

Soutenance de Thèse

On the Network Awareness of P2P-TV Applications

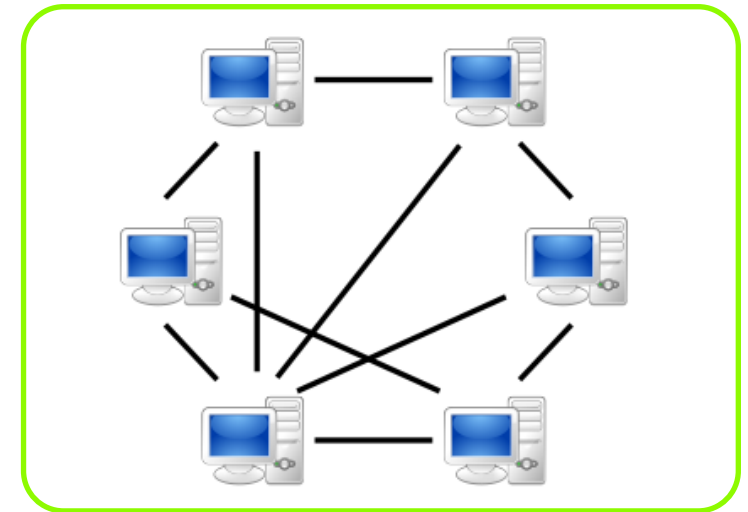
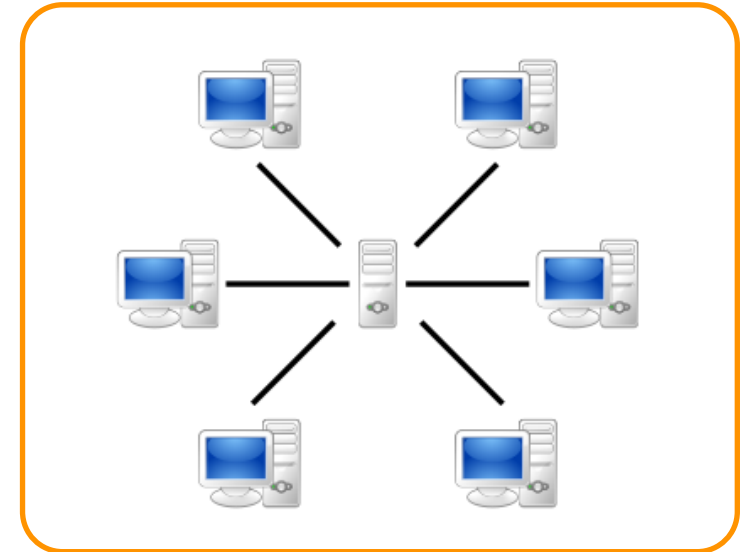
Paolo VEGLIA

Vendredi 23 septembre 2011



P2P

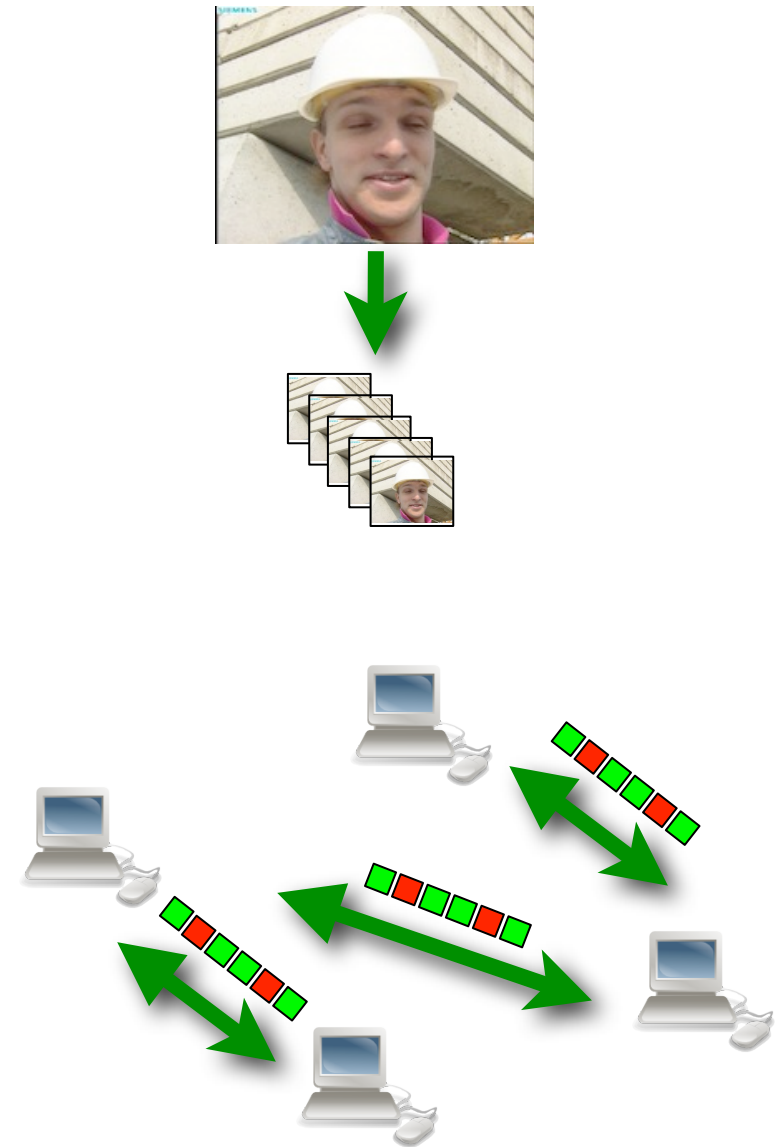
- Paradigm shift from **client-server model**
- Each node is at the same time a client and a server
- Distributed system
 - No single point of failure
- **Mesh**
 - Unstructured overlay



Font: Wikipedia

Video P2P

- Video is sliced into chunks.
- Peers store *buffer-maps* (BM).
- Peers have a neighborhood.
- Peers periodically exchange BMs with their neighbors.
- Peers need to choose a neighbor and a chunk (*peer and chunk selection*).
- **Chunks must be played before their deadline, otherwise they are lost!**



Why P2P-TV?

In 2008 P2P-TV was growing fast

- Scalable
- Low infrastructure cost
- decentralized system
- PPLive reached 104 millions users per month in 2010 [1]

In 2011 new technologies seem to take over P2P-TV (flash, OTT) [2]

- Probably due to the asymmetry of ADSL and NAT
- What happens if FTTH is deployed massively?

Operators fear P2P-TV traffic

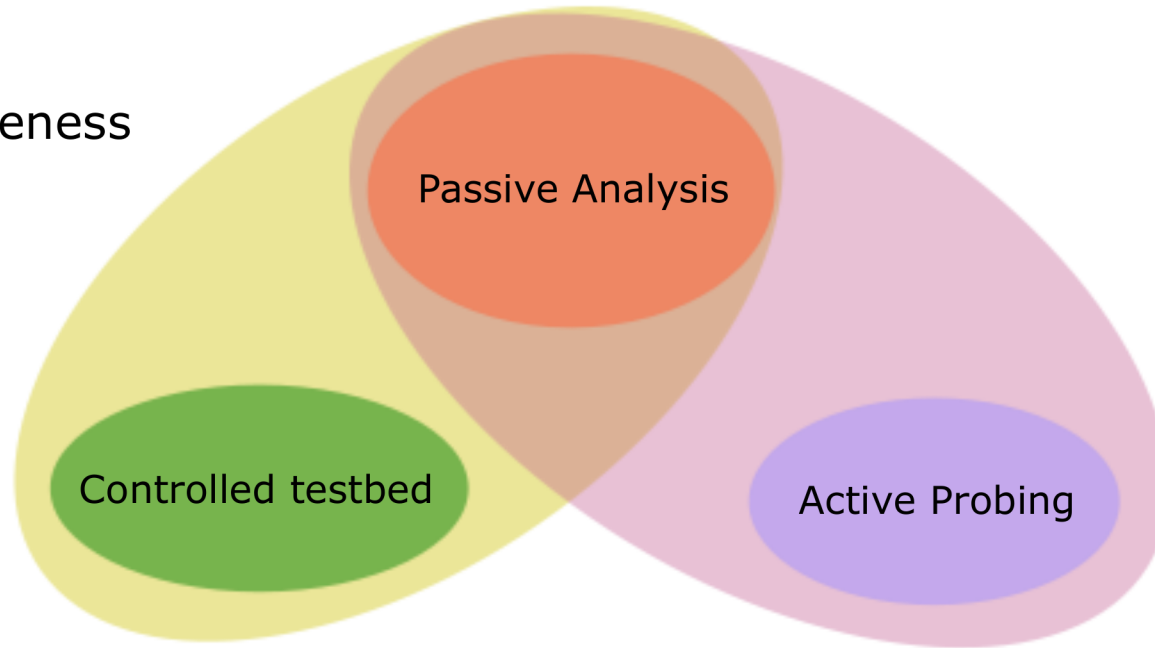
- The Comcast case

[1] <http://www.synacast.com/about/>

[2] Cisco Visual Networking Index: Forecast and Methodology, 2010-2015

Synopsis (thesis)

