the bits of $\mathbf{x}_1^L$. (bit APPs, likelihood ratios, etc). The vector $\mathbf{y}_1^L$ may be obtained at the output of a channel, of a channel decoder, or of a lower transparent protocol layer [14]. Here, one assumes that the first entry of $\mathbf{y}_1^L$ corresponds to the first bit of $\mathbf{x}_1^L$. This assumption is further discussed in Section 4.5.4.

Based on the above trellis representation of the successive variable-length packets, various decoding strategies can be applied. We apply the MAP estimator. Our aim is to obtain joint MAP estimates $\widehat{N}$ of N and $\widehat{\lambda}_n$ of λ_n, $n = 1, \ldots, \widehat{N}$, using the knowledge of $\mathbf{y}_1^L$ and all identifiable sources of redundancy in the protocol description, *i.e.*,

$$\left(\widehat{N}, \widehat{\lambda}_1, \ldots, \widehat{\lambda}_{\widehat{N}}\right) = \arg \max_{N, \lambda_1, \ldots, \lambda_N} p\left(N, \lambda_1, \ldots, \lambda_N \,|\, \mathbf{y}_1^L\right). \tag{4.6}$$

Obtaining directly the solution to (4.6) is quite difficult. That is why, we resort to a suboptimal two-step estimator consisting in estimating N first and then in estimating the locations of the beginning and of the end of the packets. *i.e.*,

1. The MAP estimate $\widehat{N}_{\text{MAP}}$ of N is

$$\widehat{N}_{\text{MAP}} = \arg \max_{N_{\min} \leqslant n \leqslant N_{\max}} P\left(S_n = L | \mathbf{y}_1^L\right). \tag{4.7}$$

Once $\widehat{N}_{\text{MAP}}$ is obtained, the MAP estimate for the index ℓ_n of the last bit of the

n-th ($n = 1, ..., \widehat{N}_{\text{MAP}}$) packet is estimated as

$$\widehat{\ell}_n = \arg\max_{\ell} P\left(S_n = \ell | \mathbf{y}_1^L\right). \tag{4.8}$$

2. The length λ_n of the n-th packet is estimated as

$$\widehat{\lambda}_n = \arg\max_{\ell} P\left(S_n = \ell | \mathbf{y}_1^L\right) - \arg\max_{\ell} P\left(S_{n-1} = \ell | \mathbf{y}_1^L\right), \tag{4.9}$$

with $n = 1, ..., \widehat{N}_{\text{MAP}}$.

This two-step estimator is suboptimal when compared to the joint MAP estimator (4.6), since the length of the n-th data packet estimated using (4.9) may not necessarily satisfy the length constraint (4.2).

The segmentation of packets aggregated within a burst requires the evaluation in (4.7), (4.8), and (4.9), of APPs $P(S_n = \ell | \mathbf{y}_1^L)$ for all possible values of n and ℓ, which is performed in the next Section.

4.5 Trellis-based FS Algorithm

The trellis representation of Section 4.3 also allows the application of the BCJR algorithm [29] to perform MAP decoding, by evaluating first the joint probability

$$\begin{aligned} P(S_n = \ell, \mathbf{y}_1^L) &= P(S_n = \ell, \mathbf{y}_1^\ell, \mathbf{y}_{\ell+1}^L) \\ &= P(S_n = \ell, \mathbf{y}_1^\ell) P(\mathbf{y}_{\ell+1}^L | S_n = \ell, \mathbf{y}_1^\ell) \\ &= \alpha_n(\ell)\, \beta_n(\ell), \end{aligned} \tag{4.10}$$

where,

$$\alpha_n(\ell) = P(S_n = \ell, \mathbf{y}_1^\ell)$$

and

$$\beta_n(\ell) = P(\mathbf{y}_{\ell+1}^L | S_n = \ell).$$

4.5.1 Evaluation of α_n and β_n

Now for $n = 1, ..., N_{\max}$, classical BCJR forward and backward recursions may be performed, with

$$\alpha_n(\ell) = \sum_{\ell'} \alpha_{n-1}(\ell') \gamma_n(\ell', \ell), \tag{4.11}$$

$$\beta_n(\ell) = \sum_{\ell'} \beta_{n+1}(\ell') \gamma_{n+1}(\ell, \ell'), \tag{4.12}$$

and

$$\gamma_n\left(\ell', \ell\right) = P(S_n = \ell, \mathbf{y}_{\ell'+1}^\ell | S_{n-1} = \ell'). \tag{4.13}$$

For the forward recursion the value of S_0 is perfectly known, leading to $\alpha_0(\ell = 0) = 1$ and $\alpha_0(\ell \neq 0) = 0$. For the backward recursion, the number of packets is only known to

satisfy (4.5), thus there are $N_{\max} - N_{\min} + 1$ allowed final states $S_n = L$. We consider two options for the initialization of $\beta_n(L)$.

1. *Coarse* initialization (CI): Assuming all allowed final states as equally likely (which is a quite coarse approximation), one gets

$$\beta_n(L) = \frac{1}{N_{\max} - N_{\min} + 1},\ N_{\min} \leqslant n \leqslant N_{\max}. \tag{4.14}$$

   All other values of $\beta_n(L)$, for $\ell < L$ are initialized to 0.

2. *Precise* initialization (PI): Using (4.3) and (4.4), more accurate initial values may be obtained as

$$\beta_n(L) = P(S_n = L),\ N_{\min} \leqslant n \leqslant N_{\max}, \tag{4.15}$$

where $P(S_n = \ell)$ is the *a priori* probability that the n-th packet ends at ℓ-th bit (or that the $n+1$-th packet starts at $(\ell+1)$-th bit) of a burst. $P(S_n = \ell)$ satisfies

$$P(S_n = \ell) = \sum_{\ell'} P(S_n = \ell | S_{n-1} = \ell') P(S_{n-1} = \ell'), \tag{4.16}$$

which may be evaluated iteratively with the help of (4.3) and (4.4), starting from $n = 1$ till $n = \lceil \ell / \ell_{\min} \rceil$, with initial condition $P(S_0 = 0) = 1$ and $P(S_0 \neq 0) = 0$.

4.5.2 Evaluation of γ_n

When performing the evaluation of $\gamma_n(\ell', \ell)$, one implicitly assumes that the n-th packet starts at the $(\ell'+1)$-th bit and ends at the ℓ-th bit of a burst. When $\ell < L$ and provided that $\ell_{\min} \leqslant \ell - \ell' \leqslant \ell_{\max}$, the n-th packet is not the last one, thus only data packets have to be considered. When $\ell = L$, depending on the value of ℓ', data and padding packets have to be considered simultaneously (parallel plain and dashed transitions in Figure 4.3) or only padding packets have to be taken into account (dashed transitions in Figure 4.3).

Many sources of redundancy, as mentioned in Section 4.2, may be taken into account for the evaluation of $\gamma_n(\ell', \ell)$. If, the n-th packet is a data packet, the first bits are determined and equal to $\mathbf{k}$, the length field $\mathbf{u}_n$ is also determined and its content should represent $\ell - \ell'$ bits. Albeit the content of the *other* field $\mathbf{o}$ is not determined, the value of the HEC $\mathbf{c}$ is strongly influenced by $\mathbf{k}$ and $\mathbf{u}_n$. When the n-th packet is the padding packet, all the bits are perfectly determined (we may assume, without loss of generality that they are all equal to 1). Taking this into account, two cases have to be considered for the evaluation of $\gamma_n(\ell', \ell)$.

4.5.2.1 First case, $\ell < L$

In this case, the transition corresponding to the n-th packet cannot be the last one, thus corresponds to a data packet. Assuming that $\ell_{\min} \leqslant \ell - \ell' \leqslant \ell_{\max}$, the bits between $\ell' + 1$ and ℓ may be interpreted as $\mathbf{x}_{\ell'+1}^{\ell} = [\mathbf{k}, \mathbf{u}_n, \mathbf{o}, \mathbf{c}, \mathbf{p}]$, where $\mathbf{u}_n = \mathbf{u}(\ell - \ell')$ is the binary representation of $\ell - \ell'$. The corresponding observation may be written as $\mathbf{y}_{\ell'+1}^{\ell} = [\mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c, \mathbf{y}_p]$. With these notations, for $\ell \neq L$, the transition metric $\gamma_n(\ell', \ell) =$

$\gamma_n^{\mathrm{d}}(\ell',\ell)$ only accounting for data packets may be written as

$$\begin{aligned}\gamma_n^{\mathrm{d}}(\ell',\ell) &= P(S_n=\ell,\mathbf{y}_{\ell'+1}^{\ell}|S_{n-1}=\ell') \\ &= \sum_{\mathbf{x}_{\ell'+1}^{\ell}\neq\mathbf{1}} P\left(S_n=\ell,\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}|S_{n-1}=\ell'\right) \\ &= p\left(S_n=\ell|S_{n-1}=\ell'\right)\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right). \end{aligned} \tag{4.17}$$

Where,

$$\begin{aligned}\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right) &= \sum_{\mathbf{x}_{\ell'+\mathbf{1}}^{\ell}\neq\mathbf{1}} P\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}|S_{n-1}=\ell',S_n=\ell\right) \\ &= \sum_{\mathbf{x}_{\ell'+\mathbf{1}}^{\ell}\neq\mathbf{1}} P\left(\mathbf{y}_{\ell'+1}^{\ell}|\mathbf{x}_{\ell'+1}^{\ell},S_{n-1}=\ell',S_n=\ell\right) \\ &\quad P\left(\mathbf{x}_{\ell'+1}^{\ell}|S_{n-1}=\ell',S_n=\ell\right). \end{aligned} \tag{4.18}$$

The sum in (4.18) is over all possible $\mathbf{x}_{\ell'+1}^{\ell}$, except $\mathbf{x}_{\ell'+1}^{\ell}=\mathbf{1}$, which corresponds to the content of a padding packet. Nevertheless, only the $\mathbf{x}_{\ell'+1}^{\ell}$s starting with $\mathbf{k}$ and $\mathbf{u}_n=\mathbf{u}(\ell-\ell')$ have to be considered since these fields are fully determined.

$$\begin{aligned}\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right) &= P(\mathbf{y}_k|\mathbf{k})\,P\left(\mathbf{y}_u|\mathbf{u}\left(\ell-\ell'\right)\right) \\ &\quad \sum_{\mathbf{o},\mathbf{c},\mathbf{p}} P\left(\mathbf{y}_o,\mathbf{y}_c,\mathbf{y}_p|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o},\mathbf{c},\mathbf{p}\right)P\left(\mathbf{o},\mathbf{c},\mathbf{p}|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right)\right). \end{aligned} \tag{4.19}$$

Moreover, assuming that the channel is memoryless and taking into account the fact that $\mathbf{k},\mathbf{u}_n,\mathbf{o}$, and $\mathbf{c}$ do not depend on $\mathbf{p}$, we get

$$\begin{aligned}&\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right) = P(\mathbf{y}_k|\mathbf{k})\,P\left(\mathbf{y}_u|\mathbf{u}\left(\ell-\ell'\right)\right)\sum_{\mathbf{p}}P(\mathbf{y}_p|\mathbf{p})\,P(\mathbf{p}) \\ &\sum_{\mathbf{o},\mathbf{c}} P\left(\mathbf{y}_o,\mathbf{y}_c|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o},\mathbf{c}\right)P\left(\mathbf{o},\mathbf{c}|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right)\right) \\ &= P(\mathbf{y}_k|\mathbf{k})\,P\left(\mathbf{y}_u|\mathbf{u}\left(\ell-\ell'\right)\right)\sum_{\mathbf{p}}P(\mathbf{y}_p|\mathbf{p})\,P(\mathbf{p}) \\ &\sum_{\mathbf{o},\mathbf{c}} P(\mathbf{y}_o|\mathbf{o})\,P\left(\mathbf{y}_c|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o},\mathbf{c}\right)P(\mathbf{o})\,P\left(\mathbf{c}|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o}\right). \end{aligned} \tag{4.20}$$

Finally, using the fact that the HEC $\mathbf{c}$ is fully determined by $\mathbf{k}$, $\mathbf{u}_n$, and $\mathbf{o}$ (*i.e.*, $P\left(\mathbf{c}|\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o}\right)=1$), (4.20) simplifies to

$$\begin{aligned}\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right) &= P(\mathbf{y}_k|\mathbf{k})\,P\left(\mathbf{y}_u|\mathbf{u}\left(\ell-\ell'\right)\right)\sum_{\mathbf{p}}P(\mathbf{y}_p|\mathbf{p})\,P(\mathbf{p}) \\ &\quad \sum_{\mathbf{o}} P(\mathbf{y}_o|\mathbf{o})\,P\left(\mathbf{y}_c|\mathbf{c}=\mathbf{f}\left(\mathbf{k},\mathbf{u}\left(\ell-\ell'\right),\mathbf{o}\right)\right)P(\mathbf{o}). \end{aligned} \tag{4.21}$$

Under the assumptions above, and for a memoryless AWGN channel with variance σ^2, one

gets

$$P\left(\mathbf{y}_u|\mathbf{u}\left(\ell-\ell'\right)\right)=P\left(\mathbf{y}_u|\mathbf{u}_n\right)=\prod_{i=1}^{\ell(\mathbf{u}_n)}\frac{1}{\sqrt{2\pi}\sigma}e^{-(\mathbf{y}_u(i)-\mathbf{u}_n(i))^2/2\sigma^2}.$$

Assuming that the values taken by $\mathbf{p}$ are all equally likely, one gets $P\left(\mathbf{p}\right)=2^{-\ell(\mathbf{p})}$, where $\ell\left(\mathbf{p}\right)$ is the length of $\mathbf{p}$.

In (4.21), the sum over all possible $\mathbf{o}$ may be quite complex to evaluate for long $\mathbf{o}$. It may be calculated using a trellis construction consisting in iteratively grouping the combinations of $\mathbf{o}$ leading to the same HEC, as proposed in [8], with a complexity of $\mathcal{O}(\ell\left(\mathbf{o}\right)2^{\ell(\mathbf{c})})$. A reduced-complexity algorithm for evaluating this sum can also be found in [8]. The calculation of $\gamma_n\left(\ell',\ell\right)$ for each transition has a similar complexity.

4.5.2.2 Second case, $\ell = L$

When $\ell=L$ and $L-\ell'<\ell_{\max}$, the n-th packet is the last one, and $\mathbf{x}_{\ell'+1}^{L}=\mathbf{1}$ has also to be considered in $\gamma_n\left(\ell',L\right)$, leading to

$$\begin{aligned}\gamma_n\left(\ell',L\right)&=\sum_{p=0,1}P(S_n=L,\mathbf{y}_{\ell'+1}^{L},P_n=p|S_{n-1}=\ell')\\&=\gamma_n^{\mathrm{d}}\left(\ell',L\right)P\left(P_n=0|S_{n-1}=\ell'\right)\\&+\gamma_n^{\mathrm{p}}\left(\ell',L\right)P\left(P_n=1|S_{n-1}=\ell'\right).\end{aligned}\tag{4.22}$$

Where P_n is a random variable indicating whether the n-th packet is a padding packet and its *a priori* probability is given by

$$P\left(P_n=1|S_{n-1}=\ell'\right)\;=\;\begin{cases}0,\text{ if }L-\ell'\geq\ell_{\max}\\1,\text{ if }0<L-\ell'<\ell_{\min}\\\sum\limits_{\lambda=L-\ell'+1}^{\ell_{\max}}\pi_\lambda,\text{ else,}\end{cases}\tag{4.23}$$

where, we have used the fact that the n-th packet is a padding packet if at state $S_{n-1}=\ell'$ the source generates a data packet of size strictly larger than $L-\ell'$ bits. Similarly, the *a priori* probability that the n-th packet is a data packet is given by

$$P\left(P_n=0|S_{n-1}=\ell'\right)\;=\;\begin{cases}0,\text{ if }0<L-\ell'<\ell_{\min}\\1,\text{ if }L-\ell'\geq\ell_{\max}\\\sum\limits_{\lambda=\ell_{\min}}^{L-\ell'}\pi_\lambda,\text{ else.}\end{cases}\tag{4.24}$$

In ((4.22)),

$$\begin{aligned}\gamma_n^{\mathrm{p}}\left(\ell',\ell\right)&=p\left(S_n=L,\mathbf{y}_{\ell'+1}^{L}|S_{n-1}=\ell',P_n=1\right)\\&=p\left(S_n=L|S_{n-1}=\ell',P_n=1\right)\\&\quad P\left(\mathbf{y}_{\ell'+1}^{L}|P_n=1,S_{n-1}=\ell',S_n=L\right)\end{aligned}$$

accounts for the padding packet. Given the fact that the n-th packet is a padding packet, *i.e.*, $P_n=1$, one can easily see from the trellis of Figure 4.3, that

$$p\left(S_n=L|S_{n-1}=\ell',P_n=1\right)=1.$$

Thus, one gets

$$\gamma_n^{\mathrm{p}}\left(\ell',\ell\right) \;=\; P\left(\mathbf{y}_{\ell'+1}^{L}|P_n=1,S_{n-1}=\ell',S_n=L\right). \tag{4.25}$$

While,

$$\begin{aligned}\gamma_n^{\mathrm{d}}\left(\ell',L\right) &= p\left(S_n=L,\mathbf{y}_{\ell'+1}^{L}|S_{n-1}=\ell',P_n=0\right)\\ &= p\left(S_n=L|S_{n-1}=\ell',P_n=0\right)P\left(\mathbf{y}_{\ell'+1}^{L}|P_n=0,S_{n-1}=\ell',S_n=L\right)\\ &= p\left(S_n=L|S_{n-1}=\ell',P_n=0\right)\sum_{\mathbf{x}_{\ell'+1}^{L}\neq\mathbf{1}}P\left(\mathbf{y}_{\ell'+1}^{L},\mathbf{x}_{\ell'+1}^{L}|S_{n-1}=\ell',S_n=L\right)\\ &= p\left(S_n=L|S_{n-1}=\ell',P_n=0\right)\varphi\left(\mathbf{y}_{\ell'+1}^{\ell},\mathbf{x}_{\ell'+1}^{\ell}\right)\end{aligned}$$

accounts for the data packet, where

$$p\left(S_n=L|S_{n-1}=\ell',P_n=0\right)=\frac{\pi_{L-\ell'}}{\sum\limits_{\lambda=\ell_{\min}}^{L-\ell'}\pi_\lambda},$$

because of the fact that if n-th packet is a data packet, *i.e.*, $P_n=0$, we have $L-\ell'-\ell_{\min}+1$ number of possible plain transitions originating from the state $S_{n-1}=\ell'$, including the one ending at state $S_n=L$, see Figure 4.3.

4.5.3 Complexity evaluation

The complexity of the FS algorithm described in Section 4.4 is proportional to the number of nodes or to the number of transitions within the trellis on which FS is performed. From Figure 4.3, one sees that the trellis is lower-bounded by the line $\ell=n\ell_{\min}$ and upper-bounded by the lines $\ell=n\ell_{\max}$ and $\ell=L$. This region may be divided into two triangular sub-regions, one with $0\leq n\leq N_{\min}-1$, bounded between $\ell=n\ell_{\min}$ and $\ell=n\ell_{\max}$ and the other with $N_{\min}\leq n\leq N_{\max}-1$, bounded between $\ell=n\ell_{\min}$ and $\ell=L$. Thus, summing the number of nodes in each sub-region, one gets the number of nodes in the trellis,

$$\mathcal{N}_n \;=\; \sum_{n=0}^{N_{\min}-1} n\left(\ell_{\max}-\ell_{\min}\right)+\sum_{n=N_{\min}}^{N_{\max}}\left(L-n\ell_{\min}\right). \tag{4.26}$$

Taking $N_{\min}\approx L/\ell_{\max}$ and $N_{\max}\approx L/\ell_{\min}$, ((4.26)) simplifies to

$$\mathcal{N}_{\mathrm{n}}=\frac{L^2}{2}\left(\frac{\ell_{\max}-\ell_{\min}}{\ell_{\max}\ell_{\min}}\right)=\mathcal{O}\left(L^2\right). \tag{4.27}$$

From each node, at most $\ell_{\max}-\ell_{\min}$ transitions may emerge. Thus, from (4.27), the number of transitions $\mathcal{N}_{\mathrm{t}}$ may also be approximated as

$$\mathcal{N}_{\mathrm{t}}=\mathcal{O}\left(L^2\right). \tag{4.28}$$

4.5.4 Limitations

The trellis-based FS technique presented in this section is an hold-and-sync FS technique, which requires the soft information about the whole burst to be received before performing FS. Furthermore, it requires knowledge of (i) the beginning and (ii) length of the burst to perform FS. These two hypotheses require an error-free decoding of the headers of lower protocol layers, which contain this information. This may be done using methods presented in [8], which enable the lower layer to forward the burst to the layer where it is processed.

The main drawback of the proposed FS technique in terms of implementation is the increase in memory requirements for storing the soft information, estimated in [14; 12] to be three to four times that of storing hard bits. Moreover, buffering the burst induces some buffering and processing delays proportional to L^2, see (4.27) and (4.28). To alleviate these problems, a low-delay and less-complex variant of the preceding FS algorithm is proposed in the next chapter.

4.6 Simulation results

We will now provide simulation results for the trellis-based FS technique and compare it with the state-of-the-art FS techniques explained in Chapter 3. The simulator model and the channel specifications as detailed in Section 3.5 are used for the simulation results.

To use the trellis-based FS algorithm introduced in this chapter, the size of the burst L is required. It is transmitted in DL-MAP for WiMAX, which is protected with a more robust modulation and coding scheme. In our simulation we have assumed that the burst size L is received without any error. Furthermore, since WiMAX MAC packets are byte-aligned, *i.e.*, the LEN field of a packet is in bytes and all MAC packets contain an integer number of bytes, α, β, and γ are evaluated for ℓs corresponding to beginning of bytes.

Figure 4.4 provides the simulation results for the trellis-based FS technique, where random sized data packets with $\ell_{\min} = 50$ bytes and $\ell_{\max} = 200$ bytes are aggregated in bursts of $L = 1800$ bytes (a burst contains thus between 9 and 36 data packets). Curves for the conventional HD-based FS and the state-of-the-art FS techniques (*i.e.*, on-the-fly MU and NP FS methods, explained in Chapter 3), serving as references, are also shown. A gain of more than 8 dB in SNR is achieved with the proposed trellis-based FS techniques compared to the HD-based FS using the CI (4.14) of β. Whereas, a little higher gain of 8.5 dB in SNR is observed while using the PI (4.15).

Figure 4.5 performs the same comparison over Rayleigh channels for $\ell_{\min} = 50$ bytes and $\ell_{\max} = 200$ bytes and bursts of $L = 1800$ bytes.

4.7 Conclusions

We have extended the JSCD ideas to perform JPCD for the robust FS of aggregated packets. For that purpose, a novel trellis-based packet segmentation method is deployed for FS. It exploits all sources of redundancy present in the packet header (known fields, presence of CRC or checksums, *etc*) and soft information provided by the channel.

For the case of WiMAX, we have shown that all successions of MAC packets within a burst may be represented by a trellis on which a variant of the BCJR algorithm has been put at work to robustly identify the boundaries of each MAC packet. For similar MAC packet localization error, on an AWGN channel, a gain of about 8 dB in SNR compared

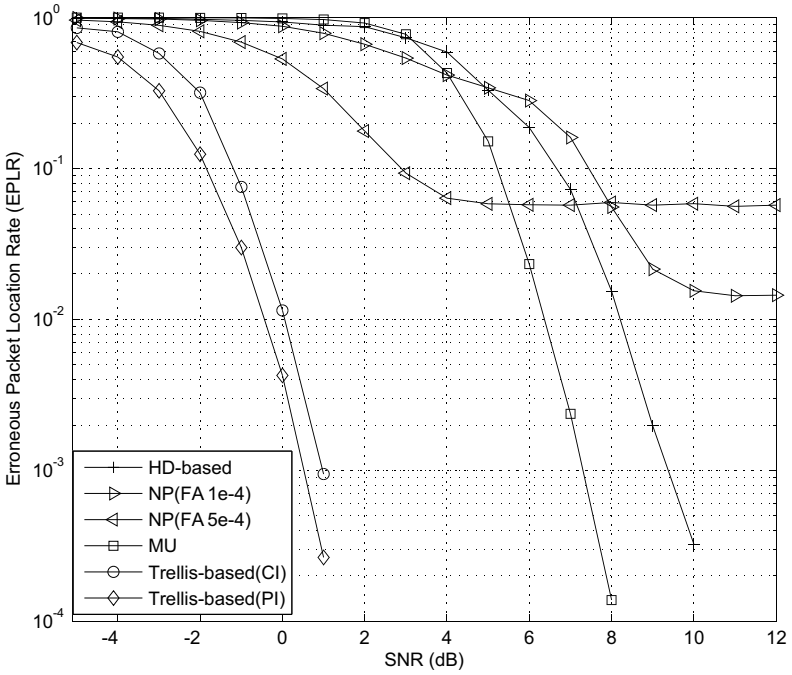

Figure 4.4: Comparison of the proposed trellis-based FS technique with HD-based, NP, and MU FS methods for bursts transmitted over an AWGN channel

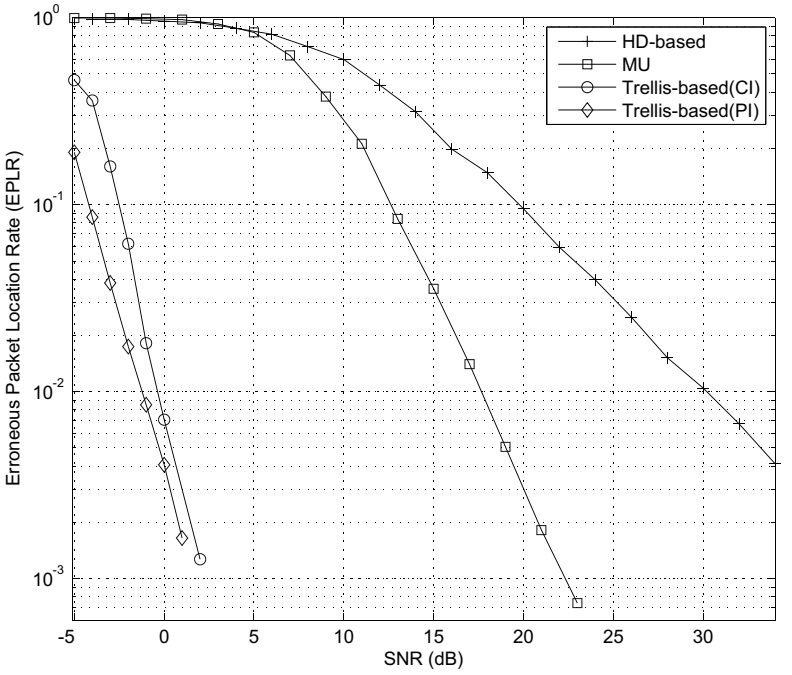

Figure 4.5: Comparison of the proposed trellis-based FS technique with HD-based and MU FS methods for bursts transmitted over Rayleigh channel

to the conventional HD-based technique is observed. The trellis-based technique provides the best performance as expected, but at the cost of a delay equivalent to the length of a burst. Thus, in the next chapter we propose necessary modifications to it to develop a low-delay and reduced-complexity FS technique.