Measuring
Network Awareness



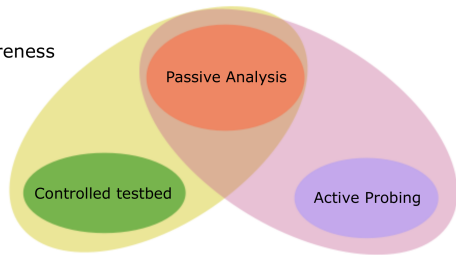
L3/L7 Interaction

Chunk-level
Simulation

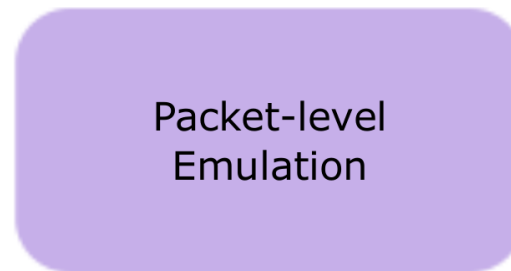
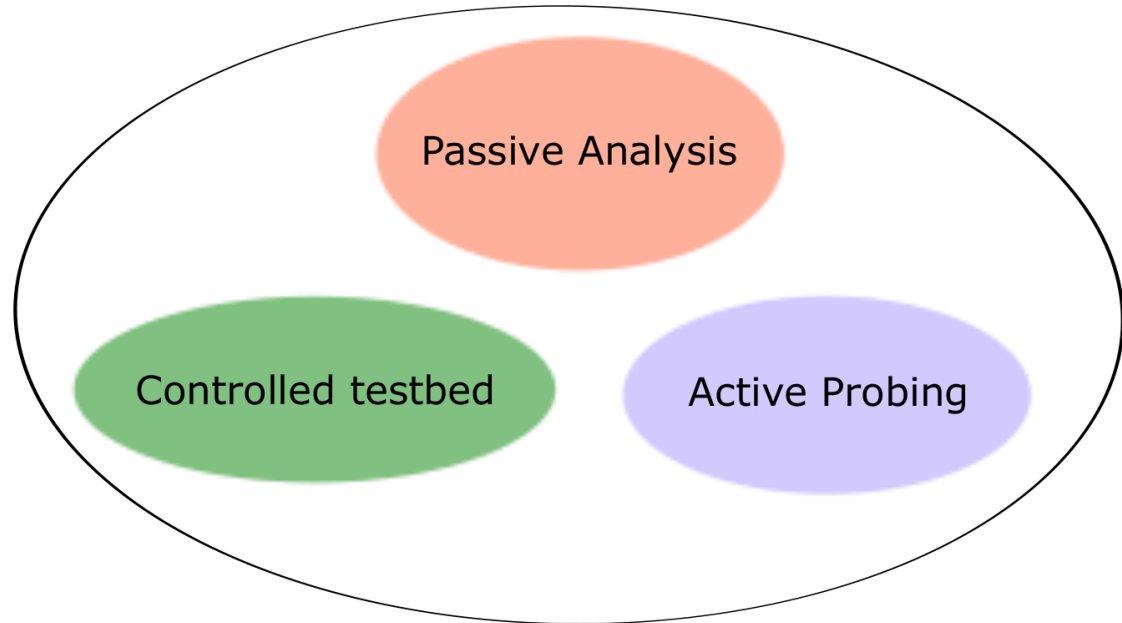
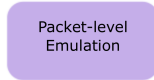
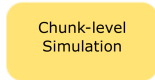
Packet-level
Emulation

Synopsis (presentation)

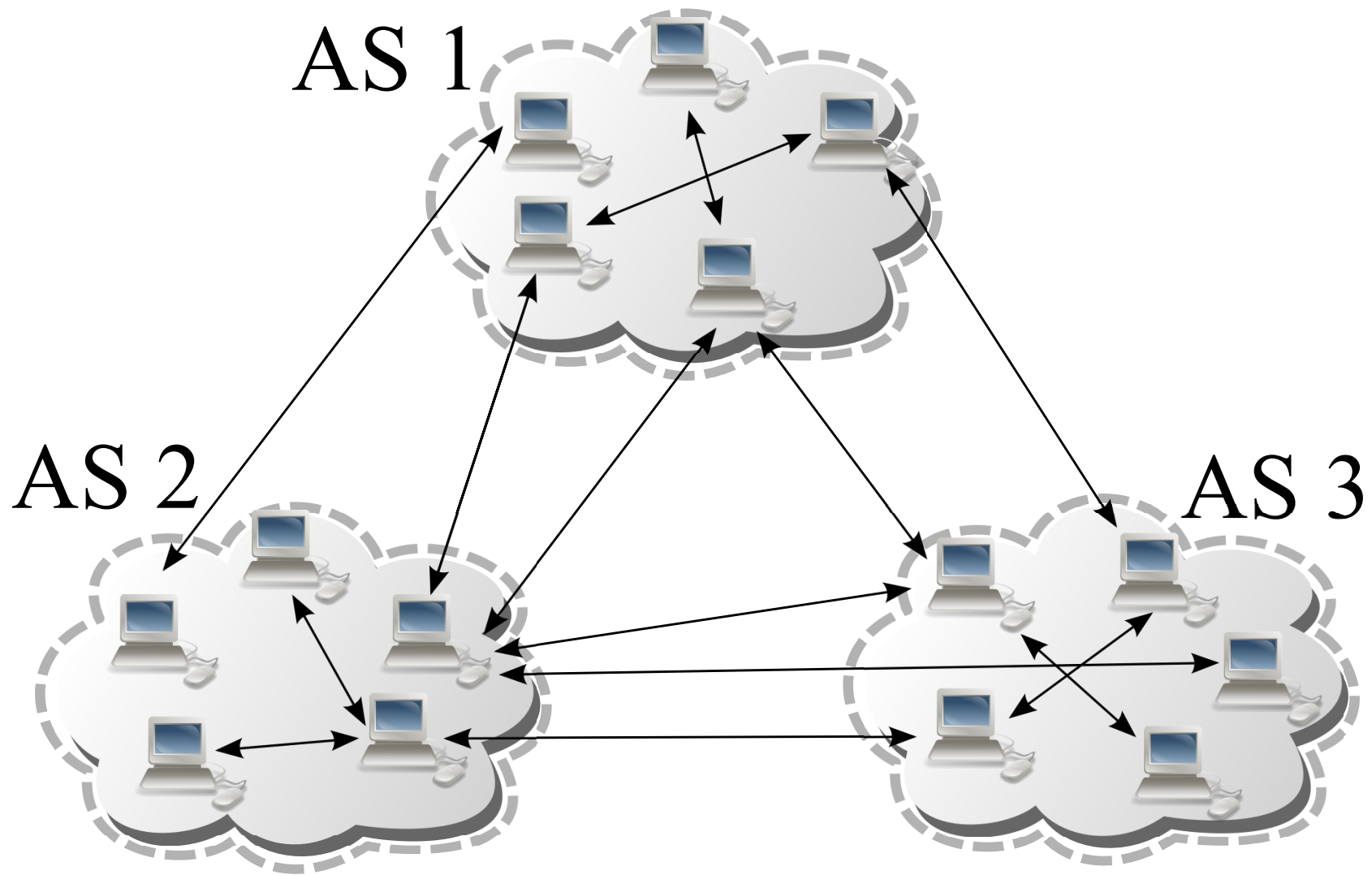
Measuring
Network Awareness



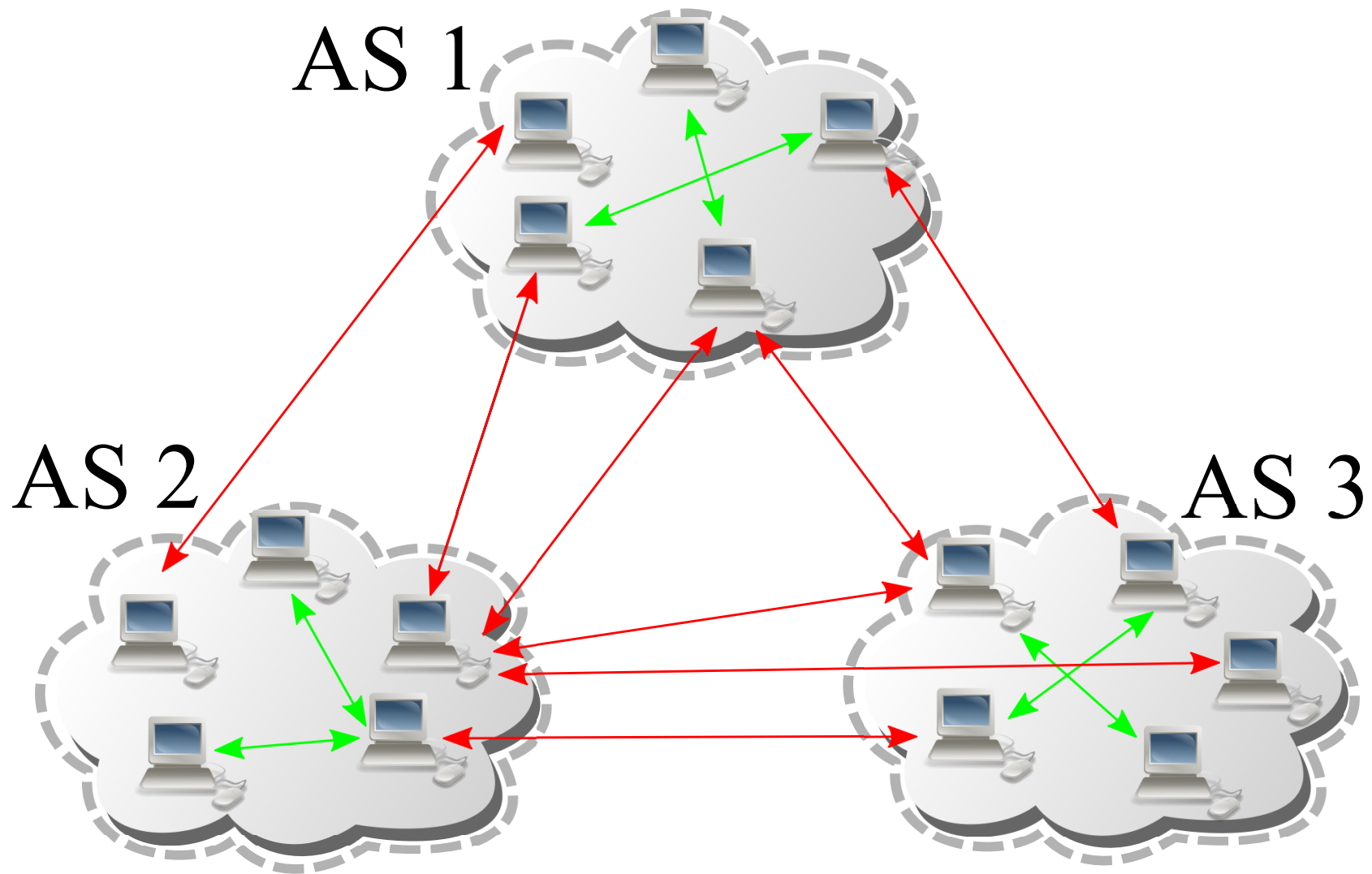
L3/L7 Interaction



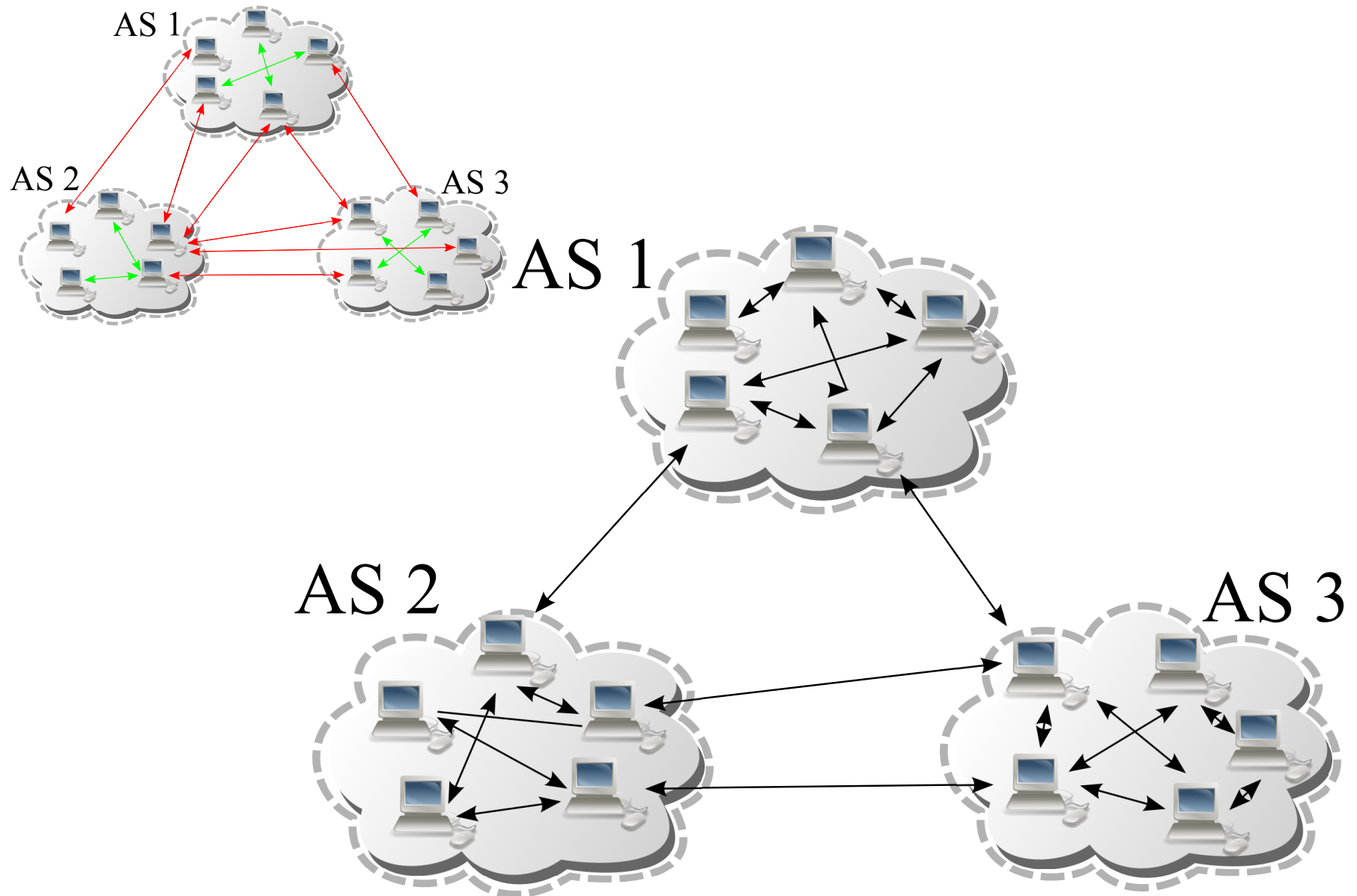