Chapter 5

Sliding Trellis-based FS

The WiMAX standard explicitly supports QoS differentiation. Its QoS architecture facilitates multimedia communication, file transfer, or interactive applications, which have different QoS requirements in terms of bandwidth, delay, and jitter. And, among all QoS requirements, the maximum latency is most critical to the quality of time-sensitive multimedia applications and thus should be properly satisfied [80]. MPEG video with 50 fps, needs a variable bit rate frames every 20 ms, and interactive gaming or voice/video telephony needs about $32 - 512$ kbps with low latency requirements of about $80 - 160$ ms [105].

The trellis-based FS technique presented in the previous chapter has to wait for the complete burst before performing FS, the delay incurred can surpass the delay constrain of the video decoder functioning at APL layer. For WiMAX, OFDM PHY frame duration can *e.g.*, range from 2 ms to 20 ms [32]. By consider a worst case of having a single large burst in the DL sub-frame, with DL-to-UL ratio of 3:1, the maximum duration of this burst can range from 1.5 ms to 15 ms. Thus, a maximum additional latency of about 15 ms could be caused by the trellis-based FS, which might not surpass the QoS requirements for certain services that do not have strict latency requirements and prefer high performance FS. But for the low latency services like video telephony, this additional latency should be avoided. In this chapter, we provide a low-delay and reduced-complexity variant of the trellis-based FS technique.

5.1 Introduction

Sliding Window variants of the BCJR algorithm (*i.e.*, SW-BCJR) [106; 30], [31] are interesting when one is concerned with the latency introduced by the BCJR algorithm. These techniques have been proposed for the decoding of CCs. A classical BCJR decoding is done within a window, which at each step is shifted bit-by-bit [106; 30] or by several bits [31]. From one window to the next one, the results obtained during the forward iteration may be reused, contrary to those for the backward iteration. The number of bits the window is shifted at each iteration determines the trade-off between latency/complexity and efficiency.

The trellis-based FS technique requires that the receiver must wait (and need storage buffer) for whole burst thus the trellis-based FS needs minor modification to overcome this limitation. The problem comes from the fact that the trellis-based FS technique is based on the assumption that the beginning of a burst is well known at the receiver,

either due to synchronizer at lower layer or through error-free reception of the location of the beginning of the burst, *e.g.*, in WiMAX MAC layer the location of each burst is transmitted in FCH/DL-MAP using more robust modulation and coding scheme. Thus, the said assumption should be overcome to provide a trade-off between latency/complexity and efficiency. Any such modification would not only reduce the latency/complexity but would also enable it to function in the situations when the synchronizer is only activated to retrieve the lost FS.

This chapter proposes an adaptation of the reduced-complexity SW-BCJR [30], to develop a low-delay and reduced-complexity version of the trellis-based FS technique. The method is called ST-based as it works by dividing the burst into an overlapping segments (called windows), which are sequentially synchronized by sliding, window-by-window, a modified version of the trellis.

This chapter is organized as follows: The proposed low-delay and reduced-complexity ST-based FS technique is presented in Section 5.2. It is illustrated in Section 5.3 with the FS of WiMAX MAC packets aggregated in bursts, which are transmitted to the PHY layer. Finally, conclusions are drawn in Section 5.4.

5.2 Sliding Trellis-based FS Algorithm

Contrary to the trellis for a CC [29], the trellis considered in Figure 4.3 has a variable number of states for each value of the packet-clock n. One may apply directly the SW-BCJR ideas, but due to the increase of the size of the trellis (at least for small values of n), this would still need large trellises to be manipulated, with an increased computation time. Here, a ST-based approach is introduced: a reduced-size trellis is considered in each decoding window. As in [31], some overlapping between windows is considered to allow better reuse of already computed quantities reaching to a complexity-efficiency trade-off.

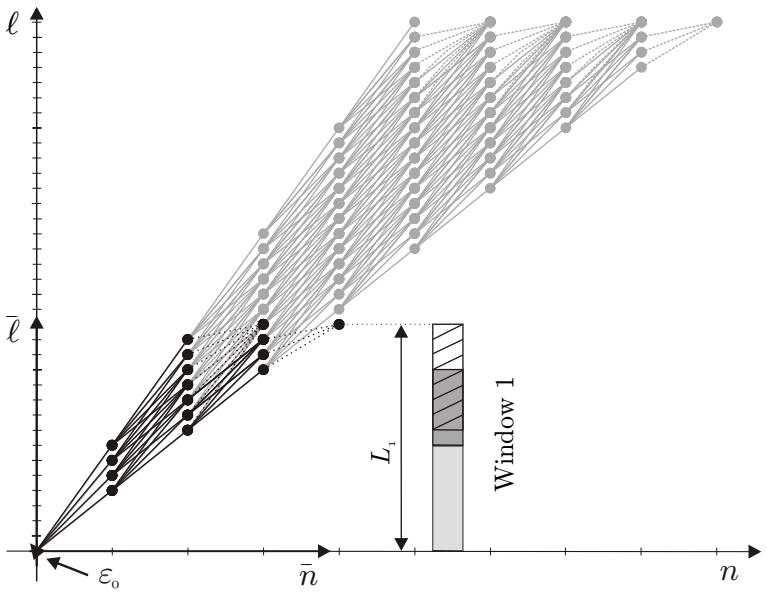

Figure 5.1: ST for the first decoding window, the original trellis is in gray

5.2.1 Sliding Trellis

In the proposed ST-based approach, a burst of L bits is divided into M overlapping windows with sizes L_m, $m = 1, ..., M$. For each of these windows, the bits from $\varepsilon_{m-1} + 1$ to $\varepsilon_{m-1} + L_m$ are considered, where ε_{m-1} is the index of the last bit of the last packet deemed reliably synchronized in the $m-1$-th window.

A ST moves from window-to-window to perform decoding. One such ST is shown in Figure 5.1. Let $\bar{n}$ and $\bar{\ell}$ be the local trellis coordinates. Once $P(S_{\bar{n}}^m = \bar{\ell} | \mathbf{y}_{\varepsilon_{m-1}+1}^{\varepsilon_{m-1}+L_m})$ is evaluated, one can apply the estimators (4.7), (4.8), and (4.9) to determine the number of packets $\widehat{N}_m$ in the m-th window (including the last truncated packet), the beginning, and the length of each packet. For the m-th window, $m < M$, among the $\widehat{N}_m$ decoded packets, only the first $\widehat{N}_m^c$ packets are considered as reliably synchronized, since enough data and redundancy properties have been taken into account. Truncated packets, especially when the HEC has been truncated, and the packet immediately preceding such packets, are not considered reliable. For this reason, only the $\widehat{N}_m^c$ complete packets ending in the first $L_m - \ell_{\max} - \ell_h$ bits of the window are considered as reliable. The unreliable region towards the boundary of the window is dashed in Figure 5.1.

The initialization of β is performed as in Section 4.5, since no knowledge from the previous window can be exploited. The initialization of α and the evaluation of γ towards the window boundary may depend on the location of the window inside a burst. Three types of window locations are considered: the *first* window at the start of a burst, the *intermediate* windows in the middle of the burst, and the *last* window at the end of the burst.

First Window, $m = 1$

Consider the first window of $L_m < L$ bits for which $\varepsilon_0 = 0$, shown in Figure 5.1. The ST representing all possible successions of packets within L_m bits is very similar to the trellis in Figure 4.3. The decoding approach, including the initialization of α for this first window is similar to that presented for the trellis-based approach in the previous chapter. An exception is the computation of $\gamma_{\bar{n}}\left(\bar{\ell}', \bar{\ell}\right)$, where two cases have again to be considered corresponding to

1. the normal data packets, leading to $\gamma_{\bar{n}}^{\mathrm{d}}\left(\bar{\ell}', L_m\right)$, and
2. the truncated data packets towards the boundary of the window, leading to $\gamma_{\bar{n}}^{\mathrm{t}}\left(\bar{\ell}', L_m\right)$, detailed in Section 5.2.2.

Intermediate Windows, $1 < m < M$

Consider now the m-th window ($m > 1$) containing the bits from $\varepsilon_{m-1} + 1$ to $\varepsilon_{m-1} + L_m < L$, see Figure 5.2. The bit index ε_{m-1}, of the last bit of the last packet deemed reliably synchronized (*i.e.*, the $\widehat{N}_{m-1}^c$-th packet) in the $m-1$-th window, corresponds in the local coordinates of the $m-1$-th ST to $\bar{\ell} = \varepsilon_{m-1} - \varepsilon_{m-2}$. The m-th ST starts at the local coordinates $\left(\bar{n} = \widehat{N}_{m-1}^c, \bar{\ell} = \varepsilon_{m-1} - \varepsilon_{m-2}\right)$ of $m-1$-th ST. The computation of $\gamma_{\bar{n}}\left(\bar{\ell}', \bar{\ell}\right)$ for an intermediate window is identical to that of the first window. A first choice for the initialization of $\alpha_{\bar{n}}^m\left(\bar{\ell}\right)$ in this ST, would be to consider $\alpha_0^m\left(\bar{\ell} = 0\right) = 1$ and $\alpha_0^m\left(\bar{\ell} \neq 0\right) = 0$. The drawback of this approach is that all computations of α performed in the $m-1$-th window are not utilized and are lost. Therefore, following the idea of the SW-BCJR decoder [106; 30], up to $\ell_{\max}$ initial values for $\alpha_0^m\left(\bar{\ell}\right)$ are propagated from the

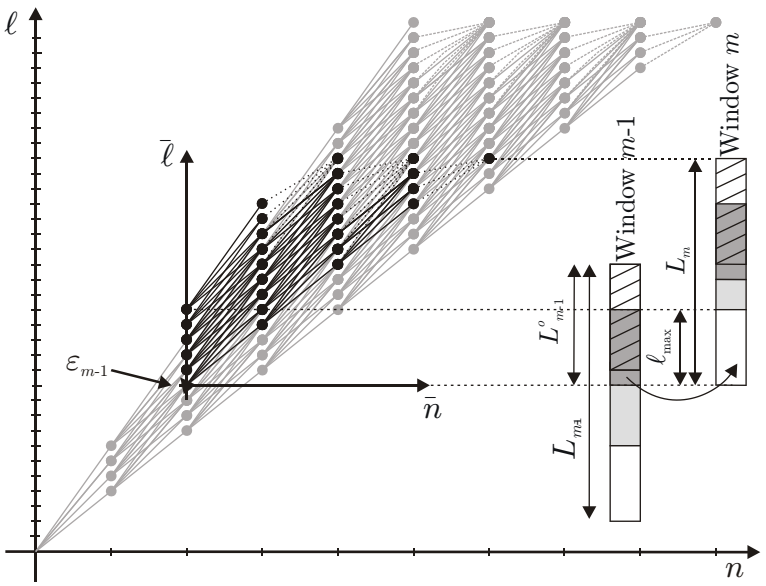

Figure 5.2: ST for the m-th intermediate decoding window, the original trellis is in gray

$m-1$-th window to the m-th window, see Section 5.2.3. This allows a better FS in case of erroneous FS in the $m-1$-th window.

Last Window, $m = M$

Finally, for the last window, the incomplete packets at the end of the window are not to be considered any more: only the presence of a padding packet has to be taken into consideration, see Figure 5.3. For this window, the decoding is performed as in the trellis-based method (Section 4.5), except for the initialization of $\alpha_0^M\left(\bar{\ell}\right)$, which is similar to that of the intermediate windows case.

The m-th and $m+1$-th windows overlap over L_m^o bits, with $\ell_h+\ell_{\max} \leqslant L_m^o < \ell_h+2\ell_{\max}$.

5.2.2 Evaluation of $\gamma_{\bar{n}}$

When $m < M$, transitions corresponding to truncated packets have to be considered at the end of the window. When the size of the truncated packet is larger than ℓ_h, the header is entirely contained in the truncated packet. In this case

$$\gamma_{\bar{n}}\left(\bar{\ell}', L_m\right) \quad = \quad \gamma_{\bar{n}}^{\mathrm{t}}\left(\bar{\ell}', L_m\right),$$

with

$$\gamma_{\bar{n}}^{\mathrm{t}}\left(\bar{\ell}', L_m\right) = p\left(S_{\bar{n}}^m = L_m | S_{\bar{n}-1}^m = \bar{\ell}'\right) \varphi^{\mathrm{t}}\left(\mathbf{y}_{\bar{\ell}'+1}^{L_m}, \mathbf{x}_{\bar{\ell}'+1}^{L_m}\right). \tag{5.1}$$

In (5.1), since truncated packets have to be considered, $p\left(S_{\bar{n}}^m = L_m | S_{\bar{n}-1}^m = \bar{\ell}'\right)$ is given by (4.4). Moreover, the length of a packet, *i.e.*, the content of the length field $\mathbf{u}$, is now only known to be between $\max(L_m - \bar{\ell}', \ell_{\min})$ and $\ell_{\max}$ bits. Thus

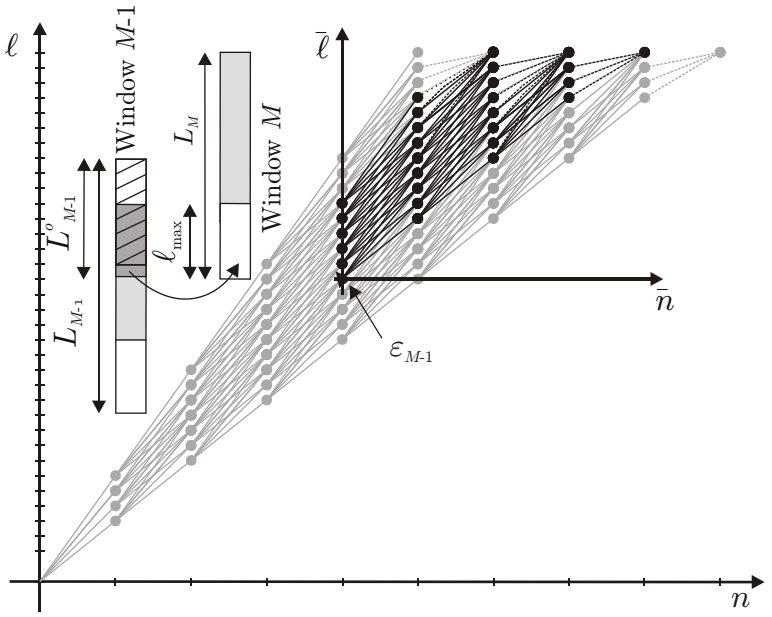

Figure 5.3: ST for the last decoding window, the original trellis is in gray

$$\varphi^{\mathrm{t}}\left(\mathbf{y}_{\bar{\ell}'+1}^{L_m}, \mathbf{x}_{\bar{\ell}'+1}^{L_m}\right) = P\left(\mathbf{y}_k|\mathbf{k}\right) \sum_{\mathbf{p}} P\left(\mathbf{y}_p|\mathbf{p}\right) P\left(\mathbf{p}\right)$$
$$\sum_{\ell=\max(L_m-\bar{\ell}',\ell_{\min})}^{\ell=\ell_{\max}} P\left(\mathbf{u}\left(\ell\right)\right) \sum_{\mathbf{o}} \left(P\left(\mathbf{y}_u|\mathbf{u}\left(\ell\right)\right) P\left(\mathbf{y}_o|\mathbf{o}\right)\right.$$
$$\left. P\left(\mathbf{y}_c|\mathbf{c}=\mathbf{f}\left(\mathbf{k},\mathbf{u}\left(\ell\right),\mathbf{o}\right)\right) P\left(\mathbf{o}\right)\right). \tag{5.2}$$

When the size of the truncated packet is strictly less than ℓ_h, for the sake of simplicity, all bits of the truncated header are assumed equally likely. In such case

$$\gamma_{\bar{n}}\left(\bar{\ell}', L_m\right) \;=\; \gamma_{\bar{n}}^{\mathrm{e}}\left(\bar{\ell}', L_m\right),$$

with

$$\gamma_{\bar{n}}^{\mathrm{e}}\left(\bar{\ell}', L_m\right) \;=\; p\left(S_{\bar{n}}^m = L_m | S_{\bar{n}-1}^m = \bar{\ell}'\right) \sum_{\mathbf{x}_{\bar{\ell}'+1}^{L_m}} P\left(\mathbf{y}_{\bar{\ell}'+1}^{L_m}|\mathbf{x}_{\bar{\ell}'+1}^{L_m}\right) P\left(\mathbf{x}_{\bar{\ell}'+1}^{L_m}\right), \tag{5.3}$$

where $p\left(S_{\bar{n}}^m = L_m | S_{\bar{n}-1}^m = \bar{\ell}'\right)$ is still given by (4.4) and

$$P(\mathbf{x}_{\bar{\ell}'+1}^{L_m}) = 2^{-\ell(\mathbf{x}_{\bar{\ell}'+1}^{L_m})},$$

since all beginning of headers are assumed equally likely.

For $m = M$, the evaluation of $\gamma_{\bar{n}}^M$ is as in the trellis-based method (Section 4.5.2).

5.2.3 Initialization of α in the sliding trellises

In the SW-BCJR algorithm proposed in [30], the α^ms evaluated in the m-th window are deduced from those evaluated in the $m-1$-th window. Here, since the number of states S_n evolves with packet-clock n, $\alpha_{\bar{n}}^m$ cannot be obtained that easily from $\alpha_{\bar{n}}^{m-1}$.

In the $m-1$-th window, one has evaluated $\alpha_{\bar{n}}^{m-1}\left(\bar{\ell}\right)$, with $0 \leqslant \bar{\ell} \leqslant L_{m-1}$ and $0 \leqslant \bar{n} \leqslant \lceil L_{m-1}/\ell_{\min} \rceil$. We choose to propagate at most $\ell_{\max}$ values of α from the packet-clock $\bar{n} = \widehat{N}_{m-1}^c$ in the $m-1$-th window to the packet-clock $\bar{n} = 0$ in the m-th window (for $\bar{\ell} = 0, ..., \ell_{\max} - 1$) as follows

$$\alpha_0^m(\bar{\ell}) = \kappa\, \alpha_{\widehat{N}_{m-1}^c}^{m-1}\left(\varepsilon_{m-1} - \varepsilon_{m-2} + \bar{\ell}\right), \tag{5.4}$$

where κ is some normalization factor chosen such that the $\alpha_0^m(\bar{\ell})$s sum to one. This allows the first packet of the m-th window to start at any bit index between $\varepsilon_{m-1}+1$ and $\varepsilon_{m-1}+\ell_{\max}$.

5.2.4 Complexity Gain

Now we will compare the complexity of the ST-based FS technique to that of the trellis-based FS technique presented in the previous chapter. For the ST-based FS technique, consider M windows of approximately the same size L^w, which are overlapping on average on $L^o = \ell_h + 1.5\ell_{\max}$ bits. For sufficiently large L, one will have to process

$$M \approx \frac{L}{L^w - L^o}$$

overlapping windows with as many STs, each one with

$$\mathcal{N}_{\mathrm{n}}^w \approx \frac{(L^w)^2}{2}\left(\frac{\ell_{\max} - \ell_{\min}}{\ell_{\max}\ell_{\min}}\right)$$

nodes. The total number of nodes to process is then

$$M\mathcal{N}_{\mathrm{n}}^w \approx \frac{L}{L^w - L^o}\frac{(L^w)^2}{2}\left(\frac{\ell_{\max} - \ell_{\min}}{\ell_{\max}\ell_{\min}}\right). \tag{5.5}$$

The total number of nodes is much less than that of the trellis-based FS technique. A comparison for different values of burst size L is provided in the Table 5.1 for window size $L^w = 480 + L^o$, in the Table 5.2 for window size $L^w = 600 + L^o$, and in the Table 5.3 for window size $L^w = 900 + L^o$. For example, when $L = 12960$ bytes and $L^w = 480 + L^o$, the ST-based technique reduces the number of nodes by a factor 10 as compared to the trellis-based technique. Figure 5.4 compares the number of nodes in the trellis-based technique with the ST-based technique for different values of burst size L and window size L^w.

The decoding complexity of the ST-based technique is thus less than that of the trellis-based technique. One can observe that the complexity gain increases with the increase in the burst size L. The graph showing the complexity gain vs. burst size L for different window sizes L^w is shown in Figure 5.5. Choosing small values for L^w reduces the decoding complexity as well as the latency, see Figure 5.5. The price to be paid is some sub-optimality in the decoding performance. Note that L^w cannot be chosen too small (smaller than $\ell_h + 2\ell_{\max}$) to ensure at least one reliable FS in each window.

L (bytes)	1800	4000	8000	12960	16000	24000
Trellis-based (# of Nodes)	24300	120000	480000	1259712	1920000	4320000
ST-based (# of Nodes)	17400	38600	77200	125100	154450	231700
Complexity Gain	1.4	3.1	6.2	10.0	12.5	18.6

Table 5.1: Complexity comparison between the trellis-based and ST-based ($L^w = 480 + L^o$) FS techniques

L (bytes)	1800	4000	8000	12960	16000	24000
Trellis-based (# of Nodes)	24300	120000	480000	1259712	1920000	4320000
ST-based (# of Nodes)	18500	41000	82000	133000	164200	246200
Complexity Gain	1.3	2.9	5.8	9.5	11.7	17.5

Table 5.2: Complexity comparison between the trellis-based and the ST-based ($L^w = 600 + L^o$) FS techniques

L (bytes)	1800	4000	8000	12960	16000	24000
Trellis-based (# of Nodes)	24300	120000	480000	1259712	1920000	4320000
ST-based (# of Nodes)	21800	48500	96900	157000	193900	290900
Complexity Gain	1.1	2.5	4.9	8.0	9.9	14.9

Table 5.3: Complexity comparison between the trellis-based and the ST-based ($L^w = 900 + L^o$) FS techniques

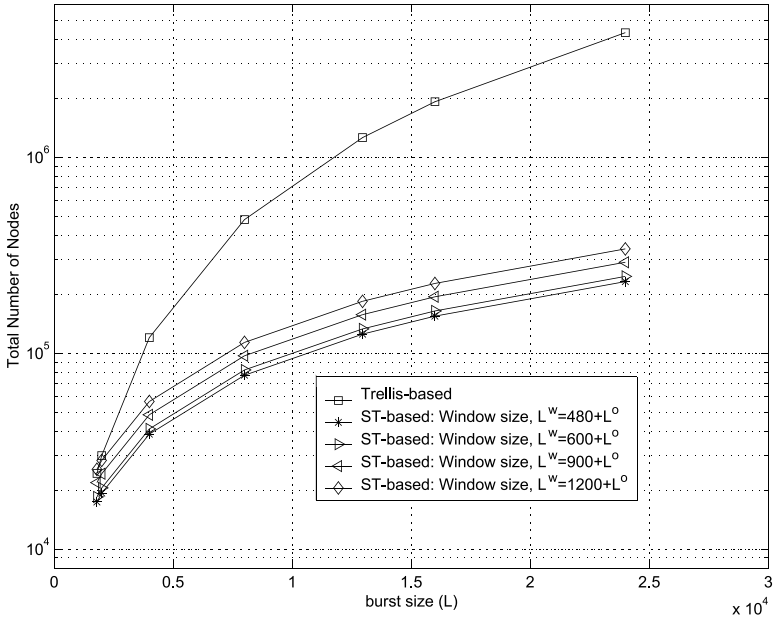

Figure 5.4: Total # of Nodes vs. burst size L for different window sizes L^w

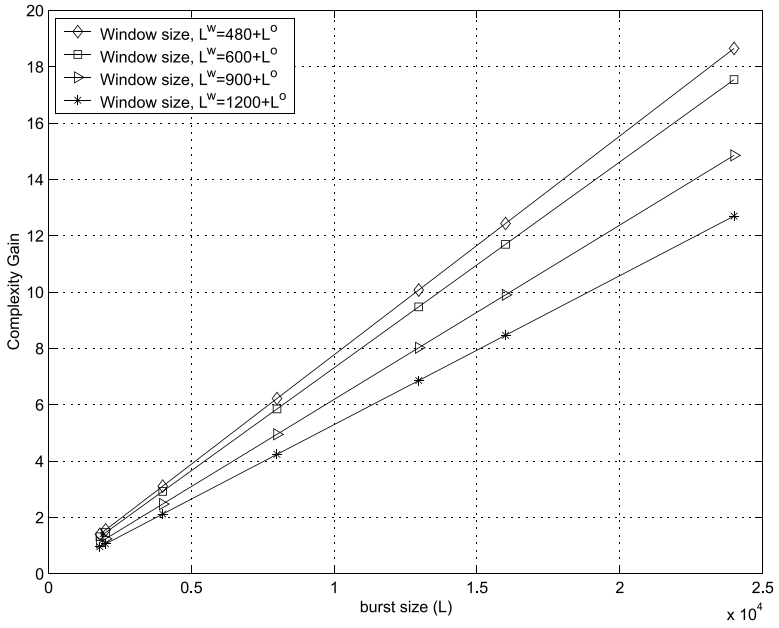

Figure 5.5: Complexity Gain vs. burst size L for different window sizes L^w

5.3 Simulation results

For simulation results, the simulator model and the channel specifications of Section 3.5 are used. Simulation results over AWGN channel comparing the trellis-based FS technique, the low-complexity ST-based approach, and the conventional HD-based FS are shown in Figure 5.6, where EPLR is plotted against SNR. Results for on-the-fly methods such as MU and NP FS methods (explained in Chapter 3) are also shown for comparison.

For the ST-based approach, the burst of $L = 1800$ bytesis divided into three overlapping windows, with $L_1 = 600$ bytes, $L_2 = 600 + L_1^o$ bytes, and $L_3 = 600 + L_2^o$ bytes. Compared to the trellis-based FS (with CI), the ST-based approach shows a slight performance degradation of 0.5 dB, but reduces the delay and the computational complexity. On average, the overlap is about 277 bytes and a decrease in complexity by a factor of 1.7 is observed.

Figure 5.7 performs the same comparison over a Rayleigh channel. Again, a very slight performance degradation for the ST-based approach is observed compared to the trellis-based method. The channel does not affect the complexity gain.

5.4 Conclusions

A low-delay, reduced-complexity, and efficient ST-based FS technique is presented, which performs FS by dividing the received burst into overlapping windows. In WiMAX the proposed ST-based method is deployed to perform FS of the bursts at the MAC layer, for which simulations are also carried out. Compared to the trellis-based FS technique, the FS delay and computational complexity are significantly reduced. The price to be paid is slight performance degradation. A significant gain in performance is obtained for both AWGN channel and Rayleigh fading channel compared to the state-of-the-art FS techniques.

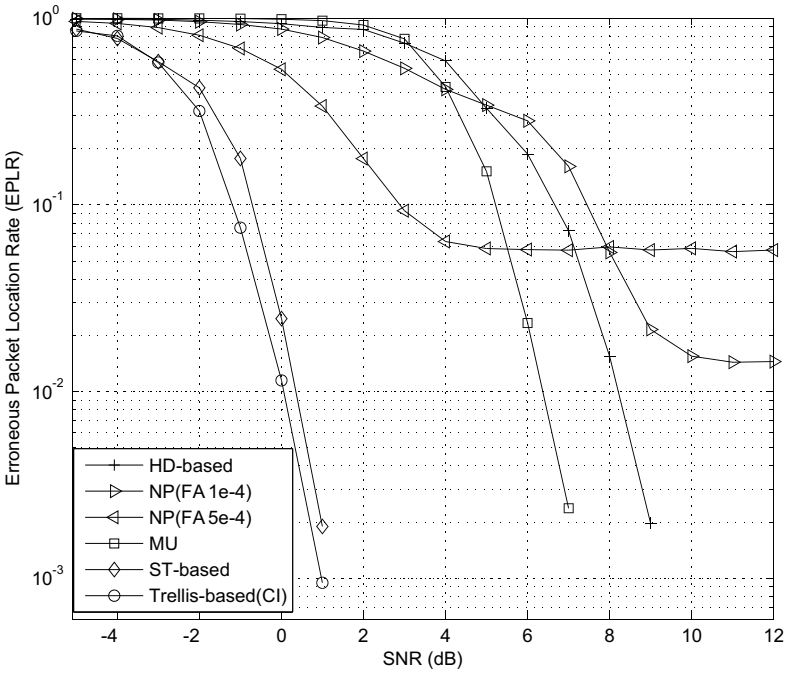

Figure 5.6: Comparison of the proposed ST-based FS technique with HD-based, NP, MU, and trellis-based (with CI) FS methods for bursts transmitted over an AWGN channel

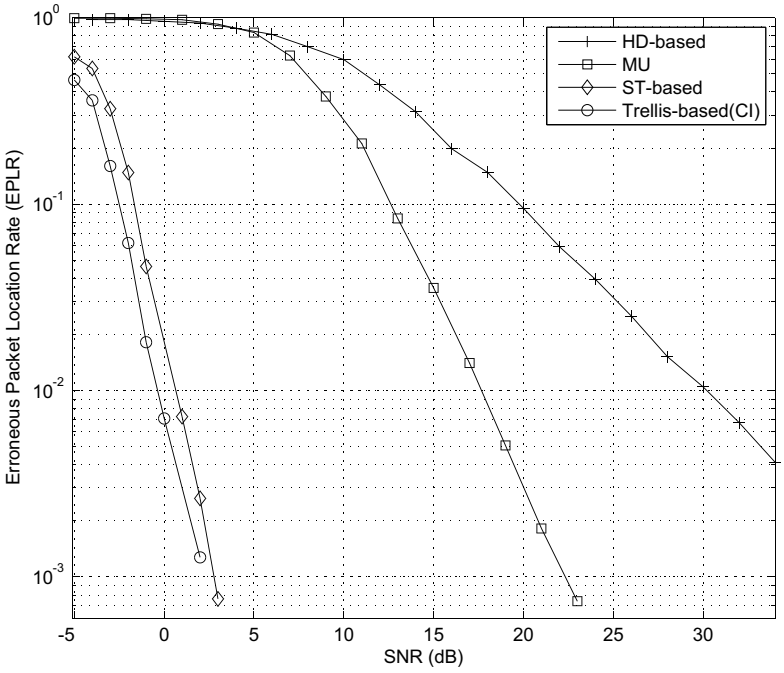

Figure 5.7: Comparison of the proposed ST-based FS technique with HD-based, MU, and trellis-based (with CI) FS methods for bursts transmitted over Rayleigh channel

Extensions to perform on-the-fly FS, by inheriting the useful features from the state-of-the-art FS techniques (presented in Chapter 3) and at the same time exploiting the structural properties of the headers of packets along with the soft information, are presented in the next chapter.

Chapter 6

Robust Three-State FS Automaton

Several end user applications have very tight delay requirements, which can even bear high priority then the quality of the multimedia being delivered. Reverting back to the S-PPSs of WiBOX and WiFi SS, in case the lower layers are using strong error correcting codes, this is the case, *e.g.*, at PHY layer in WiMAX, the soft information can be considered as highly reliable. This enables the use of HEC-based FS techniques to provide low-delay and reduce-complexity FS. The state-of-the-art HEC-based FS techniques can be ameliorated by capitalizing on the availability of the soft information and redundancy present inside the packet header. This chapter is intended to provide improvements to the state-of-the-art FS techniques discussed in Chapter 3.

6.1 Introduction

As underlined in Chapter 3, FS techniques using correlation and ML rules [88; 89; 90] work on-the-fly. Similarly, the variable-length NP [19; 20; 21; 22] and Ueda's [17] FS methods explained in Chapter 3 also perform on-the-fly FS and do not cause any additional latency. But, unfortunately it can be observed that none of these methods make full use of intra and inter-layer redundancies introduced by the structure of the header, redundancies which can be very useful. This usefulness was evidenced in the hold-and-sync trellis-based FS technique (Chapter 4) and its ST-based variant (Chapter 5), which provide high performance FS at the cost of latency. However, the receiver must wait for whole burst in the trellis-based FS technique and for a part of a burst in the ST-based FS technique, which is sometimes undesirable in certain situations (such as framing of IP packets over ATM network), thus an unavoidable trade-off between efficiency and latency is required.

Though, the ST-based technique provides a good trade-off between efficiency and latency, it still remains computationally complex. Furthermore, often the burst or the bit-stream is well synchronized and the channel conditions are favorable as well. In such a situation the synchronizer can only make use of a length field and HEC field, assumed present in the packet header, to make sure that the burst or the bit-stream is well synchronized, thus avoiding unnecessary computations. In case of FS failure due to the channel noise, one can always resynchronize.

This chapter proposes an on-the-fly technique, which makes use of the concept of JPCD and tries to take the best of the above-mentioned methods: It combines robust header recovery techniques inspired from [8] with Bayesian hypothesis testing inspired from NP FS method to localize packet boundaries via a sample-by-sample search. We use a robust 3S

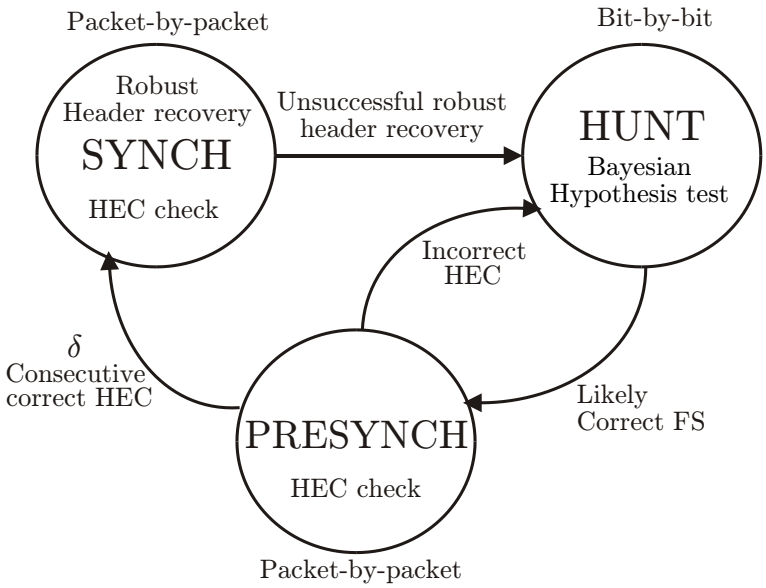

Figure 6.1: Robust 3S FS automaton

automaton, similar to the one adopted by Ueda's method, but instead of hard CRC correction, a soft header recovery technique [8] for correcting the damaged headers (exploiting all known intra and inter-layer redundancies) is exploited to estimate the length field of the header. Moreover, the Bayesian hypothesis testing is performed to search for the correct FS and like trellis-based and ST-based methods it uses the soft information combined with *a priori* information due to the redundancy present at the header of packet

This chapter is organized as follows: The proposed on-the-fly FS technique is presented in Section 6.2, followed by simulation results in Section 6.3 and conclusions in Section 6.4.

6.2 Robust 3S FS Automaton

In this section we propose several improvements to the HEC-based Ueda's method to develop a robust 3S automaton, see Figure 6.1. First, instead of performing error correction in the *SYNC* state, the robust header estimation presented in [8], and briefly recalled in Section 6.2.1, is employed to estimate the length field of a packet. In case of failure to verify HEC of header with estimated length field (HEC verification is performed after replacing the received noisy length field with the estimated length field), the automaton switches to the *HUNT* state, where Bayesian hypothesis testing is performed to search for the correct FS, see Section 6.2.2. The operation performed in the *PRESYNC* state remains unchanged.

Note that alternatively, an automaton with a single *HUNT* State (*HUNT* State Alone, HSA) may be considered to perform FS, without using the packet length field present in the header.

6.2.1 SYNC State: Header Recovery

In [8], a MAP estimator is proposed to determine some fields in the headers of the aggregated packets. In case of FS with packets of variable lengths, one is mainly interested

in the length field denoted by $\mathbf{u}_n$ for the n-th packet. Rewriting the notations of Chapter 4, we have all the fields of the n^{th} packet header in a vector $\mathbf{h}_n = [\mathbf{k}, \mathbf{u}_n, \mathbf{o}_n, \mathbf{c}_n]$ and its corresponding observation at the output of the channel represented by a vector $[\mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c]$. Using the representation of Section 4.5.2.1, one obtains the MAP estimator for the length field [8]

$$\hat{\mathbf{u}}_n = \arg\max_{\mathbf{u}_n} P\left(\mathbf{u}_n \,|\, \mathbf{k}, \mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c\right).$$

Considering equally likely packet sizes, we have

$$\hat{\mathbf{u}}_n = \arg\max_{\mathbf{u}_n} P\left(\mathbf{k}, \mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c \,|\, \mathbf{u}_n\right),$$

which, given that the $\mathbf{k}$ field is well know, becomes

$$\hat{\mathbf{u}}_n = \arg\max_{\mathbf{u}_n} P\left(\mathbf{y}_u|\mathbf{u}_n\right) P\left(\mathbf{y}_o, \mathbf{y}_c \,|\, \mathbf{k}, \mathbf{u}_n\right). \tag{6.1}$$

The term $P\left(\mathbf{y}_o, \mathbf{y}_c \,|\, \mathbf{k}, \mathbf{u}_n\right)$ is evaluated as

$$\begin{aligned}
P\left(\mathbf{y}_o, \mathbf{y}_c \,|\, \mathbf{k}, \mathbf{u}_n\right) &= \sum_{\mathbf{o}_n, \mathbf{c}_n} P\left(\mathbf{y}_o, \mathbf{y}_c, \mathbf{o}_n, \mathbf{c}_n \,|\, \mathbf{k}, \mathbf{u}_n\right) \\
&= \sum_{\mathbf{o}_n, \mathbf{c}_n} P\left(\mathbf{o}_n\right) P\left(\mathbf{y}_o, \mathbf{y}_c, \mathbf{c}_n \,|\, \mathbf{k}, \mathbf{u}_n, \mathbf{o}_n\right) \\
&= \sum_{\mathbf{o}_n, \mathbf{c}_n} P\left(\mathbf{o}_n\right) P\left(\mathbf{y}_o \,|\, \mathbf{o}_n\right) P\left(\mathbf{y}_c, \mathbf{c}_n \,|\, \mathbf{k}, \mathbf{u}_n, \mathbf{o}_n\right) \\
&= \sum_{\mathbf{o}_n, \mathbf{c}_n} P\left(\mathbf{o}_n\right) P\left(\mathbf{y}_o \,|\, \mathbf{o}_n\right) P\left(\mathbf{c}_n \,|\, \mathbf{k}, \mathbf{u}_n, \mathbf{o}_n\right) P\left(\mathbf{y}_c \,|\, \mathbf{k}, \mathbf{u}_n, \mathbf{o}_n\right),
\end{aligned}$$

which, using the same reasoning as in Section 4.5.2.1 that HEC $\mathbf{c}_n$ is fully determined by $\mathbf{k}$, $\mathbf{u}_n$, and $\mathbf{o}_n$, becomes

$$P\left(\mathbf{y}_o, \mathbf{y}_c \,|\, \mathbf{k}, \mathbf{u}_n\right) = \sum_{\mathbf{o}_n} P\left(\mathbf{o}_n\right) P\left(\mathbf{y}_o \,|\, \mathbf{o}_n\right) P\left(\mathbf{y}_c \,|\, \mathbf{c}_n = \mathbf{f}\left(\mathbf{k}, \mathbf{u}_n, \mathbf{o}\right)\right). \tag{6.2}$$

Finally, using (6.2) in (6.1), the MAP estimator becomes

$$\begin{aligned}
\hat{\mathbf{u}}_n = \arg \max_{\mathbf{u}_n \in \Omega_u} \; & P\left(\mathbf{y}_u|\mathbf{u}_n\right) \\
\sum_{\mathbf{o}_n} & P\left(\mathbf{y}_o|\mathbf{o}_n\right) P\left(\mathbf{o}_n\right) P\left(\mathbf{y}_c|\mathbf{c}_n = \mathbf{f}\left(\mathbf{k}, \mathbf{u}_n, \mathbf{o}_n\right)\right),
\end{aligned} \tag{6.3}$$

where $\Omega_u = \{\ell_{\min}, ..., \ell_{\max}\}$ is the set of lengths which may be taken by the length field. $P\left(\mathbf{o}_n\right)$is the a priori probability of $\mathbf{o}_n$, $P\left(\mathbf{y}_u|\mathbf{u}_n\right)$ is the likelihood of the length field from the channel and $\mathbf{f}$ is a generic encoding function to calculate HEC/HCS. The evaluation of (6.3) may be done optimally with a complexity $\mathcal{O}(\ell\left(\mathbf{o}\right) 2^{\ell(\mathbf{c})})$, or sub-optimally with a reduced-complexity algorithm, see [8] for more details.

6.2.2 HUNT State: Bayesian hypothesis test

In NP FS method, discussed in Section 3.4.1, hypothesis tests based on NP criterion are used to determine whether a packet starts at a given bit index. This technique is efficient

when the SW is long, but suffers limitations when it is short.

This section is devoted to the construction of Bayesian hypothesis tests exploiting all sources of redundancy present in the header along with the soft information provided by the channel as evidenced earlier in the trellis-based and the ST-based FS techniques. This allows to build more efficient LRT than the LRT of NP FS method, especially when the SW is short.

Let d_i and h_i be the i^{th} symbol of the modulated data and SW/header grouped in vectors of same size $\mathbf{d}$ and $\mathbf{h}$ (*i.e.*, $\ell(\mathbf{d}) = \ell(\mathbf{h}) = \ell_h$), respectively. Transmission is assumed to be performed over a Rayleigh fading channel. Let y_i be the received sample at i^{th} symbol. After observing ℓ_h subsequent samples, the synchronizer must choose between the following two hypotheses, H_h(header hypothesis) and H_d(data hypothesis), representing whether the ℓ^{th} location is the start of a packet or otherwise, respectively.

$$\begin{aligned} H_d &: y_i = r_i d_i + n_i,\ i = \ell, \ell+1, ..., \ell+\ell_h-1 \\ H_h &: y_i = r_i h_i + n_i,\ i = \ell, \ell+1, ..., \ell+\ell_h-1 \end{aligned}$$

Where, r_i are Rayleigh fading coefficients and n_i are independent, identically distributed Gaussian r.v.s, with zero mean and variance σ^2. Decisions are indicated by D_d and D_h corresponding to the true hypotheses H_d and H_h, respectively.

Let P_{H_j} be the *a priori* probability of hypothesis H_j , where $j \,\epsilon\, \{d,\, h\}$. Our objective is to select the hypothesis with maximum APP $q(H_j)$, *i.e.*, we choose D_h if $q(H_d) \leq q(H_h)$. Where,

$$q(H_j) = P(H_j|\mathbf{y}) = \frac{P(\mathbf{y}|H_j) P_{H_j}(\ell)}{P(\mathbf{y})}$$

Consider the bit index ℓ of a burst. Under the hypothesis H_h that a packet header $\mathbf{h} = [\mathbf{k}, \mathbf{u}, \mathbf{o}, \mathbf{c}]$ starts at ℓ, one may interpret the corresponding channel output starting at ℓ as $\mathbf{y} = [\mathbf{y}_k, \mathbf{y}_u, \mathbf{y}_o, \mathbf{y}_c] = [y_\ell, y_{\ell+1}, ..., y_{\ell+\ell_h-1}]$ and

$$P(\mathbf{y}|H_h) = \sum_{\mathbf{h}} P(\mathbf{y}|\mathbf{h}, H_h)\, P(\mathbf{h}|H_h)\,. \tag{6.4}$$

With the hypotheses of Section 4.5.2, only headers starting with $\mathbf{k}$ have to be considered, thus (6.4) becomes

$$\begin{aligned} P(\mathbf{y}|H_h) = P(\mathbf{y}_k|\mathbf{k}) \sum_{\mathbf{u}\in\Omega_u} & \left(P(\mathbf{y}_u|\mathbf{u})\, P(\mathbf{u})\right. \\ & \left. \sum_{\mathbf{o}} P(\mathbf{y}_o|\mathbf{o})\, P(\mathbf{o})\, P(\mathbf{y}_c|\mathbf{c} = \mathbf{f}(\mathbf{k}, \mathbf{u}, \mathbf{o}))\right). \end{aligned} \tag{6.5}$$

Under the hypothesis H_d that ℓ does not correspond to the beginning of a packet, now $\mathbf{y} = [y_\ell, y_{\ell+1}, ..., y_{\ell+\ell_h-1}]$ is the channel output when data bits $\mathbf{d}$ are transmitted. Assuming balanced data symbols, one gets

$$\begin{aligned} P(\mathbf{y}|H_d) &= \sum_{\mathbf{d}} P(\mathbf{y}|\mathbf{d}, H_d)\, P(\mathbf{d}|H_d) \\ &= \sum_{\mathbf{d}} P(\mathbf{y}|\mathbf{d}, H_d)\, 2^{-\ell(\mathbf{d})}. \end{aligned} \tag{6.6}$$

Bayesian hypothesis test can now be represented as

$$\Lambda(\mathbf{y}) = \frac{P(\mathbf{y}|H_d)}{P(\mathbf{y}|H_h)} \underset{D_d}{\overset{D_h}{\lessgtr}} \frac{P_a(\ell, H_h)}{P_a(\ell, H_d)}, \tag{6.7}$$

where D_h or D_d correspond to deciding H_h or H_d respectively. $P_a(\ell, H_h)$ and $P_a(\ell, H_d)$ are the *a priori* probabilities of the hypotheses at the bit index ℓ .

When $L - \ell < \ell_{\max}$, ℓ may also represent the start of a padding packet. Thus, an additional hypothesis H_p, corresponding to the presence of a padding packet, has to be considered. After observing $L - \ell$ subsequent samples towards the boundary of the burst, the synchronizer must choose between the two hypotheses, *i.e.*, H_p(padding hypothesis) and H_d(data hypothesis), representing whether the ℓ^{th} location is the start of the padding packet or otherwise, respectively.

$$\begin{aligned} H_p &: y_i = r_i 1_i + n_i,\ i = \ell, \ell+1, ..., L-1 \\ H_d &: y_i = r_i d_i + n_i,\ i = \ell, \ell+1, ..., L-1 \end{aligned}$$

The Bayesian hypothesis test for deciding between H_p and H_d is given by

$$\Lambda(\mathbf{y}) = \frac{P(\mathbf{y}|H_d)}{P(\mathbf{y}|H_p)} \underset{D_d}{\overset{D_p}{\lessgtr}} \frac{P_a(\ell, H_p)}{P_a(\ell, H_d)}. \tag{6.8}$$

Under H_p, we have $P(\mathbf{y}|H_p) = P(\mathbf{y}|\mathbf{1})$, where $\mathbf{1}$ is a vector of ones of the same size as of $\mathbf{d}$ (*i.e.*, $\ell(\mathbf{d}) = \ell(\mathbf{1}) = L - \ell$). $P(\mathbf{y}|H_d)$ is given by (6.6).

First Bayesian hypothesis test of (6.7) is applied to choose between header and data till one reaches the bit index $\ell = L - \ell_{\min}$. If data has been decided (*i.e.*, decision is D_d) for this bit index, then the Bayesian hypothesis test of (6.8) is applied for the bit indexes $\ell > L - \ell_{\max}$ starting from the last correct FS bit index, to see whether the data corresponds to the padding packet. Finally, the best bit index $\ell > L - \ell_{\max}$ is selected to signal the start of the padding packet.

6.2.2.1 A priori probabilities

To evaluate the *a priori* probabilities $P_a(\ell, H_h)$ and $P_a(\ell, H_p)$, one knows that

$$\begin{aligned} P(\ell) &= P_a(\ell, H_h) + P_a(\ell, H_p) \\ &= P(\ell) P(H_h|\ell) + P(\ell) P(H_p|\ell), \end{aligned}$$

where $P(\ell)$ is the *a priori* probability that a packet (be it a data and/or a padding packet) starts at a bit index ℓ of a burst of L bits, and

$$P(H_h|\ell) = P(P_n = 0|S_{n-1} = \ell) = \begin{cases} 0, \text{ if } 0 < L - \ell < \ell_{\min} \\ 1, \text{ if } L - \ell \geq \ell_{\max} \\ \sum_{\lambda=\ell_{\min}}^{L-\ell} \pi_\lambda, \text{ else,} \end{cases} \tag{6.9}$$

and $P(H_p|\ell) = P(P_n = 1|S_{n-1} = \ell) = 1 - P(H_h|\ell)$ are the conditional *a priori* probabilities of H_h (header hypothesis) and H_p (padding hypothesis), respectively.

To determine $P(\ell)$ consider again the trellis representation as shown in Figure 4.3 (Chapter 4). One may write

$$P(\ell) = \sum_{1 \leqslant n \leqslant \lceil \ell/\ell_{\min} \rceil} P(S_n = \ell), \tag{6.10}$$

where $P(S_n = \ell)$ is calculated using (4.16). *A priori* probability $P_a(\ell, H_d) = \overline{P}(\ell)$ corresponding to an absence of the start of packet at bit index ℓ is calculated by $P_a(\ell, H_d) = \overline{P}(\ell) = 1 - P(\ell)$.

6.3 Simulation results

Simulation results, using the simulator model and the channel specifications of Section 3.5, are now provided. The proposed on-the-fly 3S FS method is compared with several other FS techniques. First, the on-the-fly methods such as HD-based, MU, and NP FS methods explained in Chapter 3, serve as reference. Second, the hold-and-sync trellis-based FS technique (with CI) presented in Chapter 4 is also considered, merely as a lower bound for the EPLR, since the corresponding complexity and delay are not of the same order of magnitude. Simulations for the proposed 3S FS technique are done with $\delta = 1$, *i.e.*, a single correct HEC evaluation bring the automaton from the *PRESYNC* state back to the *SYNC* state.