Problem: network POV



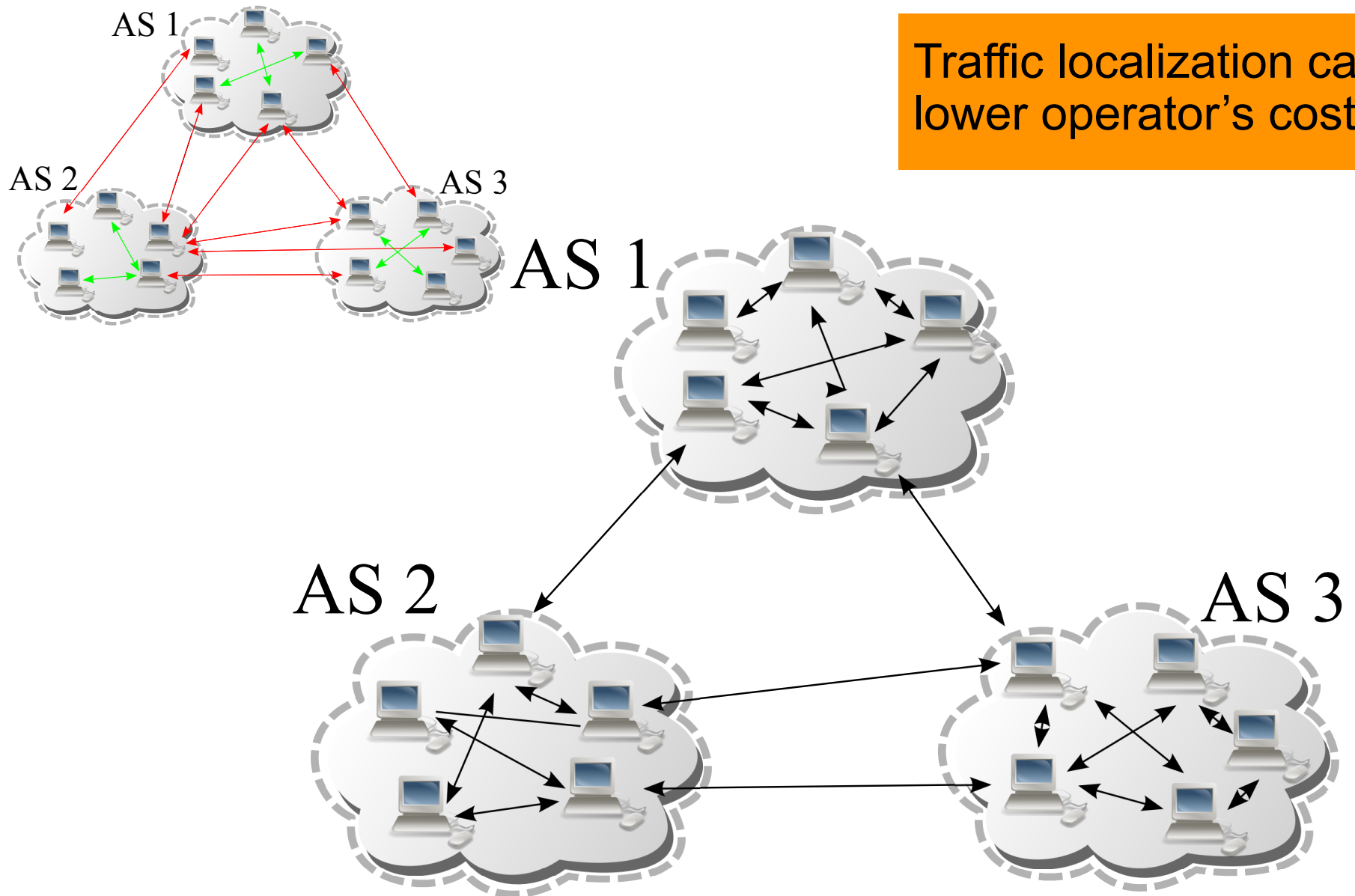
Problem: network POV



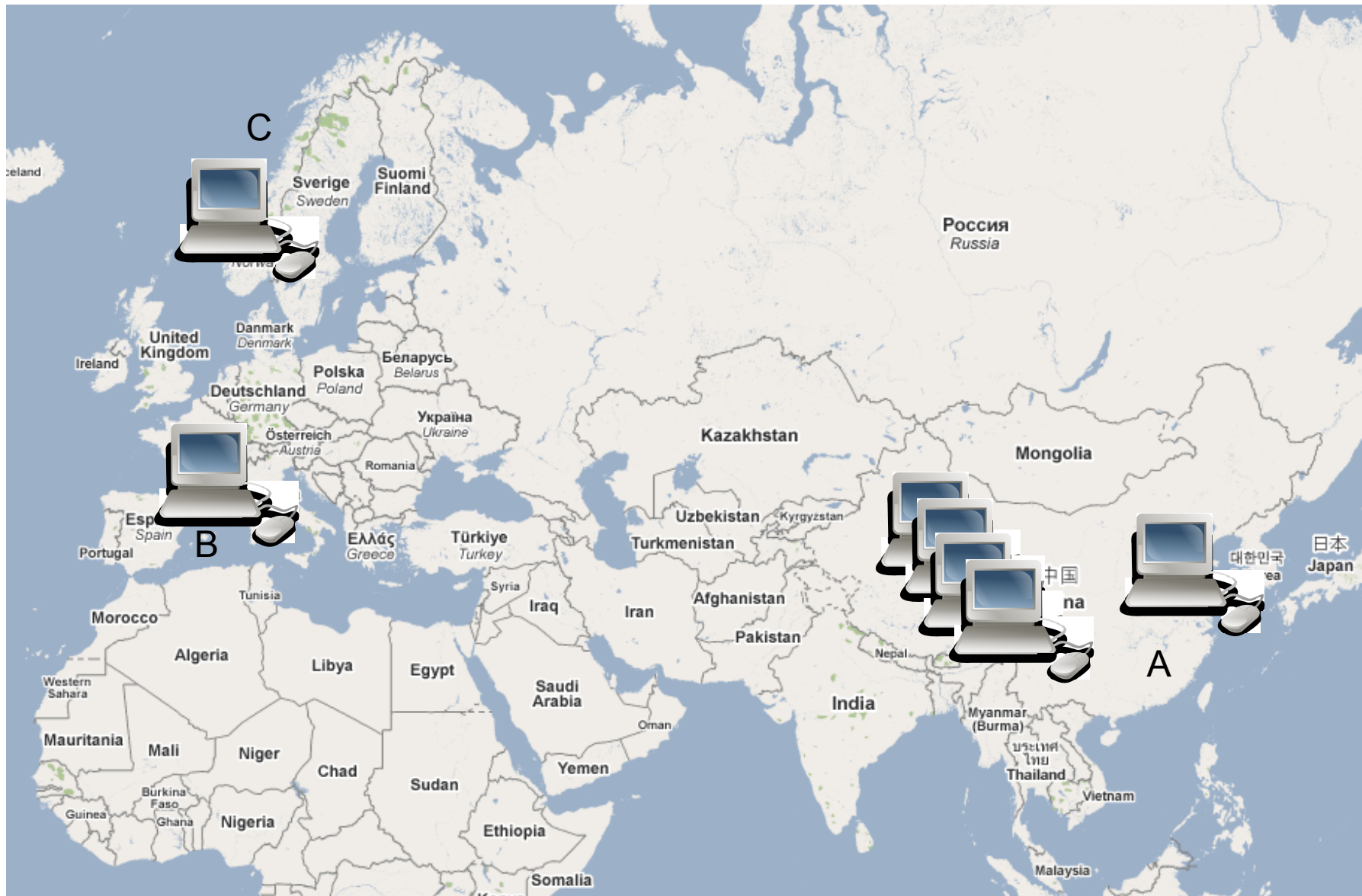
Problem: network POV (cont'd)



Problem: network POV (cont'd)