Simulation results for transmission over an AWGN channel are provided in Figure 6.2. The proposed 3S FS method performs better than MU method, due to the use of Bayesian hypothesis testing in hunt operation and to the robust header recovery technique. The MU method performs poorly, especially at low SNR, due to the fact that it uses hard HEC detection/correction and the size of HCS (8 bits) is small compared to the size of the header (48 bits), thus more than 10 candidates for two bit error syndrome have to be considered. The proposed 3S FS automaton performs better at low SNR due to the effectiveness of Bayesian hypothesis testing in the *HUNT* state, which can retrieve FS quickly. The difference between 3S FS and MU decreases at high SNR, since even though header recovery performs well, erroneous FS by Bayesian hypothesis testing degrades the performance because even in a good channel condition one can wrongly assume correct FS due to the simulation of header by random data. The trellis-based technique provides the best performance as expected, but at the cost of a delay equivalent to the length of an aggregated packet (burst). The proposed 3S FS technique clearly provides improved FS compared to the state-of-the-art algorithms, thus providing reasonable compromise between performance and latency. The HSA FS method gives some FS error floor at high SNR, as expected, due to unavoidable and persistent false alarms, as the payload data can

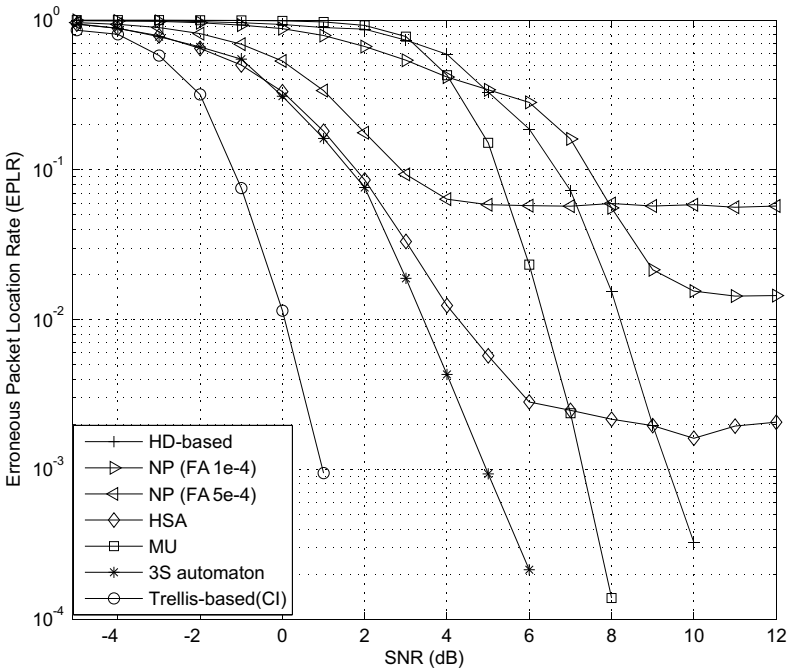

Figure 6.2: Comparison of the proposed 3S automaton and HSA FS methods with HD-based, NP, MU, and trellis-based (with CI) FS methods for bursts transmitted over an AWGN channel

simulate the header. Since, the HSA method uses more redundancy from the header, it gives a much lower floor as compared to the NP FS method.

Simulation results for transmission over a Rayleigh channel are shown in Figure 6.3. MU method performance is highly impacted by Rayleigh channel, while the results for the proposed 3S automaton clearly show its effectiveness in this context.

6.4 Conclusions

HEC detection, as an indicator of correct FS, does not performs efficiently at low SNR for variable-length packets, thus alternative methods are sooth for FS. Instead of HEC-based hunt and confirmation automaton, a robust 3S automaton is proposed in this chapter, which makes use of header recovery and Bayesian hypothesis testing, while exploiting soft information provided at the output of the channel (or channel decoder) in conjunction with *a priori* information due to the redundancy present at the packet header. Simulations clearly demonstrate an improvement in terms of EPLR compared to the state-of-the-art methods performing computations on-the-fly.

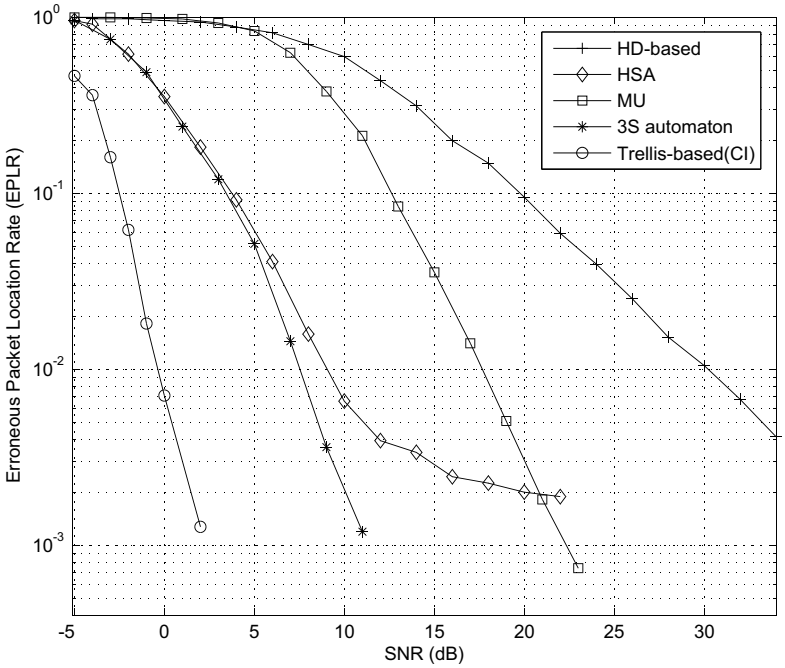

Figure 6.3: Comparison of the proposed 3S automaton and HSA FS methods with HD-based, MU, and trellis-based (with CI) FS methods for bursts transmitted over an Rayleigh channel

PART II: Packet-Level

Forward Error Correction

Decoding

Chapter 7

Introduction to Packet-Level Forward Error Correction

The reliable indoor range of a WiFi access point is only a few tens of feet before bit errors begin to appear. The causes of misbehavior on a WiFi-link thus include, in addition to congestion, low SNR and interference. These later two produce erroneous bits, residual bit errors not corrected by the PHY layer cause CRC/checksum failures at higher (MAC and transport) layers, leading to a significant number of packet drops. Even a single bit error in the packet can lead to the entire packet being discarded. Thus causing a lack of acknowledgment (ACK) and the sender to retransmit the entire packet. This results in a higher PER for larger payload sizes, see *e.g.*, [107] for an investigation on the effect of payload length adaptation and retransmission on throughput and capacity of multimedia users. In both WiMAX and WiFi, retransmission is used to recover packet errors/losses and aims to increase the reliability of data communications. High collision rate and frequent retransmissions in a wireless channel cause unpredictable delays, which degrade the quality of real-time voice and video transmission. Furthermore, often even the error-free reception can trigger retransmission due to the loss of ACKs. This problem is addressed in [108] by differentiating ACK packet losses from data packet losses, thus avoiding redundant retransmissions.

RTP retransmission [109] is an effective packet loss recovery technique for real-time applications with relaxed delay bounds. Sequence Number (SN) field of the RTP protocol can be used to detect lost packets or out-of-order packets as well as to store and retrieve packets for retransmission. The quality of video or audio transmissions can be improved by recovering the lost packets from the retransmission server available on the intranet as described in [110]. Going a step further, for the situations where the latency requirements do not permit retransmission of all lost data, a recovery mechanism using post-processing techniques [111] at the receiver to improve the quality of MPEG-4 video, can be deployed. The receiver-driven selective retransmission extension to RTP called SR-RT [112] can be used to detect packet loss and optionally request retransmission of packets based on the determined priority of the lost packets.

However, when successive retransmissions fail, they add to the traffic congestion, raise the collision probability, degrade the throughput, and increase the end-to-end delay of video streaming [113]. Though retransmission is quite effective to hide link-level loss to users, even with several improvement suggested [114; 115], the inherited delay and overall throughput degradation still remains fatal for the delay-sensitive multimedia applications.

When choosing a repair technique for a particular application, the tolerable latency of the application has to be taken into account. In the case of multimedia conferencing, the end-to-end delay has to be at most a few hundred milliseconds in order to guarantee interactivity, which usually excludes the use of retransmission. FEC at APL layer is a promising alternative for handling losses in multicast/broadcast services and could provide better bandwidth utilization and lesser delay than retransmissions [82]. For example, FEC has to be used undoubtedly for the DVB-H to keep the PER under reliable limit, as retransmissions are omitted. The price to be paid is a reduction of the throughput due to an addition of redundancy.

The idea of the packet-level FEC is that the redundant packets are introduced in each block and at the receiver side the lost packets in a block can be recovered by decoding, if no more than the number of introduced redundant packets have been lost in that block. For example, RFC 5109 [25] protocol is a basic technique to add FEC packets to allow recovery of the lost RTP packets.

This chapter does not provide any new results but rather intends to provide introduction to the packet-level FEC in order to fully understand the proposed robust packet-level FEC decoder presented in the next chapter.

7.1 Introduction

Several studies have shown the efficiency of FEC via simulations and have proved that such a scheme is promising for error correction in wireless broadcasting/multicasting. A two-stage FEC scheme, for multimedia data transmission over WLANs, is presented in [82], where packet-level FEC (stage-one) is added across the APL layer packets to correct packet losses due to congestion and route disruption and bit-level FEC (stage-two) is then added to both the APL layer packets and the stage-one FEC packets to recover bit errors from the MAC/PHY layers. Similarly in [116], FEC is not only used to provide protection against bit errors introduced by the wireless channel, but is also used provide protection against packet loss. Unequal Error Protection (UEP), a scheme that uses *a priori* knowledge of the media to differentially protect data using FEC, is proposed in [117; 118], for the media data transmitted over a DVB-H channel.

Several powerful erasure correcting codes have been proposed for packet-level FEC. The corrupted packets are treated as erasure and classical codes, such as RS codes are used to recover lost packets. For example, for an efficient IP packet retrieval at the DVB-H link layer, several RS based FEC decoding schemes are presented in [119]. Use of modern and capacity-achieving codes, such as fountain codes and Low-Density Parity Check (LDPC) codes, is addressed in [120; 121].

Unfortunately, if a dominant part of the competing flows adds redundancy, the extra overhead may cause increased packet loss ratios in the network, and in turn do more harm than good. Therefore, in [122] it has been demonstrated that by assigning a lower precedence level to FEC packets, the negative effect of the added overhead can be strongly pacified. An experimental study of the packet loss behavior of the WiFi network is presented in [24], which also gives guidelines on how to efficiently use FEC for wireless video multicast in order to improve the overall system performance.

To effectively utilize an FEC scheme, the redundancy level has to be appropriately determined so that the redundant packets do not waste network bandwidth. It should be adapted to the fluctuations of underlying networks. By increasing the amount of redundancy, the media specific schemes can recover from multiple losses. However, increasing

the amount of redundancy when the network loss rate is low will waste bandwidth. This motivates the need to develop methods to control the amount of redundancy depending on the network loss rate. A redundancy control algorithm presented in [123], called the "Bolot algorithm", tries to maintain the loss rate after reconstruction at the receiver between pre-specified LOW and HIGH loss rate limits. The control algorithm will add redundancy if the network loss rate is above the HIGH mark and decrease the amount of redundancy if the network loss rate is below the LOW mark. It is further adapted in [124], in order not to react in case of burst losses, by considering the history of packet losses in the network before changing the amount of redundancy. In [125; 126; 127; 128] the Adaptive FEC (AFEC) protocols for dynamic networks are proposed, where the degree of the redundancy injected into the network is adjusted as a function of network characteristics (network latency, packet losses, etc.). An AFEC Code Control (AFECCC) presented in [129] dynamically tunes the amount of FEC redundancy based on the arrival of ACK packets without using any specific information such as SNR or BER from receivers. While, an AFEC scheme presented in [130] optimizes extra bandwidth usage for the redundancy level.

Wireless channels are varying all the time, therefore AFEC is preferred. In this thesis, however, we plan to study a fixed FEC scheme (*i.e.,* with constant FEC rate), for the given channel condition or SNR, which can easily be adapted to AFEC scheme.

This chapter is organized as follows: A conventional model for the packet-level FEC scheme is presented in Section 7.2. A standard and simple packet-level FEC at RTP layer, *i.e.*, RFC 5109, is detailed in Section 7.3, which is followed by a theoretical analysis of the retransmission and packet-level FEC schemes in Section 7.4 and experimental analysis in Section 7.5. Finally, the chapter finishes with conclusions drawn in Section 7.6.

7.2 System Model

We will present in this section a system model for the packet-level FEC scheme considered in the rest of this thesis. This model would be deployed at the RTP-FEC layer of WiBOX, as discussed in the Chapter 2. Consider a (n, k) packet-level FEC code containing k Media Packets (MedPs) and $m = n - k$ FEC Packets (FECPs). Altogether, they form a group of n packets, such that any k of the n packets can be used to reconstruct the k MedPs. In systematic (n, k) codes the first k of the n encoded packets are identical to the k MedPs. However, such a FEC scheme introduces overhead since extra FECPs are now transmitted by the source station. The overhead introduced is the number of FECPs m to be sent for k MedPs. The number of FECPs m can be determined as follows [24]

$$m = \frac{kP^e}{(1 - P^e)},$$

where, P^e is PER. Note that the level of overhead depends on the PER P^e in the network. Thus, higher the PER, more are the FECPs to be transmitted by the server, thus increasing the overhead and reducing the FEC rate r_{FEC}, which is the ratio of the number of MedPs to the total number of packets, *i.e.*,

$$r_{FEC} = \frac{k}{k + m} = 1 - P^e.$$

From the above discussion, we conclude that it is important to have an accurate estimation of the PER, so that just enough FECPs can be added. PER depends on the

channel conditions, thus for a well specified wireless channel condition, the PER can be calculated, and subsequently the FEC rate r_{FEC} can be decided.

7.3 A Simple RTP-Level FEC Scheme

RTP protocol runs on the top of UDP and assigns increasing SN to each sent packet. Once the receiver has the SNs of the received packets, it can find which packet has been lost. But, if no retransmission mechanism is used, how can this lost packet be recovered? To answer, a generic FEC mechanism is proposed in RFC 5109 [25], where RTP encapsulated FECPs are introduced to recover the lost RTP MedPs from the received error-free packets (MedPs & FECPs).

This scheme describes a method to recover the lost MedP from the error-free hard packets (MedPs & FECPs). It is completely compatible with FEC-incapable SSs, so the SSs in a multicast/broadcast group that do not implement FEC can still work by simply ignoring the protection data. Furthermore, it is independent of the nature of the media being protected, be it audio, video, or otherwise. In particular, the protocol support required by the traditional error correcting codes, such as RS and Hamming codes, can also be provided by RFC 5109. Though, RS code can be utilized, since it is one of the well known block code with good error correction properties and is widely used, but for the sake of simplicity, we will consider a very simplified scenario of using exclusive OR (XOR) operation as an FEC encoder.

Let us consider a simple scenario, where two MedPs, let say MedP1 and MedP2, are encoded (*e.g.*, using XOR operation) to give one FECP, *i.e.*, the FEC rate r_{FEC} is 2/3. The MedPs and their associated FECP are grouped into a MedPs-FEC block, which is transmitted over the network. Now, if the MedP1 is lost it can be recovered from the error-free MedP2 and FECP. Though, RFC 5109 supports an UEP, a scheme that can provide more protection to the critical part of a packet, in this thesis we consider Equal Error Protection (EEP) for simplicity.

According to RFC 5109, an FECP consists of three types of headers, namely, RTP header, FEC header, and FEC-level header. These three headers are concatenated with several payloads to construct an FECP, see Figure 7.1. As we are using a EEP scheme, where one FECP is protecting two MedPs, we will only have a single FEC-level header and a single payload.

7.3.1 FECP Construction

The FECP generation process at RTP layer, as defined in the RFC 5109, is shown in Figure 7.2. We will now explain the construction process of the three headers and payload of an FECP.

7.3.1.1 RTP header fields of FECP.

All the fields in the RTP header of an FECP are used according to RFC 1889 [70], and are explained in detail in appendix A.5. The contents of some of the fields are provided below.

- *Version* (V): 2 bits

This field identifies the version of RTP. Nowadays, the version used by the RTP protocol is two (2).

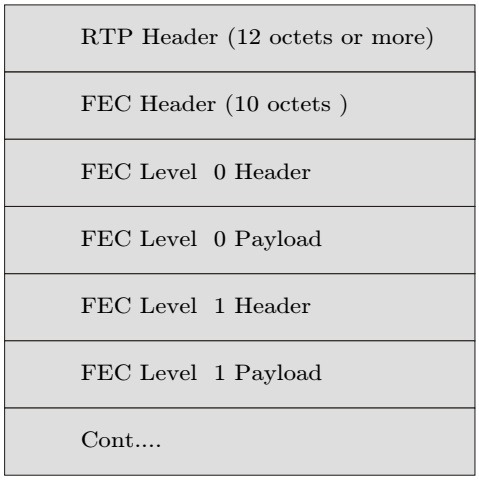

Figure 7.1: FECP Structure

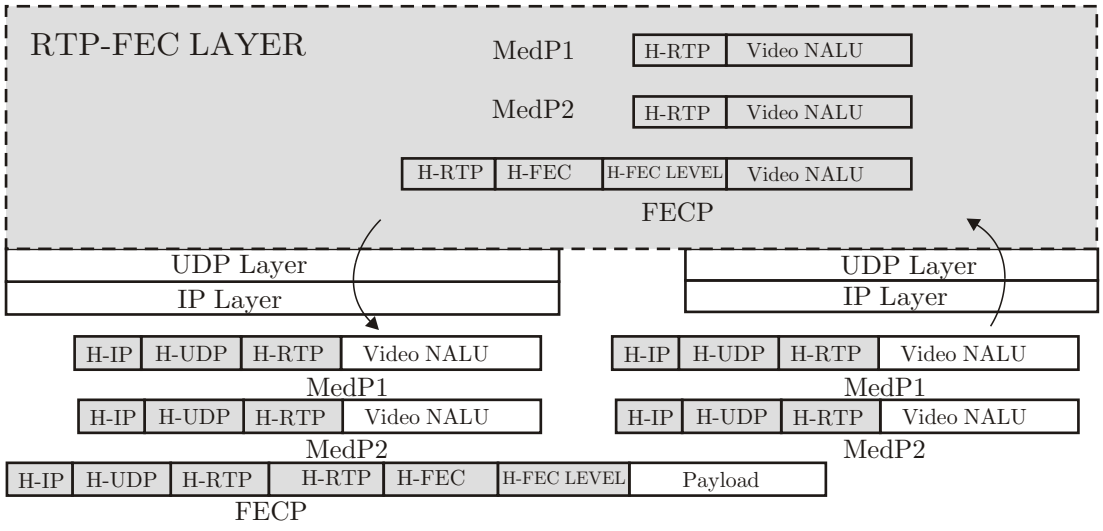

Figure 7.2: Construction process of an FECP

- *Padding* (P): 1 bit

If the padding bit is set, an FECP contains one or more additional padding bytes at the end, which are not part of the payload.

- *Extension* (X): 1 bit

If the extension bit is set, the fixed header is followed by exactly one header extension.

- *CSRC count* (CC): 4 bits

The CSRC count contains the number of contributing source identifiers (*i.e.*, CSRC) that follow the fixed header.

- *Marker* (M): 1 bit

This field is not used for FECP, and is set to 0.

- *Payload type (PT)*: 7 bits

This field identifies the format of the RTP payload and determines its interpretation by the application. It would be set to 127 to identify an FECP. The FEC mechanisms can then be used in a multicast group with mixed FEC-capable and FEC-incapable receivers. In such cases, an FECP will have a payload type that is not recognized by the FEC-incapable receivers, and will thus be disregarded.

- *Sequence Number (SN):* 16 bits

The SN has the standard definition, *i.e.*, it must be one higher than the SN in the previously transmitted FECP.

- *Timestamp (TS)*: 32 bits

The TS must be set to the value of the media RTP clock at the instant the FECP is transmitted. It is set to the TS of the last MedP, which is being protected by this FECP.

- *SSRC*: 32 bits

The SSRC value shall be the same as the SSRC value of the media stream it protects.

Observations: Following observations can be made based on the generation process of the RTP header of an FECP.

- The P, X, CC, and SN fields do not depend on the headers of MedPs being protected.
- The V(= 2), M(= 0), and PT(= 127) fields remain constant in the RTP header of an FECP.
- The TS and SSRC fields are obtained directly from MedP1 and MedP2, *i.e.*, from the TS field of MedP2 and SSRC field of the MedP1 or MedP2, respectively.

7.3.1.2 FEC header fields of FECP.

The FEC header is 10 bytes. The format of the header is shown in Figure 7.3 and consists of

- *E* field:

The E bit is the extension flag reserved to indicate any future extension to this specification. It shall be set to 0, and should be ignored by the receiver.

- *L* field:

The L bit indicates whether the long mask is used. When the L bit is not set, the mask is 16 bits long. When the L bit is set, the mask is then 48 bits long.

- *P, X, CC, M, PT* and *TS* Recovery fields:

The P recovery field, the X recovery field, the CC recovery field, the M recovery field, the PT recovery field, and the TS recovery field are obtained via the protection operation applied to the corresponding P, X, CC, M, PT, and TS values from the RTP header of

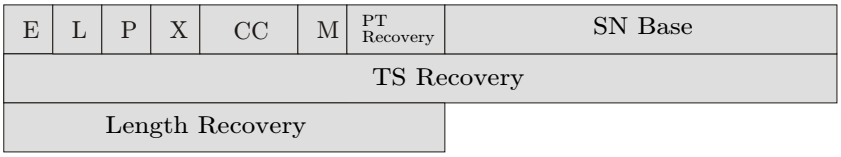

Figure 7.3: The FEC Header of an FECP

the MedPs associated with this FECP. Thus, these fields are generated by a bit-wise XOR operation on the corresponding fields of the RTP header of the MedPs.

- *SN base* field:

The SN base field must be set to the lowest SN. This allows for the FEC operation to extend over any string of at most 16 packets when the L field is set to 0, or 48 packets when the L field is set to 1.

- *Length recovery* field:

The length recovery field is used to determine the length of any recovered MedP. It is computed via the protection operation applied to the unsigned 16-bit representation of the sums of the lengths (in bytes) of the payload, CSRC list, extension, and padding of each of the MedPs associated with this FECP. Length recovery field is only required in the situations when a MedP is completely lost, it may not be required when it is erroneously received.

To generate FEC header, the MedP bit-strings (80 bits in length) is first generated for each MedP to be protected. It is formed by concatenating the following fields together in the order specified:

1. The first 64 bits of the RTP header (64 bits).
2. Unsigned 16-bit representation of the MedP length in bytes minus 12 (for the fixed RTP header), *i.e.*, the sum of the lengths of the following, if present: the CSRC list, extension header, RTP payload, and RTP padding (16 bits).

Now, the FEC bit-string is formed by applying an XOR operation on the MedP bit-strings. Finally, the FEC header is generated from the FEC bit-string as follows:

1. The first (most significant) 2 bits in the FEC bit-string are skipped.
2. E field of FEC header is set to 0.
3. L field of FEC header is set to 0 when 16 bit mask is used and to 1 if 48 bit mask is used.
4. The next bit in the FEC bit-string is written into the P recovery bit of the FEC header.
5. The next bit in the FEC bit-string is written into the X recovery bit of the FEC header.
6. The next 4 bits of the FEC bit-string are written into the CC recovery field of the FEC header.
7. The next bit is written into the M recovery bit of the FEC header.
8. The next 7 bits of the FEC bit-string are written into the PT recovery field in the FEC header.
9. The next 16 bits are skipped.

Protection Length	Mask
Mask cont.(present only when L=1)	

Figure 7.4: The FEC-level header of an FECP

10. The next 32 bits of the FEC bit-string are written into the TS recovery field in the FEC header.
11. The next 16 bits are written into the length recovery field in the FEC header.
12. SN base is set to lowest of the SN of MedPs used for generation of FECP.

Using the above rules all the fields of FEC Header are decided.

Observations: Following observations can be made based on the generation of FEC header.

- The E(= 0) and L(= 0) bits remain constant in the FEC header of an FECP.
- The P, X, CC, M, PT, and TS fields are obtained using an XOR operation (which in the considered scenario is a bit-wise XOR of P, X, CC, M, PT, and TS fields of the MedPs).
- The SN base field must be set to the lowest SN of the MedPs being protected by this FECP. Thus, this field is achieved directly from the SN field of the MedP1.
- The LEN field does not depend on the headers of the MedPs.

7.3.1.3 FEC-level header fields of FECP.

The FEC-level header is 4 or 8 bytes (depending on the L bit in the FEC header). The format of the header is shown in Figure 7.4. The FEC-level header consists of following two fields.

1. The protection length field is 16 bits long and is only useful for UEP scheme.
2. The mask field of the FEC-level header indicates which MedPs are associated with this FECP. It is 16 bits long (when the L bit is not set) or 48 bits long (when the L bit is set). If bit i in the mask is set to 1, then the MedP with $SN = SN_b + i$ is associated with this FECP, where SN_b is the SN Base field in the FEC header of an FECP. The most significant bit of the mask corresponds to $i = 0$, and the least significant to $i = 15$ when the L bit is set to 0, or $i = 47$ when the L bit is set to 1.

Observations: Following observations can be made based on the generation of FEC-level header.

- The protection length field does not depend on the headers of the MedPs.
- The mask field is obtained using the SN fields of MedP1 and MedP2.

7.3.1.4 Payload of FECP.

Finally, for the generation of the payload of an FECP, the MedP bit-strings are simply the RTP MedPs. The FEC bit-string is thus the bit-wise XOR of these MedP bit-strings. If the lengths of the protected MedPs are not equal, each shorter MedP must be padded to the length of the longest MedP by adding bytes 0x00 at the end. The FEC Payload

starts with the 13th byte in the FEC bit-string. The reason for omitting the first 12 byte (RTP header excluding CSRC field) is that this information is already protected by the FEC header.

Observations: To summarize the observations,

- Payload including the CSRC field of the RTP header is obtained by an XOR operation.

7.4 Theoretical Analysis

In this section we present a theoretical analysis of the retransmission and FEC schemes. It will provide a deeper insight into the preferable scheme for a given channel condition.

7.4.1 Retransmission Scenario

To calculate PER P^e, let us denote the probability that a packet is received correctly as the probability of success $P^s = 1 - P^e$. The success probability under a random and independent bit error is given by

$$P^s = (1 - BER)^L,$$

where, BER is the bit error rate of the channel and L is the length of a packet.

Let F be a binomial distributed random variable, which denotes the number of packets that were erroneous from the previous transmission. Using binomial distribution properties, we can compute the probability that F packets failed transmission out of N packets that were in the queue at the beginning of the transmission as follows

$$p(F|N, P^e) = \left(\begin{array}{c} N \\ F \end{array} \right) (P^e)^F (1 - P^e)^{N-F}.$$

The expected number of erroneous packets from the previous transmission is

$$E(F|N, P_e) = NP^e.$$

Now, consider the retransmission scenario, with r retransmissions. Let P_r^e be the remaining PER after r retransmissions. It can be calculated using

$$P_r^e = 1 - \left[(1 - P^e) (1 + P^e + ... + (P^e)^r) \right],$$

where by assuming that the bit errors are independent of each other, one gets

$$P^e = 1 - (1 - BER)^L.$$

It should be kept in mind that for the retransmission scheme, ultimately the remaining PER after infinite retransmissions is zero, *i.e.*, $P_\infty^e \approx 0$. The price to be paid is a large delay that is unacceptable for the real-time multimedia applications.

Thus, the goodput for the retransmission scheme is given by

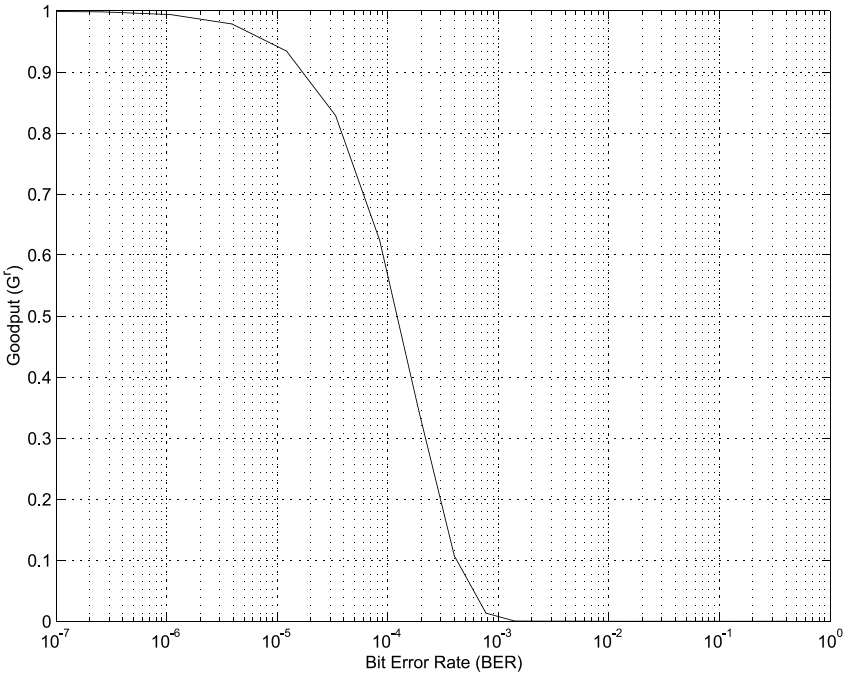

Figure 7.5: Goodput vs. BER (retransmission scheme)

$$\begin{aligned} G^r &= \frac{N\left(1-P^e\right)+NP^e\left(1-P^e\right)...}{N+N\left(P^e\right)+N\left(P^e\right)^2...} \\ &= 1-P^e. \end{aligned}$$

Note that goodput is independent of the number of the retransmissions allowed. The goodput is related to the BER by

$$\begin{aligned} G^r &= 1-1-(1-BER)^L \\ &= (1-BER)^L. \end{aligned}$$

Let us assume that one is using BPSK modulation over AWGN channel with variance σ^2, the goodput is related to SNR (dB), *i.e.* $\frac{1}{2\sigma^2}$, by

$$\begin{aligned} G^r &= 1-1-(1-BER)^L \\ &= (1-\frac{1}{2}erfc(\sqrt{\frac{1}{2\sigma^2}}))^L. \end{aligned}$$

For transmission over AWGN channel, the theoretical goodput curve is plotted against the BER in Figure 7.5. The goodput curve is drawn for packet length L of 700 bytes. One can observe that with an increase in the BER, the goodput of the retransmission scheme decreases, this is due to the fact that one needs more retransmissions at high BER. Similar observation can be made for the theoretical goodput curve plotted against PER in Figure 7.6. If the channel worsens, the goodput for the retransmission scheme would decrease due to an increase in the number of retransmissions (or due to an increase in PER).

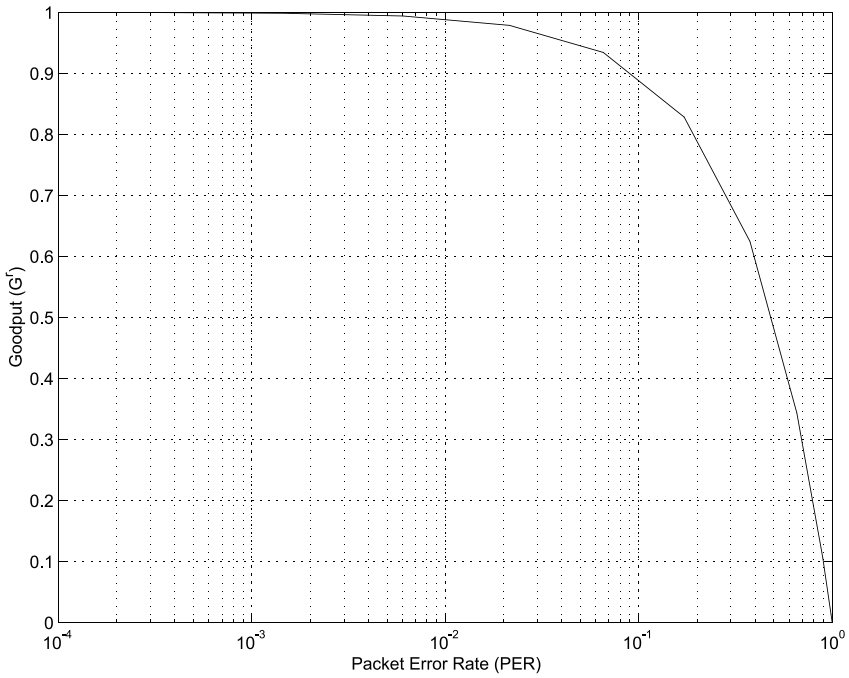

Figure 7.6: Goodput vs. PER (retransmission scheme)

7.4.2 Packet-level FEC Scheme

We have considered FEC encoding with a FEC rate $r_{FEC} = 2/3$, *i.e.*, introducing a single FECP for every two MedPs. At this rate, the simple FEC scheme presented above can only recover the lost packet if the goodput is at least

$$G^f_{min} \quad = \quad 2/3 = 66.7\%,$$

or in other words, if at least two error-free packets have reached the RTP layer. Thus, in such situation, *e.g.*, if the MedP1 is lost, it can be recovered from the MedP2 and FECP. The FEC decoding using RFC 5109 cannot recover the lost packet if the goodput falls below the above-mentioned lower limit. This problem is addressed in the next chapter, where even the erroneous packets are used during the recovery process.

7.5 Simulation results

Simulations are carried over Rayleigh fading channel, where the modulated signal is subject to zero mean and unit variance fast (bit) Rayleigh fading plus zero-mean AWGN noise. For performance analysis, the PER P^e is drawn as a function of the channel SNR. Simulations are carried out for a video stream, with the MTU (the size of the largest IP packet) of 1500 bytes.

The block diagram of the simulation model is shown in Figure 7.7. At the transmitter, the RTP-level FEC encoding (using an XOR operation as presented in Section 7.3) is applied to generate FECPs (at the RTP-FEC layer), which are packetized into UDP/IP packets. Packets in a MedPs-FEC block are modulated with BPSK before transmission

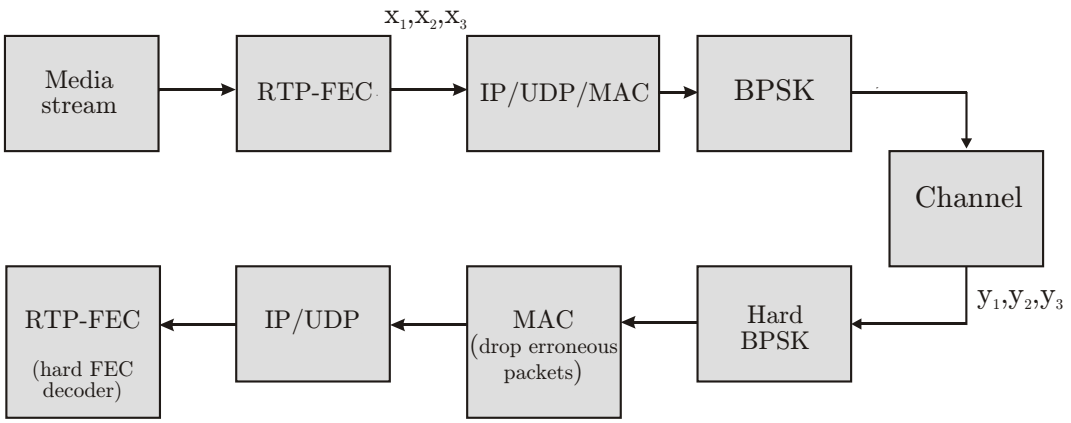

Figure 7.7: Simulation Block Diagram: The SPS is used at the receiver, the hard BPSK demodulation technique is used at the PHY layer, erroneous packets are dropped at the MAC layer, and error-free packets are used by the hard FEC decoder at the RTP-FEC layer.

over the channel. At the receiver, the SPS is deployed, where the hard BPSK demodulation technique is used at the PHY layer and the erroneous packets are dropped at the MAC layer after CRC verification.

Simulation results for the retransmission scheme and the hard RTP-level FEC scheme over Rayleigh channel are provided in Figure 7.8. For retransmission scheme, two curves are shown for two different allowed number of retransmissions . In the hard RTP-level FEC decoder, erroneous packets are dropped at the MAC layer and error-free packets are used to reconstruct the dropped packet, if at least two packets from a MedPs-FEC block have managed to reach the RTP-FEC layer for FEC decoding.

Though, the discussed RTP-level FEC scheme can be used to decrease the PER, it cannot guarantee reliable delivery of multimedia packets. That is why FEC and error-resilient features of video decoders are typically used to complement each other. Let us assume that the given video decoder can tolerate PER P^e of about 10%. While, the channel worsens, the system should be able to keep $P^e < 10\%$ when using robust video decoder in conjunction with the retransmission or FEC scheme. Results show that the SNR of greater than 30 dB is required to keep errors under the tolerable limit in case of the retransmission scheme (with 2 retransmissions allowed) and of greater than 35 dB is required in case of the RTP-level FEC scheme.

7.6 Conclusions

A simple investigative study of the two well known packet loss recovery techniques, *i.e.*, retransmission and FEC schemes, is provided. Owing to the importance of developing a standard-compliant FEC decoder, the RFC 5109 protocol is used to provide FEC protection at RTP layer. In the SPS, a simple RTP-level FEC scheme can help to reconstruct the dropped packet from the error-free packets, but recovery becomes impossible when the dropped packets are more than the number of redundant packets. Thus, it can be concluded that the use of SPS is not very effective for the FEC scheme, and the retransmission scheme

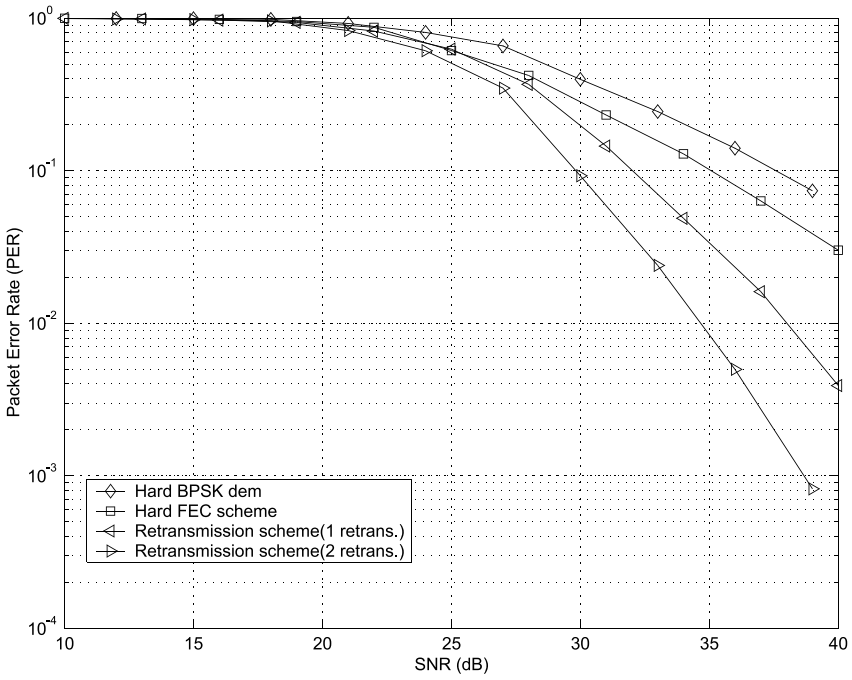

Figure 7.8: Simulation results (PER vs. SNR): Video stream (MTU of 1500 bytes) transmission over Rayleigh channel

at the cost of latency provides better performance than the simple RTP-level FEC scheme considered.

Furthermore, it has been found that there are several redundancies present in the RTP header, which could be exploited along with the FEC redundant packet, using the idea of JPCD, to improve the performance of the RTP-level FEC decoder. Any such FEC decoder should be capable enough to work on the partially erroneous s-packets provided by the lower layers. Therefore, keeping in view all of these requirements, in the next chapter, a robust RTP-level FEC decoder for the S-PPS is presented.

Chapter 8

Robust MAP Decoding for RTP-Level FEC

Often in video streaming, if the underlying source decoder at APL layer is error-resilient, then the aim of an FEC scheme is to keep the number of losses under a tolerable threshold instead of attempting 100% data recovery. For example, various error resiliency schemes employed by H.264/AVC [50] can be used to tolerate PER of up 20% [51; 52]. FEC over the WiFi-link, of the inter-networking scenario, cannot meet the required target PER with sufficient throughput at wireless receiver using SPS, which may drop erroneous packets at lower protocol layers. To facilitate FEC-based recovery, packets should not be dropped so that maximum numbers of packets are relayed to a wireless receiver's FEC decoder.

Packet losses in a wireless channel can be roughly categorized into two:

1. Packets dropped due to collisions, and
2. Packets discarded in the MAC/PHY layers due to internal bit errors.

The former can rise from the collisions due to multiple access of the channel by several users. The WiBOX, in the inter-networking scenario under investigation, uses the multiple channel access scheme, *i.e.*, HCCA, and thus can prevent packet collisions during the broadcast transmissions. The later is avoided by using S-PPS, capable of forwarding erroneous packets to the higher layers. The S-PPS complemented with a robust packet-level FEC decoder to mitigate the errors left by the MAC/PHY layers can boost the flow of s-packets (error-free and erroneous) to the APL layer.

8.1 Introduction

The objective of the permeable-layer receiver should be to forward erroneous payloads to the packet-level FEC decoder for recovery just before the source decoder at APL layer. For example, in [1] a permeable-layer receiver along with a FEC scheme (at APL layer), capable of exploiting traditionally useless erroneous information, is proposed. In this FEC scheme, all segments (even erroneous) of the IP packet are forwarded to the APL layer, where the FEC decoder recovers the IP packet, but erroneous segments still remain useless. Similarly, in [11; 9], the authors have proposed modifications to the receiver and have identified that only few important fields present in the UDP header are enough to uniquely identify each active multimedia session with few false alarms, thus even packets with erroneous headers can be forwarded to the higher layer. RS is then used, as a FEC code,

to decode errors (in corrupted packets) and erasures (from missed detections) in the UDP payload, simultaneously. Nevertheless, false alarms can desynchronize the video and/or FEC decoder, thus it demands reliable SNs to provide erasure locations.

To the best of the author's knowledge there is no packet-level FEC decoding scheme that make use of the intra and inter-packet redundancies introduced by the structure of the header, the benefit of which has already been proved in the FS approaches, presented earlier in this thesis. We argue that using the idea of JPCD, one can jointly utilize

- the soft information from the channel,
- several redundancies present inside the packet header, and
- the redundant FECP

to improve the header recovery performance of the FEC decoder. Furthermore, an unhindered flow of soft information from the PHY layer to the APL layer, through the FEC decoder, is critical for several JSCD techniques in order to increase efficiency of the video decoder. Though, the permeable protocol layer proposal [8], in the absence of packet-level FEC, can enable exchange of soft and partially erroneous information between protocol layers, there is no available packet-level soft-output FEC decoder that can allow simultaneous flow of soft and partially erroneous information to the APL layer.

This chapter presents an RTP-level MAP decoder for the RTP-level FEC scheme presented in the previous chapter. As already discussed, the RTP-level FEC scheme uses the RFC 5109 to generate FECPs (see previous chapter). We stick to the use of an XOR operation because it is the simplest FEC encoding function available and it helps to focus more on the development of the JPCD technique at the receiver. We present a JPC decoder to be deployed at the RTP-FEC layer of S-PPS, which can even utilize the partially erroneous s-packets to decrease the PER over the WiFi-link.

This chapter is organized as follows: First, the MAP estimation for the packet-level FEC decoding is proposed in Section 8.2. The structures of the RTP-level MedP and FECP are described in Section 8.3, followed by the presentation of the RTP-level MAP decoding in Section 8.4. Simulation results are presented in Section 8.5, while the conclusions are drawn in Section 8.7.

8.2 MAP Estimation for Packet-level FEC

Let us consider the same simple packet-level FEC scheme as discussed in the previous chapter, where two MedPs are encoded to give one FECP using RFC 5109 and the XOR operation on the hard error-free packets can be used to reconstruct the lost packet. However, in several situations, it is not necessary to receive error-free packets, especially in the case when the lower layers are permeable, *i.e.*, capable of forwarding erroneous packets to the higher layers (required for example for a robust decoding of the video packet), instead of dropping them. This is exactly the case in the considered S-PPS detailed in Chapter 2. Due to use of the multiple access technique (*i.e.*, HCCA) provided in 802.11e [37], one can fairly assume that no packet is completely lost in the channel, but in fact it can be erroneous. Thus, we always have all the packets (MedPs & FECP) of a MedPs-FEC block reaching the RTP-FEC layer. Furthermore, the S-PPS requires the flow of s-packets to the APL layer, which necessitates the development of soft-output packet-level FEC decoder.

Consider a s-packet $\mathbf{x}_n$, where $n = \{1, 2\}$ correspond to the MedPs and $n = 3$ corresponds to an FECP generated using the MedPs, *i.e.*,

$$\mathbf{x}_3 = \mathbf{fec}\left(\mathbf{x}_1, \mathbf{x}_2\right).$$

Where, **fec** is some FEC encoding function, which can be as simple as, *e.g.*, an XOR operation. A MedPs-FEC block, *i.e.*, group of packets $\mathbf{x}_1$, $\mathbf{x}_2$, and $\mathbf{x}_3$, is transmitted over the memoryless channel. Let the vector $\mathbf{y}_n$ contains the channel output of the packet $\mathbf{x}_n$. At the receiver, it is assumed that all the three packets (*i.e.*, $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$) are available, but they can be erroneous.

Now, our aim is to obtain joint MAP estimates $\overline{\mathbf{x}}_1$ and $\overline{\mathbf{x}}_2$ of $\mathbf{x}_1$ and $\mathbf{x}_2$, respectively, using the knowledge of $\mathbf{y}_1$, $\mathbf{y}_2$, and $\mathbf{y}_3$, *i.e.*,

$$(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2) = \arg\max_{\mathbf{x}_1,\mathbf{x}_2} p\left(\mathbf{x}_1, \mathbf{x}_2 \,|\, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right). \tag{8.1}$$

To perform MAP decoding, one needs to evaluate $p\left(\mathbf{x}_1, \mathbf{x}_2 \,|\, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right)$. Using Bayes rule, one gets

$$p\left(\mathbf{x}_1, \mathbf{x}_2 \,|\, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right) = \frac{p\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right) p\left(\mathbf{x}_1, \mathbf{x}_2\right)}{P(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3)}.$$

Thus, the MAP estimator becomes

$$\arg\max_{\mathbf{x}_1,\mathbf{x}_2} p\left(\mathbf{x}_1, \mathbf{x}_2 \,|\, \mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\right) \;=\; \arg\max_{\mathbf{x}_1,\mathbf{x}_2} p\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right) p\left(\mathbf{x}_1, \mathbf{x}_2\right), \tag{8.2}$$

where

$$\begin{aligned} p\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right) &= p\left(\mathbf{y}_1, \mathbf{y}_2 \,|\, \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_3\right) p\left(\mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right) \\ &= p\left(\mathbf{y}_1 \,|\, \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_3\right) p\left(\mathbf{y}_2 \,|\, \mathbf{x}_1, \mathbf{x}_2, \mathbf{y}_1, \mathbf{y}_3\right) p\left(\mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right). \end{aligned} \tag{8.3}$$

Given $\mathbf{x}_1$, the knowledge of $\mathbf{x}_2$ and $\mathbf{y}_3$ does not bring any additional information about $\mathbf{y}_1$. Similarly, given $\mathbf{x}_2$, the knowledge of $\mathbf{x}_1$, $\mathbf{y}_1$ and $\mathbf{y}_3$ does not bring any additional information for $\mathbf{y}_2$. Furthermore, given $\mathbf{x}_1$ and $\mathbf{x}_2$, $\mathbf{x}_3$ is well defined, *i.e.*,

$$\mathbf{x}_3 = \mathbf{fec}\left(\mathbf{x}_1, \mathbf{x}_2\right).$$

Thus, (8.3) simplifies to

$$p\left(\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3 \,|\, \mathbf{x}_1, \mathbf{x}_2\right) \;=\; p\left(\mathbf{y}_1 \,|\, \mathbf{x}_1\right) p\left(\mathbf{y}_2 \,|\, \mathbf{x}_2\right) p\left(\mathbf{y}_3 \,|\, \mathbf{fec}\left(\mathbf{x}_1, \mathbf{x}_2\right)\right). \tag{8.4}$$

Finally, using (8.4) in (8.2), the MAP estimator becomes

$$(\overline{\mathbf{x}}_1, \overline{\mathbf{x}}_2) = \arg\max_{\mathbf{x}_1,\mathbf{x}_2} p\left(\mathbf{y}_1 \,|\, \mathbf{x}_1\right) p\left(\mathbf{y}_2 \,|\, \mathbf{x}_2\right) p\left(\mathbf{y}_3 \,|\, \mathbf{fec}\left(\mathbf{x}_1, \mathbf{x}_2\right)\right) p\left(\mathbf{x}_1, \mathbf{x}_2\right). \tag{8.5}$$

8.3 Packet Structure

We will now present some hypotheses on the structure of a MedP/FECP at RTP layer, which are needed for explaining the proposed RTP-level MAP decoder in the next Section.

8.3.1 MedP Structure

Assume that the fixed-length RTP header $\mathbf{x}_{nR}$ and the variable-length RTP payload $\mathbf{x}_{np}$ are concatenated together to give the n-th variable-length RTP packet at the RTP protocol layer. Let us define a notation $\mathbf{x}_{nt}$, representing a vector, where the first index of the subscript n represents the packet number and the the second index t represents the

Figure 8.1: Types of the RTP header fields

header type. Let the channel output of this vector be represented by another vector $\mathbf{y}_{ntx}$. The length of any such vector $\mathbf{x}_{nt}$ is represented by $\ell(\mathbf{x}_{nt})$.

Various fields of the RTP header $\mathbf{x}_{nR}$ of the n-th ($n = 1, 2$) MedP may be categorized into four types, as shown in the Figure 8.1, which are

1. The *known* fields $\mathbf{k}_{nR}$, containing those fields of the RTP header, which do not change from packet-to-packet and remain constant.
2. The *redundant* fields $\mathbf{r}_{nR}$, containing those fields of the RTP header, which do not take all possible values and have explicit redundancies present inside them.
3. The *redundancy-free* fields υ_{nR}, containing those fields of the RTP header, which take all possible values and have no explicit redundancies present inside them.
4. The *xored* fields χ_{nR}, containing those fields of the RTP header, which are used to generate the corresponding fields of the FECP headers, *e.g.*, by using the simple bit-wise XOR operation.

The payload of the n-th ($n = 1, 2$) MedP $\mathbf{x}_{np}$ is assumed to be generated by a binary symmetric source.

8.3.2 FECP Structure

As explained in the previous chapter that according to RFC 5109, an FECP consists of three headers, namely, RTP header, FEC header, and FEC-level header. Let us represent these headers by vectors $\mathbf{x}_{3R}$, $\mathbf{x}_{3F}$, and $\mathbf{x}_{3L}$, respectively. The RTP header $\mathbf{x}_{3R}$ follows the same syntax as defined in Section (8.3.1).The payload of an FECP $\mathbf{x}_{3p}$ is generated by a simple bit-wise XOR operation on the payloads of the MedPs. The three headers are concatenated with the payload to construct an FECP, *i.e.*, $\mathbf{x}_3 = [\mathbf{x}_{3R}, \mathbf{x}_{3F}, \mathbf{x}_{3L}, \mathbf{x}_{3p}]$.