Problem: system POV



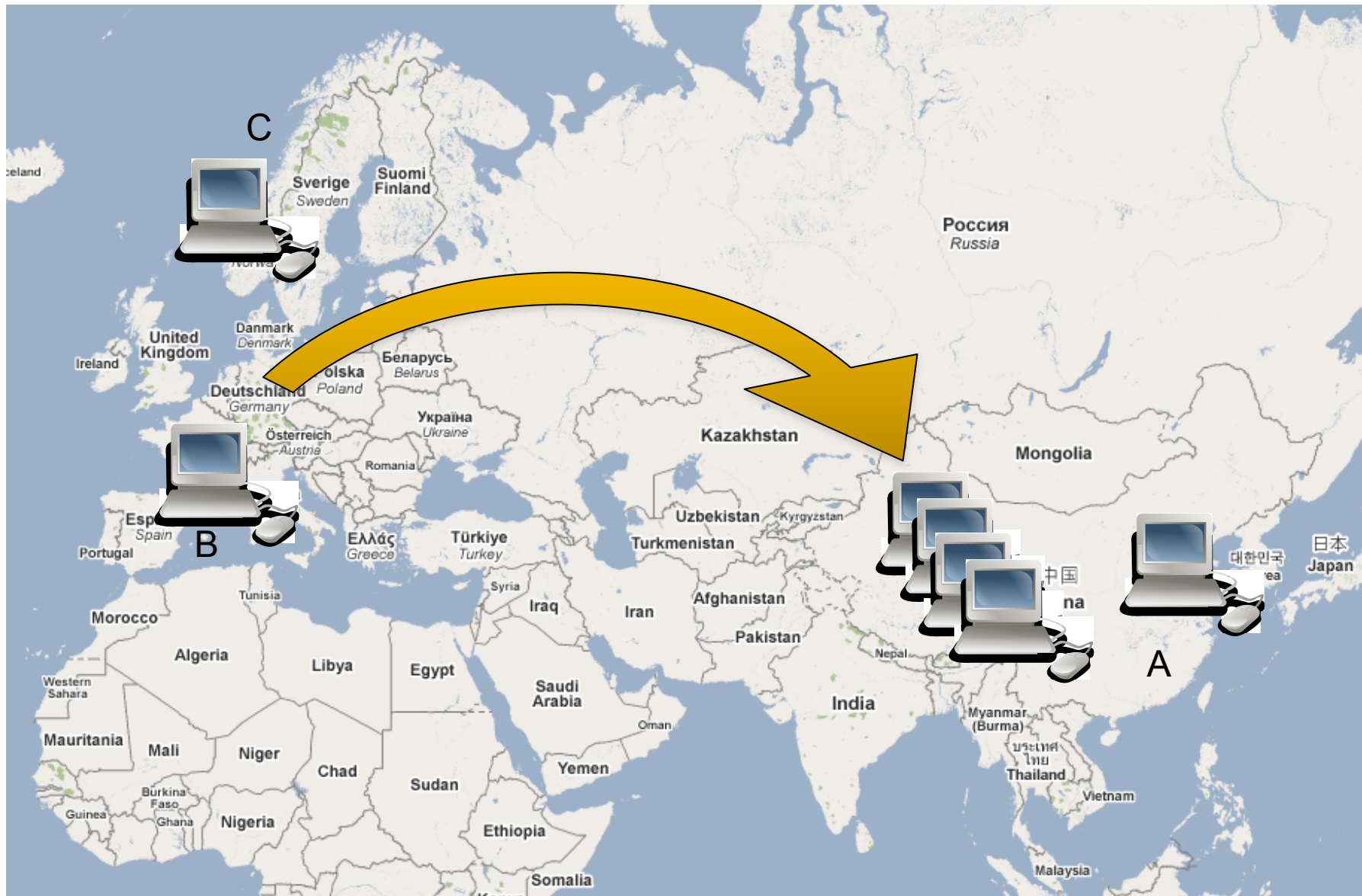
Source: <http://maps.google.com/>

Problem: system POV



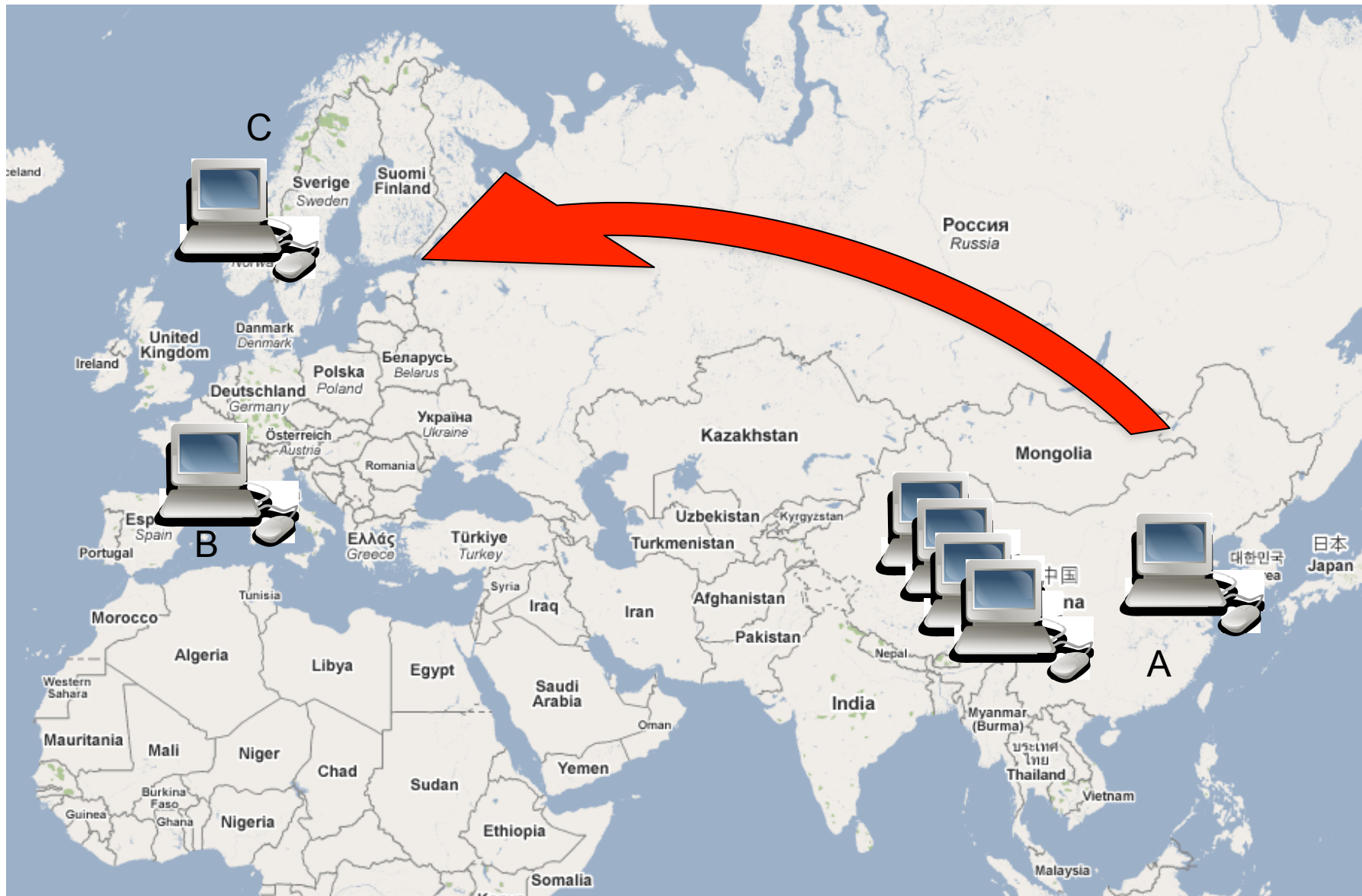
Source: <http://maps.google.com/>

Problem: system POV



Source: <http://maps.google.com/>

Problem: system POV



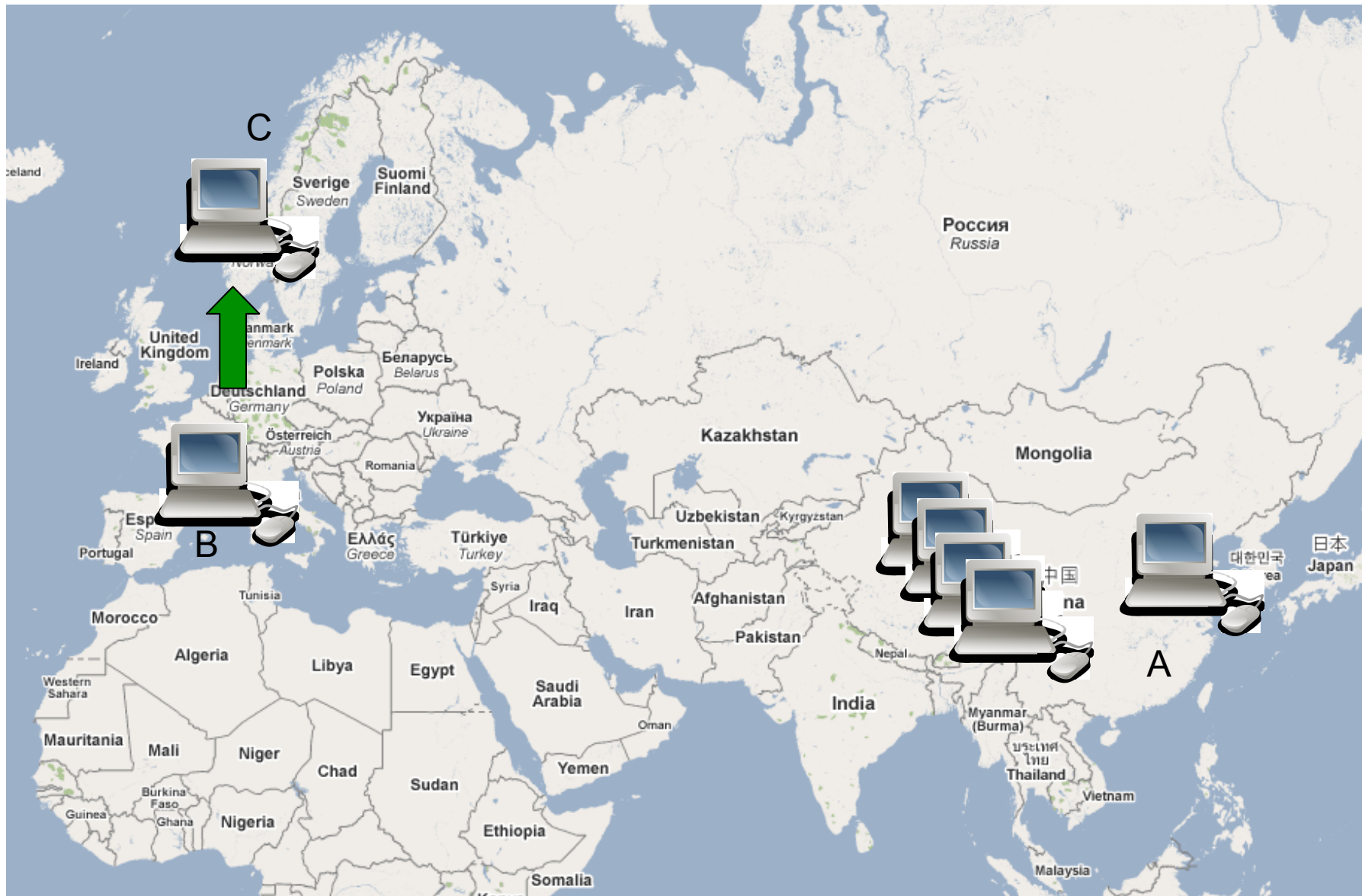
Source: <http://maps.google.com/>

Problem: system POV



Source: <http://maps.google.com/>

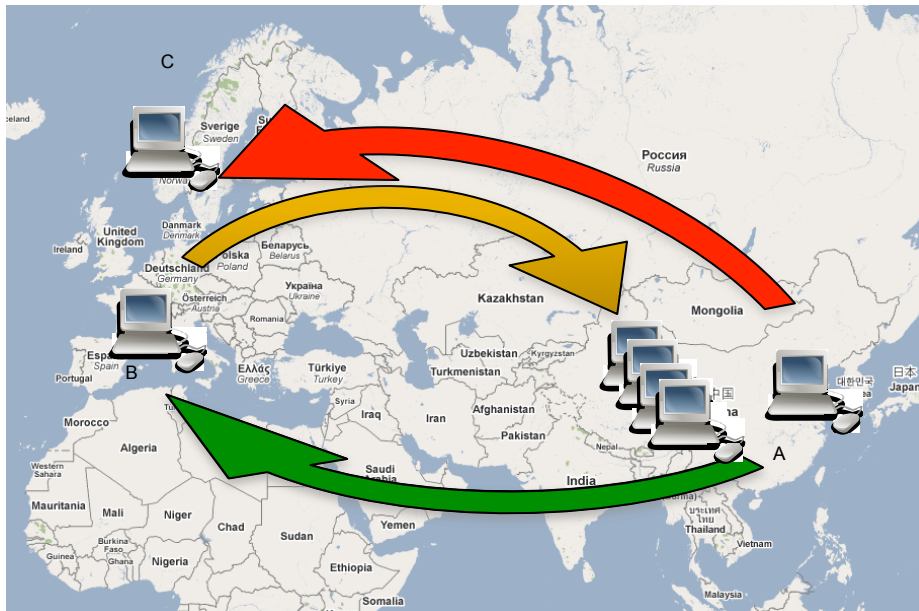
Problem: system POV



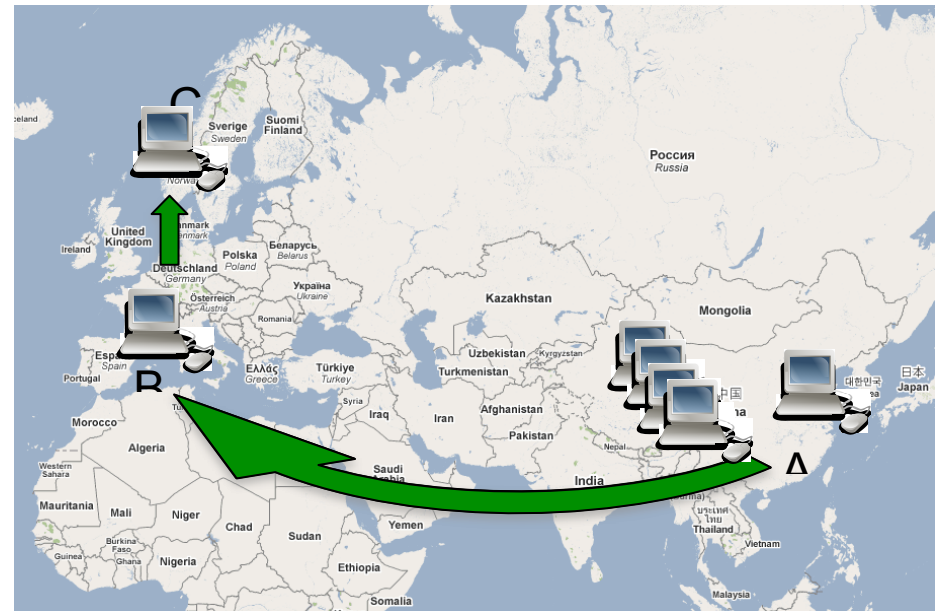
Source: <http://maps.google.com/>

Problem: system POV

Unoptimized



Optimized



Source: <http://maps.google.com/>

Solution: Network Awareness

A P2P Application is network-aware if it has some knowledge about the underlay network and uses this information for its inner algorithms.

Solution: Network Awareness

A P2P Application is network-aware if it has some knowledge about the underlay network and uses this information for its inner algorithms.

- Different methods to gather informations
 1. ISP Cooperation (“oracle” nodes) [R1,R2]
 - ☺ Operators have optimal knowledge of network topology