Now, using these notations and FECP construction process explained in the previous chapter, one can write

$$\mathbf{x}_{3R} = \mathbf{fec}_R(\mathbf{x}_{1R}, \mathbf{x}_{2R}), \tag{8.6}$$

$$\mathbf{x}_{3F} = \mathbf{fec}_F(\mathbf{x}_{1R}, \mathbf{x}_{2R}), \tag{8.7}$$

$$\mathbf{x}_{3L} = [max(\ell(\mathbf{x}_1), \ell(\mathbf{x}_2)), \mathbf{mask}], \tag{8.8}$$

and

$$\mathbf{x}_{3p} = \mathbf{fec}_p(\mathbf{x}_{1p}, \mathbf{x}_{2p}). \tag{8.9}$$

Where, $\mathbf{fec}_R$, $\mathbf{fec}_F$, and $\mathbf{fec}_p$ are some FEC encoding functionalities as defined by RFC 5109. The size of a received s-packet is assumed to be same as the size of a transmitted packet, *i.e.*, $\ell(\mathbf{x}_1) = \ell(\mathbf{y}_1)$ and $\ell(\mathbf{x}_2) = \ell(\mathbf{y}_2)$ and the vector **mask** represents the mask field in the FEC-level header, thus

$$\mathbf{x}_{3L} = [max(\ell(\mathbf{y}_1), \ell(\mathbf{y}_2)), \mathbf{mask}],$$

where $max\left(\ell\left(\mathbf{y}_1\right),\ell\left(\mathbf{y}_2\right)\right)$ is well known, and if a well defined combination pattern is used, *e.g.*, of combining two consecutive MedPs having successive SNs, the **mask** vector can become constant as well.

8.4 MAP Decoding for RTP-Level FEC

In the traditional SPS, CRC-based error detection at the MAC layer results in the removal of the corrupted packets, thus many FEC-based protocols try to recover these packets before MAC layer. The S-PPS, explained in Chapter 2, can enable FEC decoding even at higher protocol layers just before the APL layer. Furthermore, in the assumed inter-networking scenario, the multiple channel access scheme (*i.e.*, HCCA) is deployed, which is a collision-free polling-based channel access scheme, thus there is a very limited possibility of completely losing a packet in the channel.

Instead of recovering the lost or dropped MedP at RTP layer (as done by hard FEC decoder discussed in the previous chapter), as all the lower layers (*i.e.*, MAC/IP/UDP layers) of the S-PPS are able to forward even erroneous s-packets to the RTP layer, we propose to recover the MedPs from the erroneously received MedPs-FECP block. For the said purpose, an RTP-level MAP decoder is proposed, which perform JPCD to recover the MedPs, utilizing the channel likelihood in conjunction with the protocol redundancy. The protocol redundancy includes, the explicit inter and intra-packet redundancy in the MedP header, the redundancy due to the presence of a redundant FECP generated using a well defined XOR operation, and the redundancy due to the structure of an FECP.

Let us assume that a MedPs-FECP block is transmitted over a memoryless channel, at the receiver an erroneous MedPs-FECP block reaches the RTP layer using S-PPS and the transmission order is insured at the receiver, *i.e.,* FECP $\mathbf{y}_3$ is followed by the corresponding MedPs $\mathbf{y}_1$ and $\mathbf{y}_2$. Now, by considering the independence of the payload and the header of the s-packets inside the received MedPs-FECP block, we will present the MAP decoding for each separately.

8.4.1 MAP Decoding for RTP Header

Our aim is to obtain the joint MAP estimates $\overline{\mathbf{x}}_{1R}$ and $\overline{\mathbf{x}}_{2R}$ of $\mathbf{x}_{1R}$ and $\mathbf{x}_{2R}$, respectively, using the knowledge of the received MedP1 RTP header $\mathbf{y}_{1R}$, MedP2 RTP header $\mathbf{y}_{2R}$, FECP RTP header $\mathbf{y}_{3R}$, FECP FEC header $\mathbf{y}_{3F}$, and FECP FEC-level header $\mathbf{y}_{3L}$. The MAP decoder is given by

$$\left(\overline{\mathbf{x}}_{1R},\overline{\mathbf{x}}_{2R}\right) = \arg\max_{\mathbf{x}_{1R},\mathbf{x}_{2R}} p\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\,|\,\mathbf{y}_{1R},\mathbf{y}_{2R},\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\right), \tag{8.10}$$

which, using the Bayes rule, becomes

$$\left(\overline{\mathbf{x}}_{1R},\overline{\mathbf{x}}_{2R}\right) = \arg\max_{\mathbf{x}_{1R},\mathbf{x}_{2R}} p\left(\mathbf{y}_{1R},\mathbf{y}_{2R},\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) p\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right). \tag{8.11}$$

Where,

$$\begin{aligned} p\left(\mathbf{y}_{1R},\mathbf{y}_{2R},\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) &= p\left(\mathbf{y}_{1R},\mathbf{y}_{2R}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) \\ &\quad\; p\left(\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R},\mathbf{y}_{1R},\mathbf{y}_{2R}\right) \end{aligned}$$

simplifies to

$$p\left(\mathbf{y}_{1R},\mathbf{y}_{2R},\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) = p\left(\mathbf{y}_{1R}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right)p\left(\mathbf{y}_{2R}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R},\mathbf{y}_{1R}\right) p\left(\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R},\mathbf{y}_{1R},\mathbf{y}_{2R}\right). \quad (8.12)$$

Given $\mathbf{x}_{1R}$, the knowledge of $\mathbf{x}_{2R}$ brings no additional information for $\mathbf{y}_{1R}$, and given $\mathbf{x}_{2R}$, the knowledge of $\mathbf{x}_{1R}$ brings no additional information for $\mathbf{y}_{2R}$. Similarly, given $\mathbf{x}_{1R}$ and $\mathbf{x}_{2R}$, the knowledge of $\mathbf{y}_{1R}$ and $\mathbf{y}_{2R}$ bring no additional information, thus (8.12) becomes

$$p\left(\mathbf{y}_{1R},\mathbf{y}_{2R},\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) = p\left(\mathbf{y}_{1R}\,|\,\mathbf{x}_{1R}\right)p\left(\mathbf{y}_{2R}\,|\,\mathbf{x}_{2R}\right) p\left(\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right). \quad (8.13)$$

Using the encoding rules of RTP-level FEC given by (8.6), (8.7), and (8.8), one can simplify the term $p\left(\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right)$ of (8.13). Assuming independence of the headers of an FECP, one gets

$$p\left(\mathbf{y}_{3R},\mathbf{y}_{3F},\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) = p\left(\mathbf{y}_{3R}\,|\,\mathbf{fec}_R\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right)\right) p\left(\mathbf{y}_{3F}\,|\,\mathbf{fec}_F\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right)\right)p\left(\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right). \quad (8.14)$$

Finally, using (8.13) and (8.14), the MAP decoder of (8.11) becomes

$$\left(\overline{\mathbf{x}}_{1R},\overline{\mathbf{x}}_{2R}\right) = \arg\max_{\mathbf{x}_{1R},\mathbf{x}_{2R}} p\left(\mathbf{y}_{1R}\,|\,\mathbf{x}_{1R}\right)p\left(\mathbf{y}_{2R}\,|\,\mathbf{x}_{2R}\right)p\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right)p\left(\mathbf{y}_{3L}\,|\,\mathbf{x}_{1R},\mathbf{x}_{2R}\right) p\left(\mathbf{y}_{3R}\,|\,\mathbf{fec}_R\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right)\right)p\left(\mathbf{y}_{3F}\,|\,\mathbf{fec}_F\left(\mathbf{x}_{1R},\mathbf{x}_{2R}\right)\right). \quad (8.15)$$

The MAP estimator presented can be applied independently on each field of the RTP header of a MedP, as all the fields are independent in the RTP header. Furthermore, one can use the explicit inter-packet and intra-packet redundancies present inside each field. Now, MAP decoding for each field of the RTP header would be provided, keeping in view the FECP construction process explained in the previous chapter (see Section 7.3.1).

Known fields (V): The V field of the RTP header of a MedP belongs to the known field category $\mathbf{k}_{nR}$ and it remains constant from packet-to-packet, thus it does not need any estimation.

Xored fields (P, X, CC, M, PT, and TS): The P, X, CC, M, PT, and TS fields of the RTP header of a MedP belongs to the xored field category χ_{nR}. Each of these fields does not follow well defined pattern from packet-to-packet, thus it can be assumed that there is no inter-packet redundancy inside these fields, leaving no inter-packet dependency, *i.e.*, $p\left(\chi_{1R},\chi_{2R}\right) = p\left(\chi_{1R}\right)p\left(\chi_{2R}\right)$. Furthermore, there is no explicit intra-packet redundancy inside these fields (*i.e.*, they can take any possible values with equal probability), thus one can assume that the bits inside these fields are independent and equally likely, *i.e.*,

$$p\left(\chi^i_{nR}\right) = \begin{cases} \frac{1}{2},\ \chi^i_{nR} = 0 \\ \frac{1}{2},\ \chi^i_{nR} = 1. \end{cases}$$

Each of the fields P, X, CC, M, PT, and TS is used to generate corresponding field in the FEC header of an FECP (see observations of Section 7.3.1.2), *i.e.*, for each field

$$\chi^i_{3F} = \mathbf{xor}\left(\chi^i_{1R},\chi^i_{2R}\right).$$

The RTP header fields P, X, CC, M, and PT are not used to generate any field of the RTP header of an FECP (see observations of Section 7.3.1.1), and are only used to generate the corresponding fields of the FEC header of an FECP (see observations of Section 7.3.1.2). Thus, the MAP decoder for each of these fields can be written as

$$\begin{aligned}\left(\overline{\chi}^i_{1R}, \overline{\chi}^i_{2R}\right) &= \arg \max_{\chi^i_{1R}, \chi^i_{2R}} p\left(\mathbf{y}^i_{1R\chi} \,|\, \chi^i_{1R}\right) p\left(\mathbf{y}^i_{2R\chi} \,|\, \chi^i_{2R}\right) \\ & \quad p\left(\mathbf{y}^i_{3F\chi} \,|\, \mathbf{xor}\left(\chi^i_{1R}, \chi^i_{2R}\right)\right). \end{aligned} \tag{8.16}$$

On the contrary, the TS field of MedP2 is also used to generate TS field of the RTP header of an FECP (see observations of Section 7.3.1.1), *i.e.*, $\chi_{3R} = \chi_{2R}$. Thus, the MAP decoder for TS field can be written as

$$\begin{aligned}\left(\overline{\chi}^i_{1R}, \overline{\chi}^i_{2R}\right) &= \arg \max_{\chi^i_{1R}, \chi^i_{2R}} p\left(\mathbf{y}^i_{1R\chi} \,|\, \chi^i_{1R}\right) p\left(\mathbf{y}^i_{2R\chi} \,|\, \chi^i_{2R}\right) \\ & \quad p\left(\mathbf{y}^i_{3F\chi} \,|\, \mathbf{xor}\left(\chi^i_{1R}, \chi^i_{2R}\right)\right) p\left(\mathbf{y}^i_{3R\chi} \,|\, \chi^i_{2R}\right). \end{aligned} \tag{8.17}$$

Redundant fields (SN): The SN field of the RTP header of a MedP belongs to the redundant field category $\mathbf{r}_{nR}$. The SN field is not used to generate any field of the RTP header of an FECP (see observations of Section 7.3.1.1), but the SN base field in the FEC header is set to the SN field of the MedP1 (see observations of Section 7.3.1.2). The mask field inside FEC-level header is assumed to be of 16 bits (by fixing L = 0 for the sake of simplicity) and is completely defined by $\mathbf{r}_{1R}$ and $\mathbf{r}_{2R}$ (see observations of Section 7.3.1.3). We know, $\mathbf{r}_{2R} = \mathbf{r}_{1R} + i$, where $i = \{1, ...\ell\left(\mathbf{mask}\right) - 1\}$. Let the channel output of the SN base field of FEC header and the mask field of FEC-level header be represented by vectors $\mathbf{y}_{3Fr}$ and $\mathbf{y}_{3Lr}$, respectively. Using these observation, the MAP decoder for the SN field of a MedP is given by

$$\begin{aligned}\left(\overline{\mathbf{r}}_{1R}, \overline{\mathbf{r}}_{2R}\right) &= \arg \max_{\mathbf{r}_{1R}, \mathbf{r}_{2R}} p\left(\mathbf{y}_{1Rr} \,|\, \mathbf{r}_{1R}\right) p\left(\mathbf{y}_{2Rr} \,|\, \mathbf{r}_{2R}\right) p\left(\mathbf{r}_{1R}, \mathbf{r}_{2R}\right) \\ & \quad p\left(\mathbf{y}_{3Fr} \,|\, \mathbf{r}_{1R}\right) p\left(\mathbf{y}_{3Lr} \,|\, \mathbf{r}_{1R}, \mathbf{r}_{2R}\right). \end{aligned} \tag{8.18}$$

Inter-packet redundancies between successive MedPs (due to the increasing SN from packet-to-packet) can bring intra-packet redundancies, but no knowledge is assumed to be available from the MAP decoding of the previous MedPs-FEC block, therefore no intra-packet redundancy is assumed to be available in the SN field of the MedP1, but given SN field of MedP1 there is an explicit intra-packet redundancy in the SN field of the MedP2 (*i.e.*, SN field of the MedP2 cannot take any possible value with equal probability).

For SN field, there is an explicit inter-packet redundancy between the MedPs of a MedPs-FEC block. Let Ω_{SN} represents the set of all possible pairs $(\mathbf{r}_{1R}, \mathbf{r}_{2R})$, *i.e.*,

$$\left(\mathbf{r}_{1R}, \mathbf{r}_{2R}\right) \, \epsilon \, \Omega_{SN},$$

and we know

$$\Omega_{SN} = \cup_{i=1}^{i=15} \Omega^i_{SN}.$$

Where,

$$\Omega^i_{SN} = \left(\mathbf{r}_{1R}, \mathbf{r}_{1R} + i\right)$$

represents the subset of all possible pairs $(\mathbf{r}_{1R}, \mathbf{r}_{2R})$ for which the i-th bit of the mask field

would be set. We now have $p(\mathbf{r}_{1R}, \mathbf{r}_{2R}) = \frac{1}{|\Omega_{SN}|}$, with $|\Omega_{SN}|$ denoting the cardinal number of Ω_{SN} given by $|\Omega_{SN}| = \sum_{i=1}^{i=15} |\Omega_{SN}^{i}|$, where

$$|\Omega_{SN}^{i}| = 2^{16} - i.$$

Thus, the MAP decoder would select the best pair from Ω_{SN}, using the MAP estimator

$$\begin{aligned}(\overline{\mathbf{r}}_{1R}, \overline{\mathbf{r}}_{2R}) &= \arg \max_{(\mathbf{r}_{1R}, \mathbf{r}_{2R}) \,\epsilon\, \Omega_{SN}} p(\mathbf{y}_{1Rr} \,|\, \mathbf{r}_{1R})\, p(\mathbf{y}_{2Rr} \,|\, \mathbf{r}_{2R}) \\ &\quad p(\mathbf{y}_{3Fr} \,|\, \mathbf{r}_{1R})\, p(\mathbf{y}_{3Lr} \,|\, \mathbf{r}_{1R}, \mathbf{r}_{2R}). \end{aligned} \tag{8.19}$$

To reduce the number of computations, in this thesis only the successive MedPs are combined to generate an FECP, thus limiting the search space to the subset Ω_{SN}^{1}.

Redundant-free fields (SSRC): The SSRC field of the RTP header of a MedP belongs to the redundant-free field category υ_{nR} . We know that the SSRC field in the RTP header of an FECP is similar to the SSRC field of MedP1 or MedP2 (see observations of Section 7.3.1.1), *i.e.*, $\upsilon_{3R} = \upsilon_{1R} = \upsilon_{2R}$, thus the MAP decoder for this field becomes

$$\begin{aligned}(\overline{\upsilon}_{1R}, \overline{\upsilon}_{2R}) &= \arg \max_{\upsilon_{1R}, \upsilon_{2R}} p(\mathbf{y}_{1R\upsilon} \,|\, \upsilon_{1R})\, p(\mathbf{y}_{2R\upsilon} \,|\, \upsilon_{2R})\, p(\upsilon_{1R}, \upsilon_{2R}) \\ &\quad p(\mathbf{y}_{3R\upsilon} \,|\, \upsilon_{1R}). \end{aligned} \tag{8.20}$$

There is no explicit intra-packet redundancy inside the SSRC field (*i.e.*, SSRC can take any possible value with equal probability), thus one can assume that the bits inside υ_{nR} are independent and equally likely. But, given that $\upsilon_{1R}^{i} = \upsilon_{2R}^{i}$, we have an evident inter-packet redundancy for this field, *i.e.*,

$$p\left(\upsilon_{1R}^{i}, \upsilon_{2R}^{i}\right) = \begin{cases} \frac{1}{2}, \upsilon_{1R}^{i} = 0, \upsilon_{2R}^{i} = 0 \\ 0,\ \upsilon_{1R}^{i} = 0, \upsilon_{2R}^{i} = 1 \\ 0,\ \upsilon_{1R}^{i} = 1, \upsilon_{2R}^{i} = 0 \\ \frac{1}{2},\ \upsilon_{1R}^{i} = 1, \upsilon_{2R}^{i} = 1. \end{cases}$$

Thus, the MAP decoder for the SSRC field becomes

$$\begin{aligned}\overline{\upsilon}_{1R}^{i} &= \arg \max_{\upsilon_{1R}^{i}} p\left(\mathbf{y}_{1R\upsilon}^{i} \,|\, \upsilon_{1R}^{i}\right) p\left(\mathbf{y}_{2R\upsilon}^{i} \,|\, \upsilon_{1R}^{i}\right) \\ &\quad p\left(\mathbf{y}_{3R\upsilon}^{i} \,|\, \upsilon_{1R}^{i}\right). \end{aligned} \tag{8.21}$$

8.4.2 MAP Decoding for RTP Payload

The MAP decoder for the payload, by assuming independence of the payloads $\mathbf{x}_{1p}$ and $\mathbf{x}_{2p}$, can be deduced from (8.5), *i.e.*,

$$\begin{aligned}(\overline{\mathbf{x}}_{1p}, \overline{\mathbf{x}}_{2p}) &= \arg \max_{\mathbf{x}_{1p}, \mathbf{x}_{2p}} p(\mathbf{y}_{1p} \,|\, \mathbf{x}_{1p})\, p(\mathbf{y}_{2p} \,|\, \mathbf{x}_{2p})\, p(\mathbf{x}_{1p})\, p(\mathbf{x}_{2p}) \\ &\quad p(\mathbf{y}_{3p} \,|\, \mathbf{fec}_{p}(\mathbf{x}_{1p}, \mathbf{x}_{2p})). \end{aligned} \tag{8.22}$$

Moreover, the bits of the payloads are assumed to be independent and FEC encoding function $\mathbf{fec}_{p}$ is a simple bit-wise XOR operation **xor** (see observations of Section 7.3.1.4),

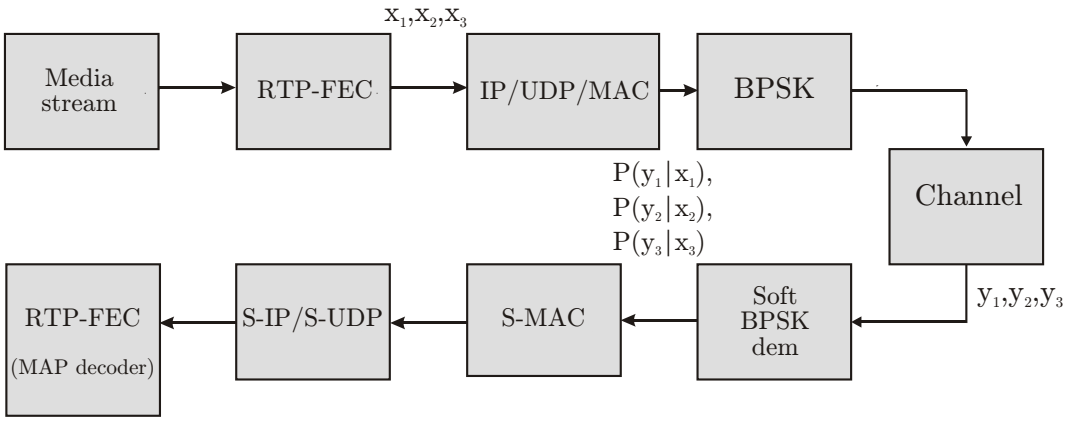

Figure 8.2: Simulation Block Diagram: The S-PPS is used at the receiver, the soft BPSK demodulation technique is used at the PHY layer, and erroneous packets are used by the MAP decoder at the RTP-FEC layer.

thus we have

$$\begin{aligned}\left(\overline{\mathbf{x}}^i_{1p}, \overline{\mathbf{x}}^i_{2p}\right) &= \arg\max_{\mathbf{x}^i_{1p}, \mathbf{x}^i_{2p}} p\left(\mathbf{y}^i_{1p} \,|\, \mathbf{x}^i_{1p}\right) p\left(\mathbf{y}^i_{2p} \,|\, \mathbf{x}^i_{2p}\right) p\left(\mathbf{x}^i_{1p}\right) p\left(\mathbf{x}^i_{2p}\right) \\ &\quad p\left(\mathbf{y}^i_{3p} \,|\, \mathbf{xor}\left(\mathbf{x}^i_{1p}, \mathbf{x}^i_{2p}\right)\right). \end{aligned} \tag{8.23}$$

8.5 Simulation results

The block diagram of the simulation model is shown in Figure 8.2. The transmitter remains unchanged, while the S-PPS is used at the receiver, where the soft BPSK demodulator is deployed at the PHY layer and the RTP-level MAP decoding (of Section 8.4) is performed at the RTP-FEC layer . The APPs are assumed to be reaching the MAP decoder, functioning at the RTP-FEC layer, due to the use of S-PPS.

Simulation results for the transmission of video stream with the MTU of 1500 bytes over Rayleigh fading channel are provided in the Figure 8.3. The channel specifications are as provided in Section 7.5. MAP decoding is performed on the RTP header fields and the payload independently as outlined in Section 8.4. To have a fair comparison, simulation results for the retransmission scheme, where 2 retransmissions are allowed, are also shown. The retransmission scheme uses the SPS at the receiver, while the proposed MAP decoder uses S-PPS. Compared to the retransmission scheme, a gain of slightly more than 10 dB is observed using the proposed MAP decoder at a PER of 10^{-2}.

Assume that the given error-resilient video decoder can tolerate PER P^e of about 10%. Analyzing the simulation results, the FEC scheme using the proposed MAP decoder can manage to keep $P^e < 10\%$ till the channel degrades to 18.5 dB. On the contrary, as we have already seen in the previous chapter, using an error-resilient video decoder in conjunction with the retransmission scheme, the system would be able to keep $P^e < 10\%$ till the channel degrades to $30\,dB$. Note that for the retransmission scheme the goodput depends only on the PER P^e ($G^r = 1 - P^e$, and P^e only depends on the SNR), thus only on the channel condition and not on the remaining PER after r retransmissions P^e_r. Therefore,

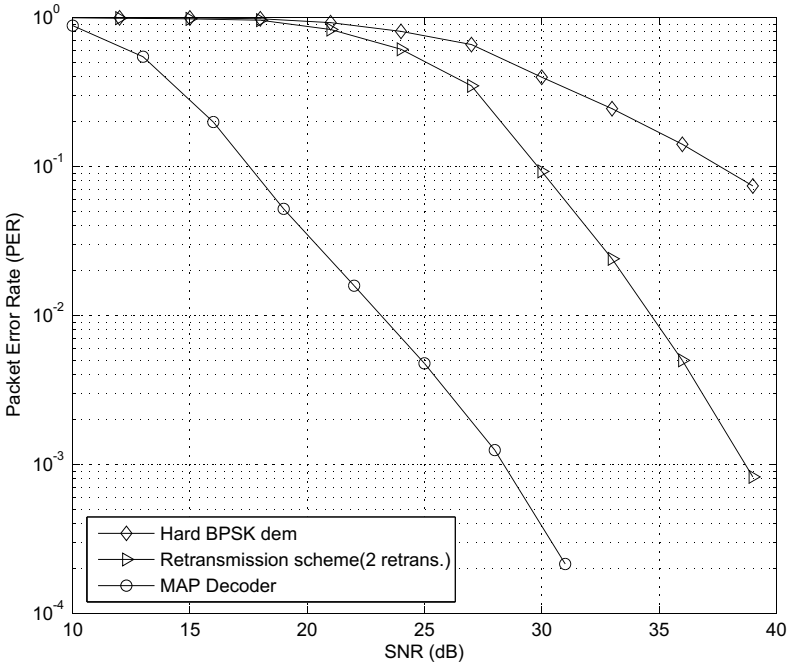

Figure 8.3: Simulation results (PER vs. SNR): Video stream (MTU of 1500 bytes) transmission over Rayleigh channel

as the channel degrades the goodput for the retransmission scheme decreases. The MAP decoder performs more efficiently due to the use of the soft information that reaches the RTP layer using S-PPS.

8.6 Limitations

The H.264 video decoder is particularly sensitive to bit errors and can crash after a single bit error. Though its error-resilient features can be exploited, it needs to know which packet is erroneous or lost in order to recover it. The presented RTP-level MAP decoding suffers from a practical anomaly as at the RTP layer there is no method to distinguish correctly recovered MedPs after MAP decoding.

Residual redundancy in the video stream can be used to detect erroneously recovered MedPs. The robust video decoders, like the one presented in [55; 84], can detect erroneous NALU inside the RTP packet provided that the RTP-level MAP decoding error results in an H.264 encoder syntax error. In [83] several error detection methods are presented and evaluated through simulation. However, the reliability of syntax analysis based error detection is limited and residual error can still cause desynchronized decoding. In such case, a resynchronization mechanism proposed in [131] could be used. Furthermore, watermarking-based error detection mechanism [132] can also be used, but requires modifications to the transmitter.