 - ☹ Need to define interfaces for the information exchange
 - ☹ Applications have to trust operators
 2. P2P Measurements (Latency [R3,R5], Bandwidth [R4,R5], Locality)
 - ☺ No need for external entity
 - ☹ Bias can be introduced (bandwidth measures, multiple parallel measures)

Solution: Network Awareness

A P2P Application is network-aware if it has some knowledge about the underlay network and uses this information for its inner algorithms.

- Different methods to gather informations
 1. ISP Cooperation (“oracle” nodes) [R1,R2]
 - ☺ Operators have optimal knowledge of network topology
 - ☹ Need to define interfaces for the information exchange
 - ☹ Applications have to trust operators
 2. P2P Measurements (Latency [R3,R5], Bandwidth [R4,R5], Locality)
 - ☺ No need for external entity
 - ☹ Bias can be introduced (bandwidth measures, multiple parallel measures)

Which network features must be taken into account to reach *Network Awareness*?

Features

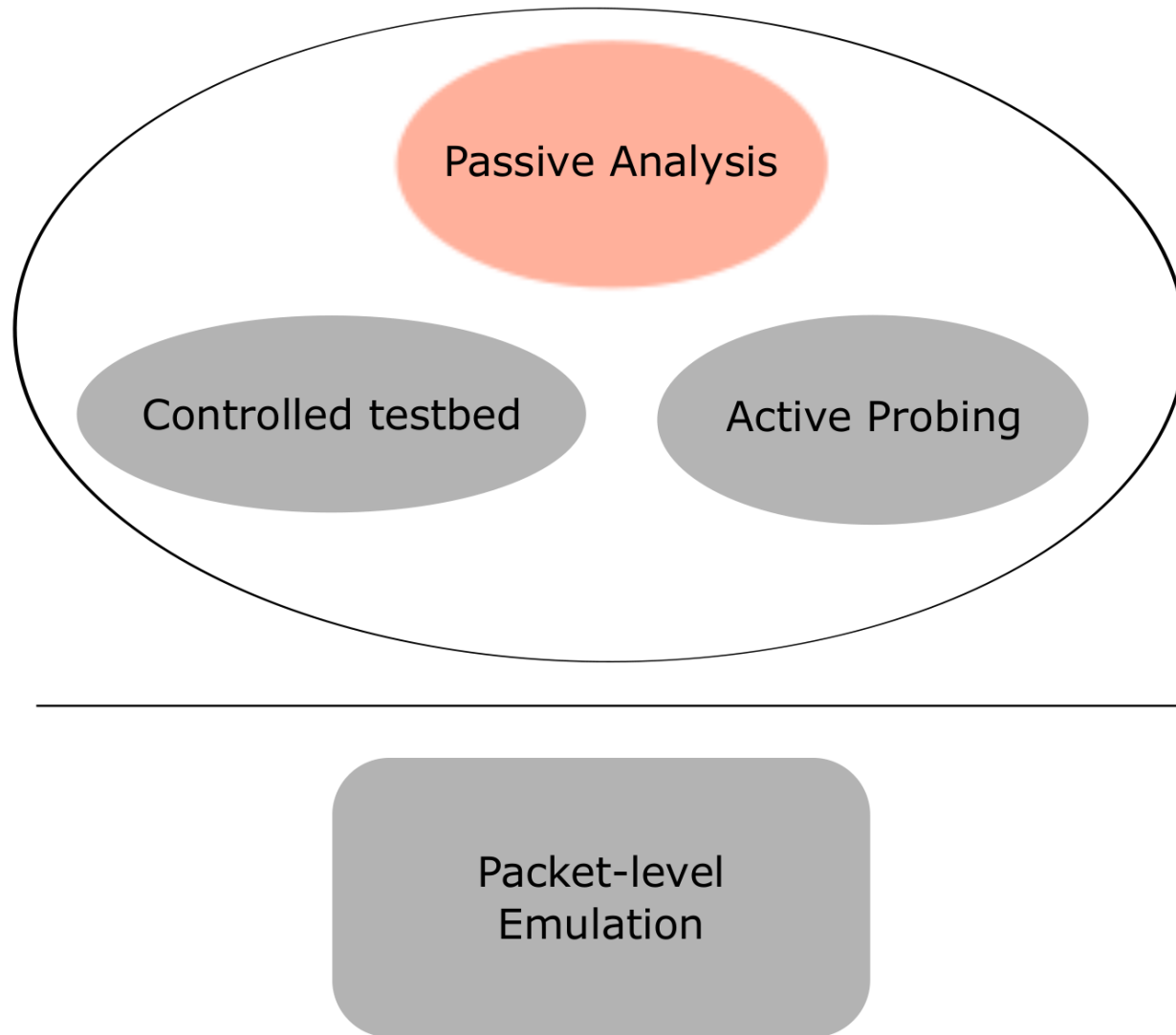
Path-Wise metrics

- Bottleneck capacity (CAP)
 - *CapProbe*
- Round Trip Time (RTT)
 - *ping, CapProbe*
- Loss rate
- Hop Count (TTL)
 - Time To Live

Peer-Wise metrics

- Autonomous System (AS)
 - Passive DB
- Geographical Location (CC)
 - Passive DB
- /16, /24 IP prefix (NET)
 - straightforward
- Access Capacity
 - application configuration

Measuring Network Awareness [2]



Multiple live measurements

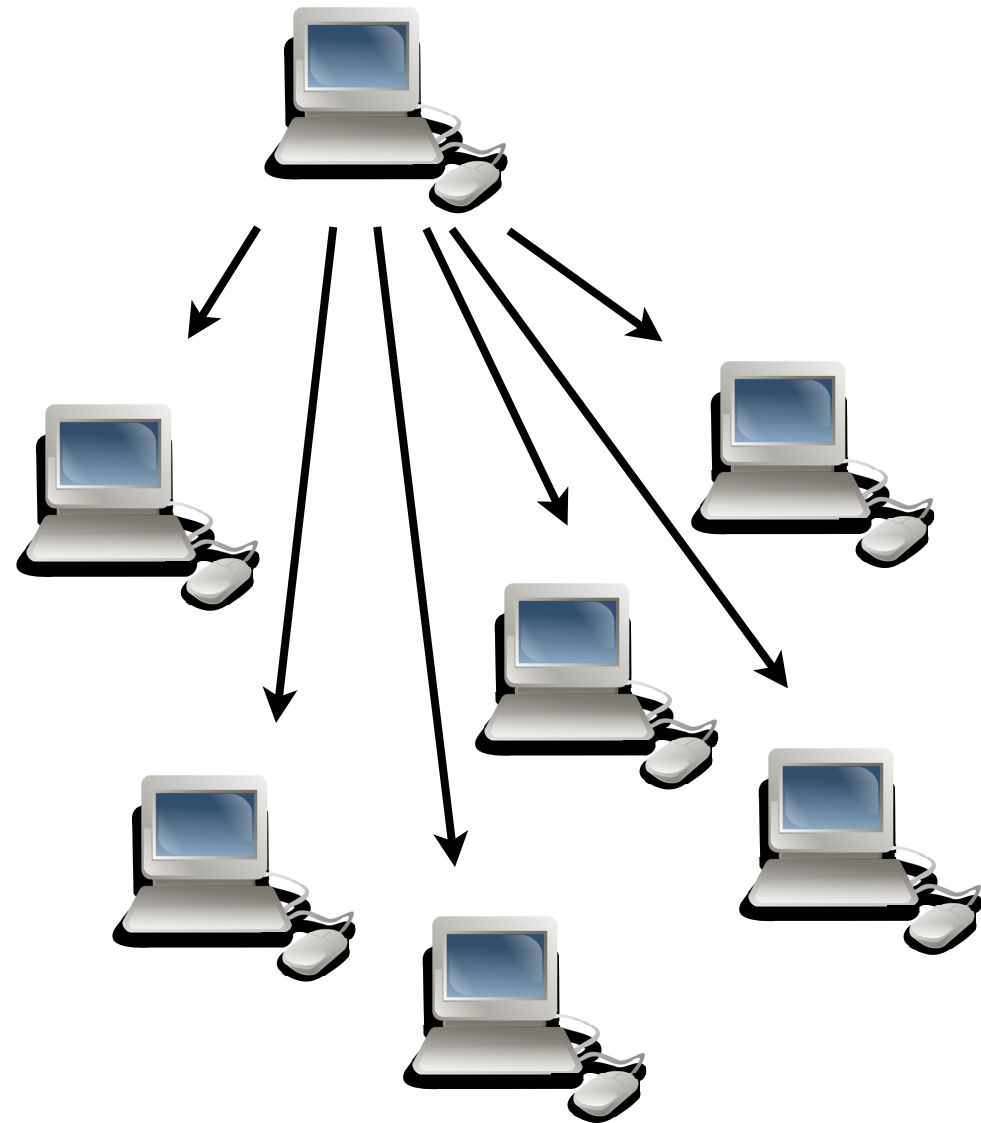


- 10 partners, 5 countries
- 40 PCs (campus LAN and residential ADSL)
- 80 hours of experiment, more than 100M of collected packets



Preferential Partition

- We split the set of contributing peers between “good” and “bad”
 - According to different metrics (AS, CC, HOP, IP)
- we count number of peers and bytes exchanged with “good” peers



$$PP_{peers} = \frac{Peers_{good}}{Peers_{good} + Peers_{bad}}$$

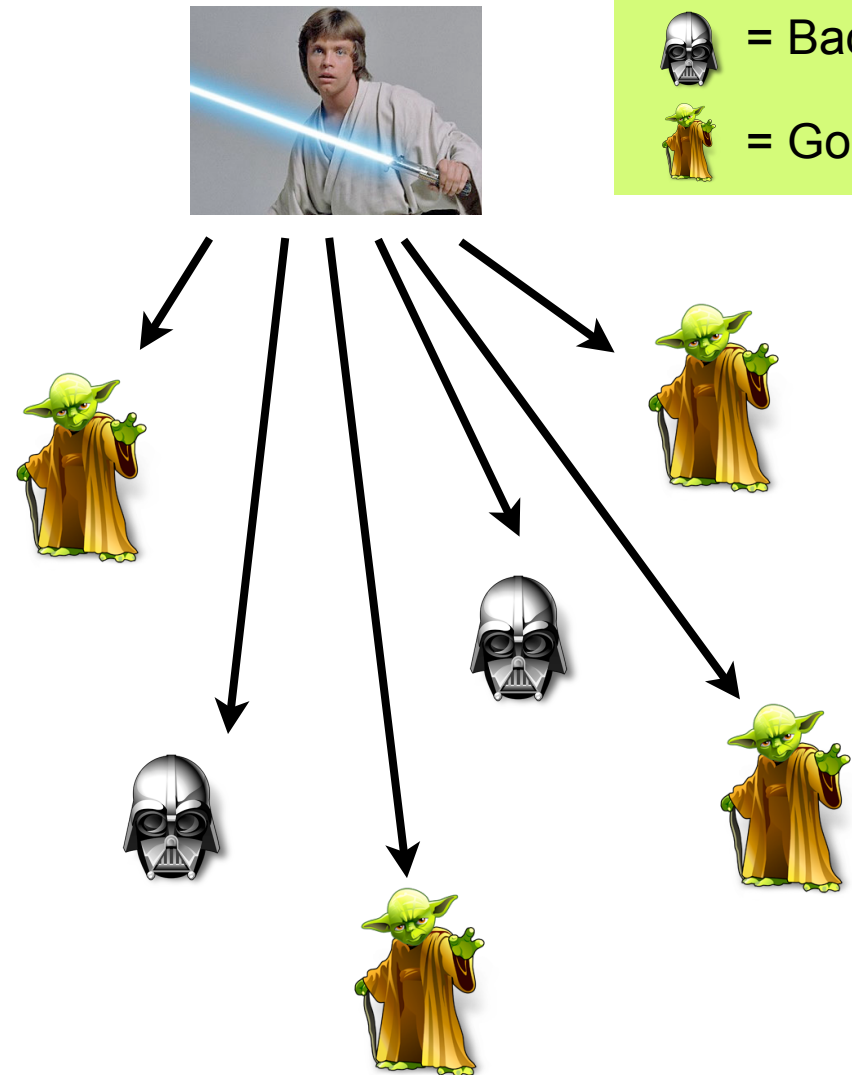
$$PP_{bytes} = \frac{Bytes_{good}}{Bytes_{good} + Bytes_{bad}}$$

Preferential Partition

- We split the set of contributing peers between “good” and “bad”
 - According to different metrics (AS, CC, HOP, IP)
- we count number of peers and bytes exchanged with “good” peers

$$PP_{peers} = \frac{Peers_{good}}{Peers_{good} + Peers_{bad}}$$

$$PP_{bytes} = \frac{Bytes_{good}}{Bytes_{good} + Bytes_{bad}}$$



* At least in episodes IV & V

Autonomous system and geo-location

	% of Contributors	% of Rx bytes from Contributors
Same AS	1,3	12,8
Same CC	1,4	13,1

- Hosts don't discover a big number of peer within the same Autonomous System, or Country but downloads a significant amount of Bytes from them.