8.7 Conclusions

An RTP-level FEC decoder is presented is this chapter, which utilizes (i) the introduced redundant FECP, (ii) the explicit redundancy present in the RTP header, and (iii) the likelihood information from the channel. It takes full benefit from the S-PPS by exploiting the soft information relayed to it, which makes it effective even with erroneous packets. Furthermore, it shows more robustness towards the bit errors due to the use of the structural redundancies.

The comparison with the traditional retransmission scheme showed that the MAP decoding would be able to tolerate more channel degradation then the retransmission scheme. It reduces the PER and gives more compensation margin in case the channel degrades, while causing no delay and hindrance to the flow of soft information, through the RTP layer, to the APL layer. The MAP decoder coupled with the robust video error detectors and decoders (at APL layer) is an effective tool to mitigate the errors left by the lower layers of S-PPS.

Though, only a simple case of EEP is presented, the decoder can easily be adapted for UEP. The decoder proposed can also be extended for more complex and robust FEC codes, like RS codes. Furthermore, it can be deployed at the lower layers, *e.g.*, at UDP or MAC layers, where one can also exploit redundancy available due to the checksums/CRC.

Chapter 9

Conclusion

In this chapter we conclude the thesis and highlight some future directions and applications.

9.1 Contributions

The robust decoding tools proposed by several researchers for performance improvement neglect the complete protocol stack, thus require several non-compliant features often not available with the SPS. In this thesis, we studied the global framework of video broadcast over WiMAX network and subsequent rebroadcast over WiFi network. It encompasses a complete transmission chain needed for the inter-networking of WiMAX and WiFi ranging from the APL layer of the source, across the network, till the video decoder at APL layer of the destination. It uses a modified protocol stack, *i.e.*, S-PPS, often required by several robust video decoding techniques like JSCD, at the receiver, while the transmitter functionality remains unchanged. Issues that can limit the full functionality of S-PPS at the receiver, *e.g.*, FS and packet-level FEC decoding, are studied and robust solutions are proposed.

Several S-PPS compliant strategies for FS are presented. The trellis-based FS technique gives high performance, but is not preferable for the delay-constrained broadcast. It is thus modified to provide an efficient performance-complexity trade-off. Its ST-based FS variant also decreases the latency by performing FS window-by-window, *i.e.*, on the small portion of the burst at a time. Despite that several applications might prefer on-the-fly FS instead of high performance FS. Therefore, we have also presented an on-the-fly FS technique, which is based on the 3S automaton and uses the robust header recovery and Bayesian hypothesis test. Finally, at the end of this thesis we have presented a robust packet-level FEC decoder, which enables robust rebroadcast over the WiFi-link. This decoder, while functioning at the RTP layer, makes full use of the S-PPS, and act as the last front to reduce PER before the APL layer. The FS techniques and the packet-level decoder proposed are based on the idea of JPCD, and benefit from the soft information provided by the PHY layer and from the redundancies available due to the structure of the header.

Simulations results are also provided for the proposed FS and the packet-level decoding techniques. We have illustrated FS techniques for the WiMAX MAC layer, but the proposed FS techniques can easily be applied to other situations. The trellis-based approach brings high performance FS, the ST-based approach brings low-delay and reduced-complexity FS, and the 3S automaton brings low performance but on-the-fly FS to the

S-PPS. Similarly, the robust MAP decoder at RTP layer brings valuable packet-level FEC decoder to the S-PPS.

Practically, these proposals will result in reducing the amount of packets that need to be dropped and increase the number of packets relayed to the APL layer. Both of these tools complemented with the robust JSCD techniques at APL layer would provide considerable improvements in the quality of the received video.

9.2 Future Work and Applications

As a future direction, the available inter-layer redundancy can be put to use to further improve the performance of the tools presented in this thesis. Furthermore, the fact that the packet begins with several headers, *i.e.*, MAC, IP, UDP, and RTP headers followed by a payload, can bring performance gain to the FS. This would require detail investigation of the redundancies in all of these headers. Furthermore, source redundancies in the H.264 can also be utilized for cross-layer FS. Cross-layer FS has already been suggested in [23], where *a priori* information about the statistical prevalence of ones and zeros in the H.264 bitstream drive the FS metric.

A complete FEC decoder with adaptive functionality would be an interesting future direction along with the extension of the proposed packet-level FEC decoder to more robust codes.

9.2.1 FS Applications

Now we will analyze the applicability of the proposed FS techniques on other network and protocols. Simple Data Link (SDL) and IEEE 802.11n are provided as applications of the proposed FS techniques.

9.2.1.1 SDL Framing

The SDL [133] framing protocol is uniquely designed to align frames in high-speed communications. SDL can identify the boundaries of variable-length frames by using length indication and HEC check. SDL framing protocol can be used to transmit IP packets efficiently over network. Once SDL frames are aligned, IP packets can be immediately identified. The frame consists of header, payload and trailer. The header contains at least the Frame Length (FL) and HEC field as shown in Figure 9.1. The payload can contain data such as an IP packet. The trailer is optional and contains a Frame Check Sequence (FCS).

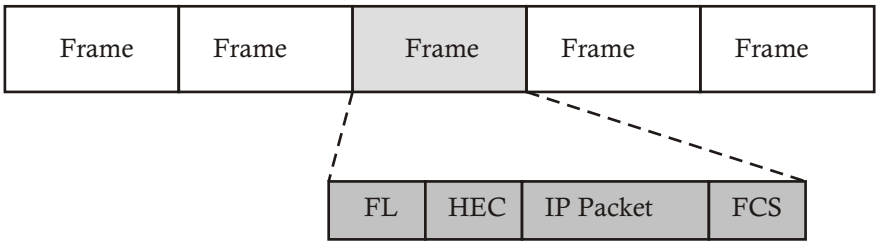

Figure 9.1: SDL frame Aggregation

In SDL, due to continuous bitstream communication, the trellis-based FS presented in Chapter 4 is not applicable, but ST-based FS presented in Chapter 5 is readily applicable, where the bitstream can be divided in to several windows of interest and FS can be performed on each window. Similarly, the 3S automaton can be applied as an on-the-fly FS solution.

9.2.1.2 IEEE 802.11n Packet Aggregation

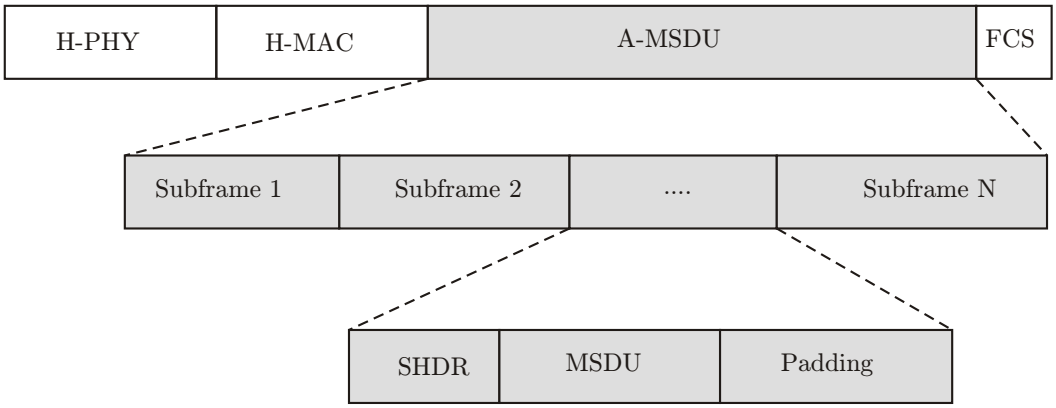

Figure 9.2: MSDU Aggregation (AMSDU)

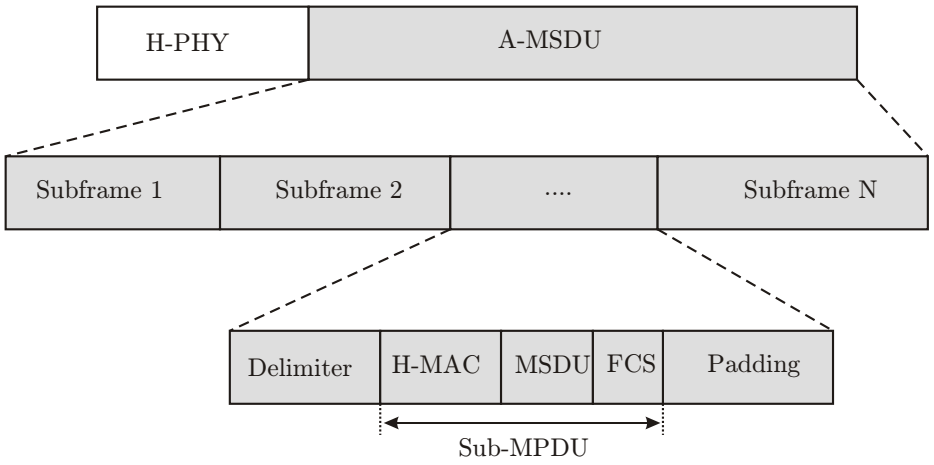

Figure 9.3: MPDU-aggregation (AMPDU)

One of the applications of the proposed FS approach lies in IEEE 802.11n [101], which builds on previous 802.11 standards by adding packet aggregation to the MAC layer. There are two ways to perform packet aggregation at the MAC layer. The first technique is by concatenating several MAC SDUs (MSDUs) to form the data payload of a large MAC PDU (MPDU). The PHY header and MAC header, along with the FCS, are then appended to form the PHY SDU (PSDU). This technique is known as MSDU Aggregation (AMSDU) and is shown in Figure 9.2. The second technique is called MPDU-aggregation (AMPDU).

It begins with each MSDU appending with its own MAC header and FCS to form a sub-MPDU. An MPDU delimiter is then inserted before each sub-MPDU. Padding bits are also inserted so that each sub-MPDU is a multiple of 4 bytes in length, which can facilitate FS at the receiver. Then, all the sub-MPDUs are concatenated to form a large PSDU. Figure 9.3 shows the format for AMPDU.

The three proposed FS methods are readily applicable to these packet aggregation scenarios.

9.2.2 Packet-Level FEC Decoder Applications

The FEC protection and the packet-level MAP decoder presented in this thesis can be extended to other rebroadcast scenarios *e.g.*, receiving video from mobile TV and then rebroadcasting it over WiFi network.

DVB-H [33; 34] is one of prevalent mobile TV format. DVB-H is a technology designed to enable the broadcasting of multimedia content to mobile devices. After receiving the mobile TV, WiFi-rebroadcasting is often consider as a gap-filler approach for indoor reception. It is noticed that rebroadcasting over the WiFi-link increases the PER significantly because the link-layer retransmissions are omitted. Thus, RTP-level FEC encoding/decoding can be used to decrease the PER and increase the performance.

Appendix A

Appendices of different Chapters

A.1 WiMAX PHY FRAME

A.1.1 OFDM PHY DL Sub-frame

Each DL sub-frame is transmitted as follows:

- *Preamble*:

It is used for synchronization, and is the first OFDM symbol of the frame.

- *Frame Control Header* (FCH):

One OFDM symbol long FCH follows the preamble. It provides the frame configuration information such as MAP message length and coding scheme and usable sub-channels. It has Down Link Frame Prefix (DLFP) to specify burst profiles and length of burst profiles of one or several DL bursts immediately following FCH.

- *DL-MAP*:

It indicates burst profile, location and duration of zones within the DL frame. DL-MAP message if present shall be the first MAP PDU transmitted after FCH. It is modulated with BPSK rate 1/2 with the mandatory code scheme.

- *UL-MAP*:

It provides the sub-channel and slot allocation and other control information for the UL sub-frame. It should immediately follow DL-MAP(If one is transmitted) or the DLFP.

- *DL Channel Descriptor* (DCD):

It is transmitted by BS at a periodical interval to define the characteristics of a DL frame.

- *UL Channel Descriptor* (UCD):

It is transmitted by BS at a periodical interval to define the characteristics of an UL frame.

A.1.2 OFDM PHY UL Sub-frame

The UL sub-frame is divided into following three slots:

- Contention slots allowing initial ranging.
- Contention slots allowing bandwidth requests.
- One or many UL bursts.

A.2 WIMAX MAC PDU

There can be two types of MAC PDU headers:

- The *Generic MAC Header* (GMH):

This is the header of MAC packets containing either MAC management messages or CS data. The CS data may be user data or other higher layer management data. The generic MAC header is the only one used in the DL.

- The *Bandwidth Request Header* (BRH):

For MAC packets with this type of header format, the MAC header is not followed by any MPDU payload and CRC. This header name has been introduced by the IEEE 802.16e amendment. Previously, in IEEE 802.16-2004, the bandwidth request header was defined to request additional bandwidths.

A.2.1 Generic MAC Header (GMH)

As we are considering only DL case, where the connection is already established and MAC PDUs inside BB contains only CS data, so only the GMH is possible inside BB. Format of GMH as specified in IEEE 802.16-2004 [32] is illustrated in Figure A.1, and its various fields are described below:

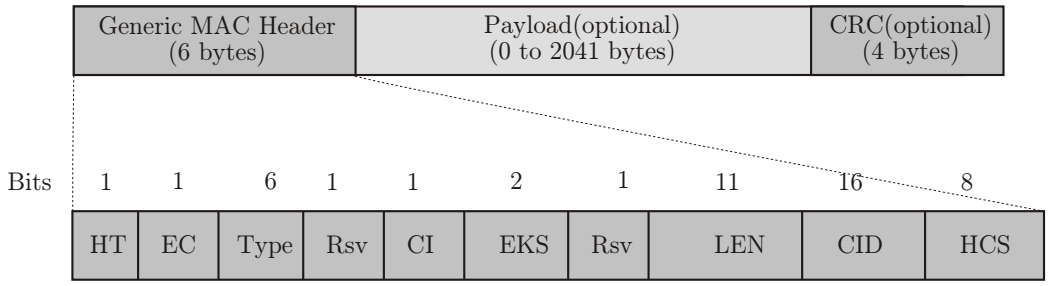

Figure A.1: Generic MAC header

- *Header Type* (HT):

This field is single bit, set to 0 for generic MAC header.

- *Encryption Control* (EC):

This field specifies whether payload is encrypted or not and is set to 0 when payload is not encrypted and to 1 when it is encrypted.

- *Type*:

This field indicates the subheaders and special payload types present in the message payload (5 subheaders possible, see below).

- *Reserved* (Rsv):

This field is two bit, and is set to 00.

- *Extended Subheader Field* (ESF):

This field is a single bit, set to 1 if the extended subheader is present and follows the generic MAC header immediately (applicable in both the DL and UL).

- *CRC Indicator* (CI):

This field is a single bit, set to 1 if CRC is included and is set to 0 if no CRC is included.

- *Encryption Key Sequence* (EKS):

This field is two bit, it is the index of the Traffic Encryption Key (TEK) and initialization vector used to encrypt the payload. Evidently, this field is only meaningful if the EC field is set to one.

- *Length* (LEN):

This field is 11 bit, it specifies the length in bytes of the MAC PDU including the MAC header and the CRC, if present.

- *Connection IDentifier* (CID):

This field is 16 bit, and represents the connection identifier of the user.

- *Header Check Sequence* (HCS):

This field is 8 bit, and is used to detect errors in the header

In MAC PDU five types of subheaders may be present depending on the Type field in the generic header:

1. The Mesh Sub header: used only in Mesh Mode not in PMP mode.
2. The Grant Management sub header: Used in the UL only.
3. Fragmentation sub header: No fragmentation used.
4. The FAST-FEEDBACK allocation sub header: only used in OFDMA PHY layer.
5. Packing sub header: No packing is used.
6. The header without payload (Type I and II), only used in the UL, has the same size as the generic MAC header, but the fields differ.

A.2.2 The Bandwidth Request Header (BRH)

Bandwidth request header, as shown in Figure A.2, has the same size as the generic MAC header, but the fields differ. Bandwidth request PDU consists of only header and does not contain a payload. Fields of the header are provided below:

- *Header Type* (HT):

This field is single bit, set to 1 for BRH.

- *Encryption Control* (EC):

This field is a single bit, set to 0, indicating no encryption.

- *Type*:

This field is 3 bit, indicates the type of bandwidth request header and takes two values , '000' for incremental and '001' for aggregate.

- *Bandwidth Request* (BR):

This field is 19 bit, and indicates the number of bytes requested.

- *Connection IDentifier* (CID):

This field is 16 bit, and represents the connection identifier of the connection for which UL bandwidth is requested.

- *Header Check Sequence* (HCS):

This field is 8 bit, and is used to detect errors in the header.

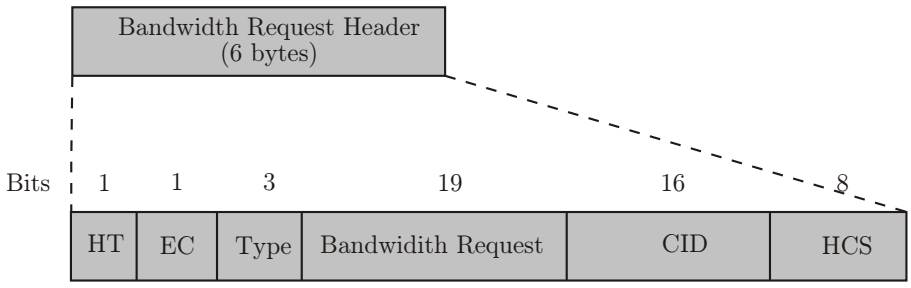

Figure A.2: Bandwidth Request Header

A.3 UDP Header

Time-sensitive applications often use UDP because dropping packets is preferable to waiting for delayed packets, which may not be an option in a real-time system. UDP uses a simple transmission model without implicit hand-shaking dialogues for guaranteeing reliability, ordering, or data integrity. Thus, UDP provides an unreliable service and datagrams may arrive out of order, appear duplicated, or go missing without notice. Different fields of the UDP header are provided below:

- *Source Port*: 16 bits

It is port number of the sender. It describes where a reply packet should be sent. This can actually be set to zero if it is not used. For example, sometimes we don't require a reply packet, and the packet can then be set to source port zero. In most implementations, it is set to some port number.

- *Destination Port*: 16 bits

It is port number to which this packet is addressed to. This is required for all packets, as opposed to the source port of a packet.

- *Length*: 16 bits.

The length in bytes of the UDP header and the encapsulated data. The minimum value for this field is 8. The field size sets a theoretical limit of 65,535 bytes (8 byte header + 65,527 bytes of data) for a UDP datagram. The practical limit for the data length which is imposed by the underlying IPv4 protocol is 65,507 bytes (65,535 - 8 byte UDP header - 20 byte IP header).

- *Checksum*: 16 bits

Computed as the 16-bit one's complement of the one's complement sum of a pseudo header of information from the IP header, the UDP header, and the data, padded as needed with zero bytes at the end to make a multiple of two bytes. If the checksum is cleared to zero, then checksuming is disabled. If the computed checksum is zero, then this field must be set to 0xFFFF.

A.4 UDP-Lite Header

UDP-Lite is a connectionless protocol, very similar to UDP. Unlike UDP, where either all or none of a packet is protected by a checksum, UDP-Lite allows for partial checksums that only cover part of a datagram, and will therefore deliver packets that have been partially corrupted. It is designed for multimedia protocols, such as voice over IP, in which receiving a packet with a partly damaged payload is better than receiving no packet at all.

Since most modern link layers protect the carried data with a strong CRC and will discard damaged packets, making effective use of UDP-Lite requires the link layer to be aware of the network-layer data being carried. Since no current TCP/IP stacks implement such cross-layer interactions, making effective use of UDP-Lite currently requires specially modified device drivers. Different fields of the UDP-lite header are provided below:

- *Source Port*: 16 bits

The port number of the sender. Cleared to zero if not used.

- *Destination Port*: 16 bits

The port number this packet is addressed to.

- *Checksum coverage*: 16 bits

The number of bytes, counting from the first byte of the UDP-Lite header, covered by the checksum. The UDP-Lite header MUST always be covered by the checksum. Despite

this requirement, the Checksum Coverage is expressed in bytes from the beginning of the UDP-Lite header in the same way as for UDP. A Checksum Coverage of zero indicates that the entire UDP-Lite packet is covered by the checksum. A UDP-Lite packet with a Checksum Coverage value of 1 to 7 MUST be discarded by the receiver. Irrespective of the Checksum Coverage, the computed Checksum field MUST include a pseudo-header, based on the IP header. UDP-Lite packets with a Checksum coverage greater than the IP length MUST also be discarded.

- *Checksum*: 16 bits

Computed as the 16-bit one's complement of the one's complement sum of a pseudo-header of information collected from the IP header, the number of bytes specified by the Checksum coverage field (starting at the first byte in the UDP-Lite header), virtually padded with a zero byte at the end if necessary to make a multiple of two bytes. Prior to computation, the checksum field MUST be cleared to zero. If the computed checksum is 0, it is transmitted as all ones (the equivalent in one's complement arithmetic).

A.5 RTP Header

The first twelve octets are present in every RTP packet, while the list of CSRC identifiers is present only when inserted by a mixer. The fields have the following meaning:

- *Version* (V): 2 bits

This field identifies the version of RTP. The version defined by this specification is two (2). (The value 1 is used by the first draft version of RTP and the value 0 is used by the protocol initially implemented).

- *Padding* (P): 1 bit

If the padding bit is set, the packet contains one or more additional padding octets at the end which are not part of the payload. The last octet of the padding contains a count of how many padding octets should be ignored. Padding may be needed by some encryption algorithms with fixed block sizes or for carrying several RTP packets in a lower-layer protocol data unit.

- *Extension* (X): 1 bit

If the extension bit is set, the fixed header is followed by exactly one header extension.

- *CSRC count* (CC): 4 bits

The CSRC count contains the number of CSRC identifiers that follow the fixed header.

- Marker (M): 1 bit

The interpretation of the marker is defined by a profile. It is intended to allow significant events such as video frame boundaries to be marked in the packet stream. A profile may define additional marker bits or specify that there is no marker bit by changing the number of bits in the payload type field.

- *Payload Type* (PT): 7 bits

This field identifies the format of the RTP payload and determines its interpretation by the application. A profile specifies a default static mapping of payload type codes to payload formats. Additional payload type codes may be defined dynamically through non-RTP means.

- *Sequence Number* (SN): 16 bits

The sequence number increments by one for each RTP data packet sent, and may be used by the receiver to detect packet loss and to restore packet sequence. The initial value of the sequence number is random (unpredictable) to make known-plain text attacks on encryption more difficult, even if the source itself does not encrypt.

- *Timestamp*: 32 bits

The timestamp reflects the sampling instant of the first octet in the RTP data packet. The sampling instant must be derived from a clock that increments monotonically and linearly in time to allow synchronization and jitter calculations (see Section 6.3.1). The resolution of the clock must be sufficient for the desired synchronization accuracy and for measuring packet arrival jitter (one tick per video frame is typically not sufficient). The clock frequency is dependent on the format of data carried as payload and is specified statically in the profile or payload format specification that defines the format, or may be specified dynamically for payload formats defined through non-RTP means. If RTP packets are generated periodically, the nominal sampling instant as determined from the sampling clock is to be used, not a reading of the system clock. As an example, for fixed-rate audio the timestamp clock would likely increment by one for each sampling period. If an audio application reads blocks covering 160 sampling periods from the input device, the timestamp would be increased by 160 for each such block, regardless of whether the block is transmitted in a packet or dropped as silent.

The initial value of the timestamp is random, as for the sequence number. Several consecutive RTP packets may have equal timestamps if they are (logically) generated at once, *e.g.*, belong to the same video frame. Consecutive RTP packets may contain timestamps that are not monotonic if the data is not transmitted in the order it was sampled, as in the case of MPEG interpolated video frames. (The sequence numbers of the packets as transmitted will still be monotonic.)

- *SSRC*: 32 bits

The SSRC field identifies the synchronization source. This identifier is chosen randomly, with the intent that no two synchronization sources within the same RTP session will have the same SSRC identifier. Although the probability of multiple sources choosing the same identifier is low, all RTP implementations must be prepared to detect and resolve collisions. If a source changes its source transport address, it must also choose a new SSRC identifier to avoid being interpreted as a looped source.

- *CSRC list*: 0 to 15 items, 32 bits each

The CSRC list identifies the contributing sources for the payload contained in this packet. The number of identifiers is given by the CC field. If there are more than 15 contributing sources, only 15 may be identified. CSRC identifiers are inserted by mixers, using the SSRC identifiers of contributing sources. For example, for audio packets the SSRC identifiers of all sources that were mixed together to create a packet are listed, allowing correct talker indication at the receiver.

Bibliography

[1] H. Jenkac, T. Stockhammer, and W. Xu, "Permeable-layer receiver for reliable multicast transmission in wireless systems," in *Proc. IEEE Wireless Communications and Networking Conference*, vol. 3, pp. 1805–1811, 13-17 March 2005.

[2] L. A. Larzon, M. Degermark, L. E. Jonsson, and G. Fairhurst, "The lightweight user datagram protocol (UDP-Lite)," Tech. Rep. RFC 3828, The Internet Society, 2004.

[3] M. Larzon, L.-A.and Degermark, S. Pink, and M. Degermark, "UDP lite for real time multimedia applications," in *Proceedings of the IEEE International Conference of Communications, ICC'99*, 1999.

[4] A. Servetti, J. Carlos, and D. Martin, "802.11 MAC protocol with selective error detection for speech transmission," *Springer*, vol. 3375, pp. 509–519, 2005.

[5] M. Masala, E. Bottero and J. De Martin, "MAC-level partial checksum for H.264 video transmission over 802.11 ad hoc wireless networks," in *61st IEEE Vehicular Technology Conference, VTC*, vol. 5, pp. 2864 – 2868, 2005.

[6] E. Masala, M. Bottero, and J.C. De Martin, "Link-level partial checksum for real-time video transmission over 802.11 wireless networks," in *Proc. 14th International Packet Video Workshop (PVW)*, (Irvine, CA), December 2004.

[7] S. A. Khayam, S. Karande, H. Radha, and D. Loguinov, "Performance analysis and modeling of errors and losses over 802.11b lans for highbitrate real-time multimedia," *Signal Proc.: Image Commun.*, vol. 18, no. 7, pp. 575–595, 2003.

[8] C. Marin, Y. Leprovost, M. Kieffer, and P. Duhamel, "Robust header recovery based enhanced permeable protocol layer mechanism," in *Proc. IEEE SPAWC*, pp. 91–95, 2008.

[9] S. A. Khayam, S. S. Kar, M. U. Ilyas, and H. Radha, "Header detection to improve multimedia quality over wireless networks," *IEEE Transactions on Multimedia*, vol. 9, pp. 377–385, 2007.

[10] S. A. Khayam and H. Radha, "Maximum-likelihood header estimation: A cross-layer methodology for wireless multimedia," *IEEE Transactions on Wireless Communications*, vol. 6, no. 11, pp. 3946–3954, 2007.

[11] S. A. Khayam, S. Kar, M. U. Ilyas, and H. Radha, "Improving wireless multimedia quality using header detection with priors," in *IEEE International Conference on Communications, ICC '06.*, (Istanbul, Turkey), pp. 5457 – 5462, June 2006.

[12] R. G. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: cooperative packet recovery using physical layer information," in *Proc, ACM MobiCom*, pp. 147 – 158, 2007.

[13] M. G. Martini, M. Mazzoti, C. Lamy-Bergot, J. Huusko, and P. Amon, "Content adaptive network aware joint optimization of wireless video transmission," *IEEE Communications Magazine*, vol. 45, no. 1, 2007.

[14] G. Panza, E. Balatti, G. Vavassori, C. Lamy-Bergot, and F. Sidoti, "Supporting network transparency in 4G networks," in *Proc. IST Mobile and Wireless Communication Summit*, 2005.

[15] H. P. Sze, S. C. Liew, S. M. Ieee, J. Y. B. Lee, and D. C. S. Yip, "A multiplexing scheme for H.323 Voice-over-IP applications," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 1360–1368, 2002.

[16] P. C. Ng, S. C. Liew, and C. Lin, "Voice over wireless LAN via IEEE 802.16 wireless MAN and IEEE 802.11 wireless distribution system," in *in Proc. of IEEE WIRELESSCOM*, 2005.

[17] H. Ueda, U. Yamaguchi, and R. Watanabe, "Reducing misframe frequency for HEC-based variable length frame suitable for IP services," in *Proc. IEEE ICC 2001*, pp. 1196 – 1200, 2001.

[18] H. Ueda, U. Yamaguchi, N. Miki, and R. Watanabe, "A method of improving misframe for HEC-based variable length frame suitable for IP services," *Electronics and Communications in Japan*, vol. 86, no. 6, pp. 46–58, 2003.

[19] M. Chiani and M. G. Martini, "Optimum synchronization of frames with unknown, variable lengths on Gaussian channels," in *Proc. IEEE GLOBECOM*, pp. 4087 – 4091, Nov 2004.

[20] M. Chiani and M. Martini, "Practical frame synchronization for data with unknown distribution on AWGN channels," *IEEE Communication Letter*, vol. 9, no. 5, pp. 456 – 458, 2005.

[21] M. Chiani and M. G. Martini, "On sequential frame synchronization in AWGN channels," *IEEE Trans. Comm.*, vol. 54, no. 2, pp. 339 – 348, 2006.

[22] M. G. Martini and M. Chiani, "Optimum metric for frame synchronization with Gaussian noise and unequally distributed data symbols," in *Proc. IEEE SPAWC*, (Perugia, Italy), 21-24 June 2009.

[23] M. G. Martini and C. Hewage, "Cross-layer frame synchronization for H.264 video over WiMAX," in *Proc. IEEE SPAWC*, (Marrakech, Morocco), June 2010.

[24] O. Alay, T. Korakis, Y. Wang, and S. Panwar, "An experimental study of packet loss and forward error correction in video multicast over ieee 802.11b network," in *CCNC'09: Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, (Piscataway, NJ, USA), pp. 973–977, IEEE Press, 2009.

[25] E. A. Li, "RTP payload format for generic forward error correction," December 2007.

[26] A. Nafaa, T. Taleb, and L. Murphy, "Forward error correction strategies for media streaming over wireless networks," *IEEE Communications Magazine*, vol. 46, no. 1, pp. 72–79, 2008.

[27] "Part 11. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Quality of Service (QoS) Enhancements," Nov 2005.

[28] R. Bauer and J. Hagenauer, "Symbol-by-symbol MAP decoding of variable length codes," in *Proc. 3rd ITG Conference Source and Channel Coding*, (München), pp. 111–116, 2000.

[29] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Info. Theory*, vol. 20, no. 2, pp. 284–287, 1974.

[30] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Soft-output decoding algorithms for continuous decoding of parallel concatenated convolutional codes," in *Proc. ICC*, (Dallas, TX), pp. 112 – 117, 1996.

[31] J. Gwak, S. K. Shin, and H. M. Kim, "Reduced complexity sliding window BCJR decoding algorithms for turbo codes," in *IMA - Crypto & Coding'99* (M. Walker, ed.), pp. 179–184, Springer-Verlag, 1999.

[32] ANSI/IEEE, "802.16: Standard for local and metropolitan area networks, air interface for fixed broadband wireless access systems," tech. rep., IEEE, 2004.

[33] G. Faria, J. A. Henriksson, E. Stare, and P. Talmola, "DVB-H: digital broadcast services to handheld devices," in *Proceedings of the IEEE*, vol. 94, pp. 194–209, 2006.

[34] "DVB-H - transmission system for handheld terminals," November 2004.

[35] H. Nguyen, P. Duhamel, J. Brouet, and D. Rouffet, "Robust vlc sequence decoding exploiting additional video stream properties with reduced complexity," in *Proc. IEEE International Conference on Multimedia and Expo (ICME)*, pp. 375–378, June 2004. Taipei, Taiwan.

[36] C. Bergeron and C. Lamy-Bergot, "Soft-input decoding of variable-length codes applied to the H.264 standard," in *Proc. IEEE 6th Workshop on Multimedia Signal Processing*, pp. 87–90, 29 Sept.-1 Oct. 2004.

[37] "IEEE 802.16e, IEEE standard for local and metropolitan area networks, air interface for fixed broadband wireless access systems, amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands and corrigendum 1," February 2006.

[38] M. Kopeetsky and A. Lin, "Modified cell delineation strategy in packet switched networks," in *Second International Working Conference Performance Modelling and Evaluation of Heterogeneous Networks*, July 2004.

[39] "B - ISDN user-network interface - Physical layer specification: General characteristics," 1999.

[40] M. Chiani and M. G. Martini, "Frame synchronization for unequally distributed data symbols," in *Proceedings of IEEE Globecom*, 2006.

[41] ANSI/IEEE, "802.11, part 11 : Wireless LAN medium access control (MAC) and physical layer (PHY) specifications," tech. rep., IEEE, 1999.

[42] E. Halepovic, M. Ghaderi, and C. Williamson, "Multimedia application performance on a WiMAX network," in *Proc. MMCN*, vol. 7253, (San Jose, California), p. 725309, SPIE, 2009.

[43] J. She, F. Hou, P.-H. Ho, and L.-L. Xie, "IPTV over WiMAX: Key success factors, challenges, and solutions," *IEEE Communications Magazine*, vol. Volume: 45, Issue: 8, pp. 87–93, August 2007.

[44] R. K. Kalle, D. Das, and A. Lele, "On the performance of triple play over 802.16e based networks for rural environments," in *Proceedings of IEEE Asia-Pacific Conference on Communication (APCC - 2007)*, 2007.

[45] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," in *Proc. Globecom*, (Dallas, TX), pp. 1680–1686, 1989.

[46] J. Jiang and K. R. Narayanan, "Iterative soft decoding of Reed Solomon codes," *IEEE Communication Letters*, vol. 8, pp. 244–246, 2004.

[47] G. C. Clark and J. B. Cain, *Error-Correction Coding for Digital Communications*. Perseus Publishing, 1981.

[48] M. C. Hong, H. Schwab, L. P. Kondi, and A. K. Katsaggelos, "Error concealment algorithms for compressed video," *Signal Processing: Image Communication*, vol. 14, pp. 473–492, 1999.

[49] W.-Y. KUNG, C.-S. KIM, and C.-C. J. KUO, "Spatial and temporal error concealment techniques for video transmission over noisy channels," *IEEE transactions on circuits and systems for video technology*, vol. 16, pp. 789–802, 2006.

[50] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE *Trans. on Circuits and Systems for Video Technology*, vol. 13(7), pp. 560–576, 2003.

[51] S. Wenger, "H.264/AVC over IP," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 645–656, July 2003.

[52] S. Kumar, L. Xu, M. K. Mandal, and S. Panchanathan, "Error resiliency schemes in H.264/AVC standard," *Journal of Visual Communication and Image Representation*, vol. 17, no. 2, pp. 425 – 450, 2006. Introduction: Special Issue on emerging H.264/AVC video coding standard.

[53] J. Wang, M. Venkatachalam, and Y. Fang, "System architecture and cross-layer optimization of video broadcast over WiMAX," *IEEE Journal on selected areas in Communication*, vol. 25, no. 4, pp. 712 – 721, 2007.

[54] P. Duhamel and M. Kieffer, *Joint source-channel decoding: A cross-layer perspective with applications in video broadcasting*. Academic Press, 2008.

[55] G. Sabeva, S. Ben Jamaa, M. Kieffer, and P. Duhamel, "Robust decoding of H.264 encoded video transmitted over wireless channels," in *Proceedings of MMSP*, (Victoria, Canada), pp. 9–12, 2006.

[56] C. Lee, M. Kieffer, and P. Duhamel, "Soft decoding of VLC encoded data for robust transmission of packetized video," in *Proceedings of ICASSP*, pp. 737–740, 2005.

[57] J. Hagenauer, "Source-controlled channel decoding," *IEEE trans. on Communications*, vol. 43, no. 9, pp. 2449–2457, 1995.

[58] S. Kaiser and M. Bystrom, "Soft decoding of variable-length codes," in *Proc. IEEE ICC*, vol. 3, (New Orleans), pp. 1203–1207, 2000.

[59] R. Thobanen and J. Kliewer, "Robust decoding of variable-length encoded markov sources using a three-dimensional trellis," *IEEE Communications Letters*, vol. 7, no. 7, pp. 320–322, 2003.

[60] R. Hu, X. Huang, M. Kieffer, O. Derrien, and P. Duhamel, "Robust critical data recovery for MPEG-4 AAC encoded bitstreams," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP*, 2010.

[61] F. Mériaux and M. Kieffer, "Robust IP and UDP-lite header recovery for packetized multimedia transmission," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 2010.

[62] D. Niyato and E. Hossain, "Integration of WiMAX and WiFi: Optimal pricing for bandwidth sharing," *IEEE Communications Magazine*, vol. 45, no. 5, pp. 140–147, 2007.

[63] R. Y. Kim, J. S. Kwak, and K. Etemad, "WiMAX femtocell: requirements, challenges, and solutions," *Comm. Mag.*, vol. 47, no. 9, pp. 84–91, 2009.

[64] "IEEE 802.16e standard, http://standards.ieee.org/getieee802/802.16.html."

[65] C. Eklund, R. B. Marks, S. Ponnuswamy, K. L. Stanwood, and N. van Waes, *WirelessMAN: Inside the IEEE 802.16 Standard for Wireless Metropolitan Area Networks*. New York: IEEE Press, 2006.

[66] L. Nuaymi, *WiMAX: Technology for Broadband Wireless Access*. Wiley, January 2007.

[67] D. Chen, D. Gu, and J. Zhang, "Supporting real-time traffic with QoS in IEEE 802.11e based home networks," in *In the Proc. of IEEE CCNC '04*, 2004.

[68] Y. P. Fallah, P. Nasiopoulos, and H. Alnuweiri, "Efficient transmission of H.264 video over multirate IEEE 802.11e WLANs," *EURASIP Journal on Wireless Communications and Networking*, pp. 1–14, 2008.

[69] M. Van der Schaar, Y. Andreopoulos, and Z. Hu, "Optimized scalable video streaming over IEEE 802.11a/e HCCA wireless networks under delay constraints," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 755–768, 2006.

[70] H. Schulzrinne, G. Fokus, S. Casner, R. Frederick, and V. Jacobson, "RFC1889 - RTP: A transport protocol for real-time applications," January 1996.

[71] J. Postel, "RFC 768 : User Datagram Protocol (udp)," August 1980.

[72] "RFC 791 : Internet Protocol IP," September 1981.

[73] S. Wenger, "RFC3984: RTP payload format for H.264 video," February 2005.

[74] B. Patil, F. Xia, B. Sarikaya, J. H. Choi, and S. Madanapalli, "RFC5121: Transmission of IPv6 via the IPv6 convergence sublayer over IEEE 802.16 networks," February 2008.

[75] "IEEE 802.11-2007 IEEE standard for information technology – part 11: Wireless lan medium access control (MAC) and physical layer (PHY) specifications," June 2007.

[76] P. Neves, S. Sargento, and R. L. Aguiar, "Support of real-time services over integrated 802.16 metropolitan and local area networks," in *ISCC '06: Proceedings of the 11th IEEE Symposium on Computers and Communications*, (Washington, DC, USA), pp. 15–22, IEEE Computer Society, 2006.

[77] K. Gakhar, A. Gravey, and A. Leroy, "IROISE: A new qos architecture for IEEE 802.16 and IEEE 802.11e interworking," in *in IEEE Proc. International Conf. on Broadband Networks*, pp. 607–612, 2005.

[78] R. Fantacci and D. Tarchi, "Bridging solutions for a heterogeneous WiMAX-WiFi scenario," *Journal of Communications and Networks*, vol. 8, DECEMBER 2006.

[79] P. Djukic and S. Valaee, "Towards guaranteed QoS in mesh networks: Emulating WiMAX mesh over WiFi hardware," in *ICDCSW '07: Proceedings of the 27th International Conference on Distributed Computing Systems Workshops*, (Washington, DC, USA), p. 15, IEEE Computer Society, 2007.

[80] H.-T. Lin, Y.-Y. Lin, W.-R. Chang, and R.-S. Cheng, "An integrated WiMAX/WiFi architecture with QoS consistency over broadband wireless networks," in *CCNC'09: Proceedings of the 6th IEEE Conference on Consumer Communications and Networking Conference*, (Piscataway, NJ, USA), pp. 474–480, IEEE Press, 2009.

[81] K. Ritesh Kumar, V. Sagar, S. Kumar, A. Lele, and D. Das, "A novel interface gateway architecture for seamless interoperability between 802.11e and 802.16e," in *3rd International Conference on Communication Systems Software and Middleware and Workshops*, (Bangalore, India), pp. 480 – 487, 2008.

[82] Y. Shana, S. Yib, S. Kalyanaramanc, and J. W. Woods, "Adaptive two-stage FEC scheme for scalable video transmission over wireless networks," *Signal Processing: Image Communication*, vol. 24, pp. 718 – 729, October 2009.

[83] O. Nemethova, *Error Resilient Transmission of Video Streaming over Wireless Mobile Network*. PhD thesis, Institut für Nachrichten- und Hochfrequenztechnik, 2007.

[84] L. Superiori and O. Nemethova, "Performance of a H.264/AVC error detection algorithm based on syntax analysis," *Journal of Mobile Multimedia*, pp. 49–58, 2006.

[85] A. Sidelnikov, J. Yu, and S. Choi, "Fragmentation/aggregation scheme for throughput enhancement of IEEE 802.11n WLAN," in *proc. IEEE APWCS*, 2006.

[86] A. E. Payzin, "Analysis of a digital bit synchronizer," *IEEE Trans. Commun.*, vol. 31, p. 554Ű 560, April 1983.

[87] A. Polydoros and C. Weber, "A unified approach to serial search spread-spectrum code acquisition–part i: General theory," *IEEE Transactions on Communications*, vol. 32, pp. 542–549, May 1984.

[88] R. H. Barker, *Group synchronization of binary digital systems in Communication Theory.* Butterworth, London, 1953.

[89] J. L. Massey, "Optimum frame synchronization," *IEEE Trans. on Comm.*, vol. 20, no. 4, pp. 115–119, 1972.

[90] R. A. Scholtz, "Frame synchronization techniques," *IEEE Trans. on Comm.*, vol. 28, no. 8, pp. 1204 – 1213, 1980.

[91] G. L. Lui and H. H. Tan, "Frame synchronization for Gaussian channels," *IEEE Trans. on Comm.*, vol. 35, no. 8, pp. 818–829, 1987.

[92] R. Talluri, "Error-resilient video coding in the ISO MPEG-4 standard," *IEEE Commun. Magazine*, vol. 36, no. 36, pp. 112–119, 1998.

[93] Y. Wang, G. Wen, S. Wenger, and A. Katsaggelos, "Review of error resilient techniques for video communications," *IEEE Signal Processing Magazine*, vol. 17, no. 4, pp. 61–82, 2000.

[94] ETSI, "Digital video broadcasting (DVB); framing structure, channel coding and modulation for digital terrestrial television," tech. rep., ETSI EN 300 744 v1.5.1, jun. 2004.

[95] Lorden, M. R., and L. Swanson, "Node synchronization for the viterbi decoder," *IEEE Trans. Commun.*, vol. 32, pp. 524–531, 1984 1984.

[96] P. Robertson, "Maximum likelihood frame synchronization for flat fading channels," in *Proc. IEEE Int. Conf. Commun., ICC*, pp. 1426–1430, June 1992.

[97] M. M. K. Howlader and B. D. Woerner, "Decoder-assisted frame synchronization for packet transmission," *IEEE J. Sel. Areas Commun.*, vol. 19, pp. 2331–2345, December 2001.

[98] Z. Y. Choi and Y. H. Lee, "Frame synchronization in the presence of frequency offset," *IEEE Trans. Commun.*, vol. 50, no. 7, pp. 1062–1065, 2002.

[99] D.-U. Lee, P. Kim, and W. Sung, "Robust frame synchronization for low signal-to-noise ratio channels using energy-corrected differential correlation," *EURASIP J. Wirel. Commun. Netw.*, vol. 2009, pp. 1 – 8, 2009.

[100] M. Chiani, "Noncoherent frame synchronization," *IEEE Trans. on Communications*, vol. 58, MAY 2010.

[101] "IEEE802.11n part 11: Wireless LAN Medium Access Control (MAC) and physical layer (PHY) specifications: Enhancements for higher throughput," March 2006.

[102] C. Marin, P. Duhamel, K. Bouchireb, and M. Kieffer, "Robust video decoding through simultaneous usage of residual source information and MAC layer CRC redundancy," in *Proceedings of Globecom 07*, pp. 2070–2074, 2007.

[103] J. W. Nieto and W. N. Furman, "Cyclic redundancy check (CRC) based error method and device," Aug 2007.

[104] L. Nuaymi, N. Bouida, N. Lahbil, and P. Godlewski, "Headers overhead estimation, header suppression and header compression in WiMAX," in *Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, 2007.

[105] A. Kumar, *Mobile Broadcasting with WiMAX: Principles, Technology, and Applications*. Focal Press, 2008.

[106] S. Benedetto, D. Divsalar, G. Montorsi, and F. Pollara, "Algorithm for continuous decoding of turbo codes," *Electronics Letters*, vol. 32, no. 4, pp. 314 – 315, 1996.

[107] S. Choudhury, I. Sheriff, J. D. Gibson, and E. M. Belding-Royer, "Effect of payload length variation and retransmissions on multimedia in 802.11a WLANs," in *Proceedings of the international conference on Wireless communications and mobile computing, IWCMC '06*, (New York, NY, USA), pp. 377–382, ACM, 2006.

[108] H. Wang, J. Miao, and J. M. Chang, "An enhanced link layer retransmission scheme for IEEE 802.11," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2003)*, pp. 66–71, March 2003.

[109] J. Rey, D. Leon, A. Miyazaki, V. Varsa, and R. Hakenberg, "RTP retransmission payload format," July 2006.

[110] N. F. Maxemchuk, K. Padmanabhan, and S. Lo, "A cooperative packet recovery protocol for multicast video," in *Proc. International Conference on Network Protocols*, pp. 146–3, 1997.

[111] N. Feamster and H. Balakrishnan, "Packet loss recovery for streaming video," in *Proc. of the 12th International Packet Video Workshop*, 2002.

[112] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: A transport protocol for real-time applications," January 1996.

[113] A. Chan, S.-J. Lee, X. Cheng, S. Banerjee, and P. Mohapatra, "The impact of link-layer retransmissions on video streaming in wireless mesh networks," in *Proceedings of the 4th Annual International Conference on Wireless Internet, WICON '08*, (ICST, Brussels, Belgium, Belgium), pp. 1–9, ICST, 2008.

[114] S. Han and I. Yeom, "Retransmission scheduling for multimedia delivery over wireless home networks," *IEEE Transactions on Consumer Electronics*, vol. 52, no. 3, 2006.

[115] E. Rozner, A. P. Iyer, Y. Mehta, L. Qiu, and M. Jafry, "ER: efficient retransmission scheme for wireless LANs," in *Proceedings of the ACM CoNEXT, CoNEXT '07*, (New York, NY, USA), pp. 1–12, ACM, 2007.

[116] C. Lamy-Bergot, R. Fracchia, G. Feher, G. Jeney, M. Mazzotti, J. Zhuo, G. Panza, E. Piri, T. Sutinen, J. Vehkapera, and P. Amon, "Optimisation of multimedia over wireless ip links via x-layer design: an end-to-end transmission chain simulator," in *Mobimedia '09: Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, (ICST, Brussels, Belgium, Belgium), pp. 1–5, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[117] V. Malamal Vadakital, M. Hannuksela, M. Rezaei, and M. Gabbouj, "Method for unequal error protection in DVB-H for mobile television," in *The 17th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC06)*, (Helsinki), pp. 1 – 5, Sept. 2006.

[118] H. Mohammed and N. Farber, "An inter-burst UEP-scheme for DVB-H using redundant transmission," in *IEEE International Symposium Broadband Multimedia Systems and Broadcasting*, (Las Vegas, NV), pp. 1 – 6, 2008.

[119] J. Paavola, H. Himmanen, T. Jokela, J. Poikonen, and V. Ipatov, "The performance analysis of MPE-FEC decoding methods at the DVB-H link layer for efficient IP packet retrieval," *IEEE Transactions on Broadcasting*, vol. 53, no. 1, pp. 263–275, 2007.

[120] J. Byers, M. Luby, and M. Mitzenmacher, "A digital fountain approach to reliable distribution of bulk data," *IEEE J. Select. Areas Commun.*, vol. 20, no. 8, pp. 1528–1540, 2002.

[121] E. Paolini, G. Liva, B. Matuz, and M. Chiani, "Generalized ira erasure correcting codes for hybrid iterative / maximum likelihood decoding," *IEEE Commun. Lett.*, vol. 12, pp. 450–452, Jun. 2008.

[122] B. Libaek and O. Kure, "Protecting scalable video flows from congestion loss," *Networking and Services, International conference on*, vol. 0, pp. 228–234, 2009.

[123] J.-C. Bolot and A. V. García, "Control mechanisms for packet audio in the internet," in *Proc. of the Fifteenth IEEE Annual Joint Conference on Computer Communications, INFOCOM '96*, vol. 1, pp. 232 – 239, 1996.

[124] C. Padhye, K. Christensen, W. Moreno, and K. J. Christensen, "A new adaptive FEC loss control algorithm for voice over IP applications," in *In Proceedings of IEEE International Performance, Computing and Communication Conference*, pp. 307–313, 2000.

[125] K. Park and W. Wang, "AFEC: An adaptive forward error-correction protocol for end-to-end transport of real-time traffic," in *Proc. IEEE IC3N*, pp. 196–205, 1997.

[126] K. Park and W. Wang, "QoS-sensitive transport of real-time MPEG video using adaptive forward error correction," in *Proceedings of IEEE Multimedia Systems*, pp. 426–432, 1999.

[127] Y.-W. Kwon, H. Chang, and J. Kim, "Adaptive FEC control for reliable high-speed UDP-based media transport," in *PCM (2)*, pp. 364–372, 2004.

[128] K. Park, Y. OH, K. LIM, and K.-R. CHO, "A dynamic packet recovery mechanism for realtime service in mobile computing environments," *ETRI journal*, vol. 25, pp. 356–368, 2003.

[129] J.-S. Ahn, S.-W. Hong, and J. Heidemann, "An adaptive FEC code control algorithm for mobile wireless sensor networks," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 489–499, 2005. to appear.

[130] T. Tsugawa, N. Fujita, T. Hama, H. Shimonishi, and T. Murase, "TCP-AFEC: An adaptive FEC code control for end-to-end bandwidth guarantee," in *16th International Packet Video Workshop*, November 2007.

[131] O. Nemethova and J. C. Rodriguez, "Improved detection for H.264 encoded video sequences over mobile networks," in *Proceedings of the 8th International Symposium on Communication Theory and Applications*, p. 343, 2005.

[132] O. Nemethova, G. C. Forte, and M. Rupp, "Robust error detection for H.264/AVC using relation based fragile watermarking," in *Proc. of Int. Conf. on Systems, Signals and Image Processing (IWSSIP), Budapest, Hungary*, 2006.

[133] B. Doshi, S. Dravida, E. Hernandez-Valencia, W. Matragi, M. Qureshi, P. Langner, J. Anderson, and J. Manchester, "A simple data link (SDL) protocol for next generation packet network," *IEEE Journal on Selected Areas in Communications*, vol. 18, Issue 10, pp. 1825 – 1837, Oct 2000.