Autonomous system and geo-location

	% of Contributors	% of Rx bytes from Contributors
Same AS	1,3	12,8
Same CC	1,4	13,1

- Hosts don't discover a big number of peer within the same Autonomous System, or Country but downloads a significant amount of Bytes from them.

PPLive prefers downloading from near peers. Is this behavior explicitly enforced in the application?

Autonomous system and geo-location

	% of Contributors	% of Rx bytes from Contributors
Same AS	1,3	12,8
Same CC	1,4	13,1

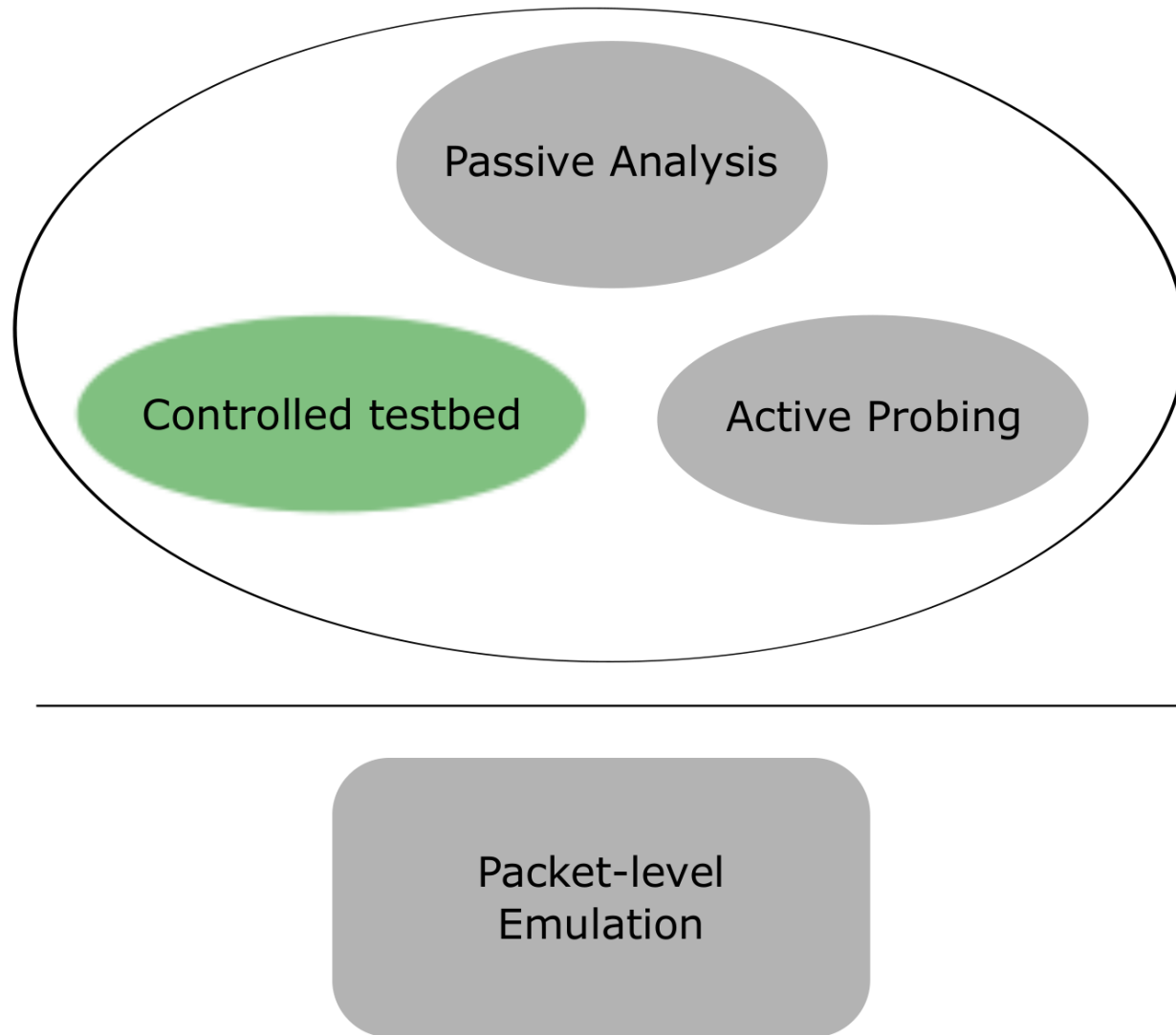
- Hosts don't discover a big number of peer within the same Autonomous System, or Country but downloads a significant amount of Bytes from them.

PPLive prefers downloading from near peers. Is this behavior explicitly enforced in the application?

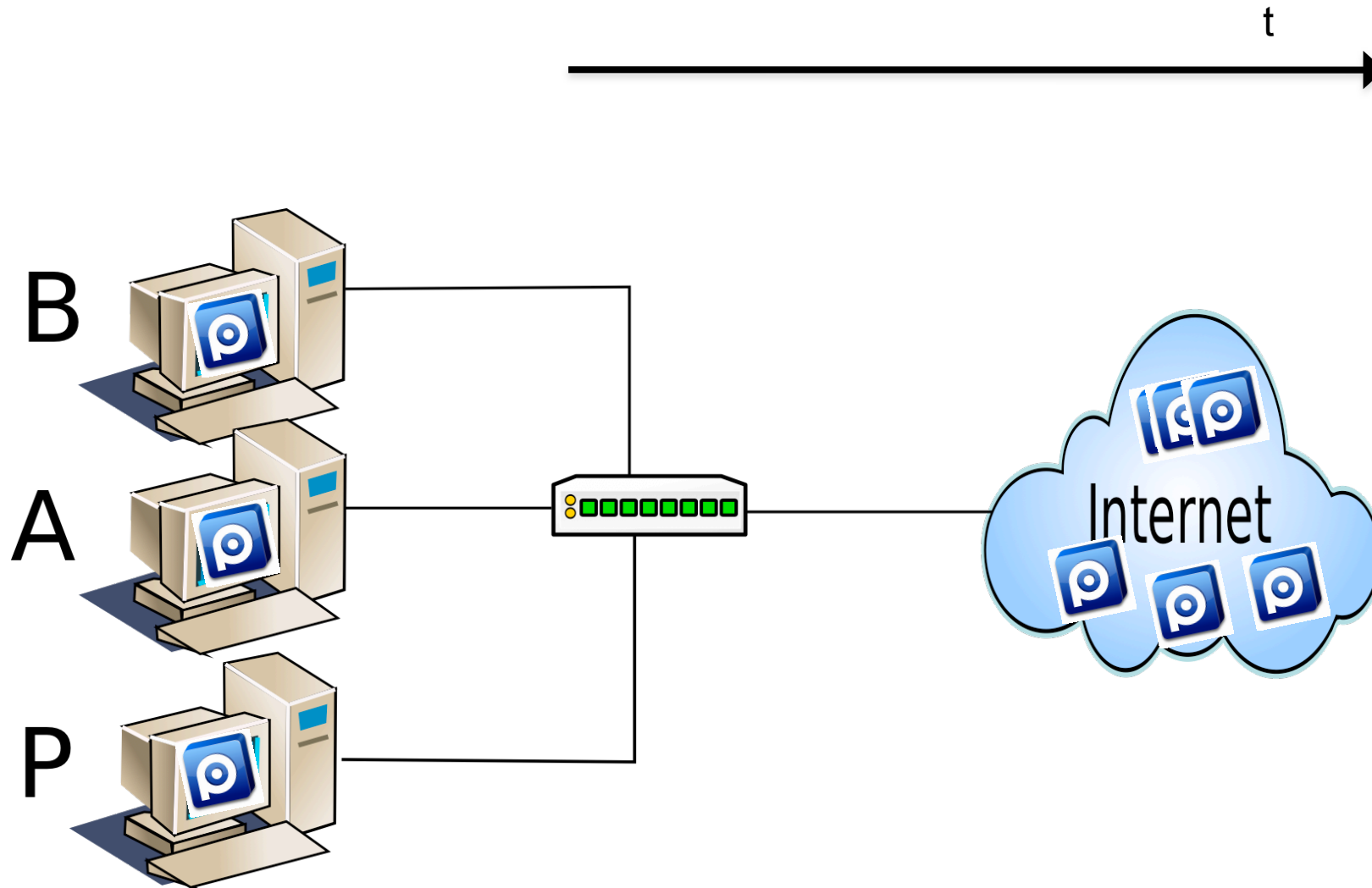
This method presents two drawbacks:

1. Path-wise metrics are not considered
2. There is an implicit bias in the dataset

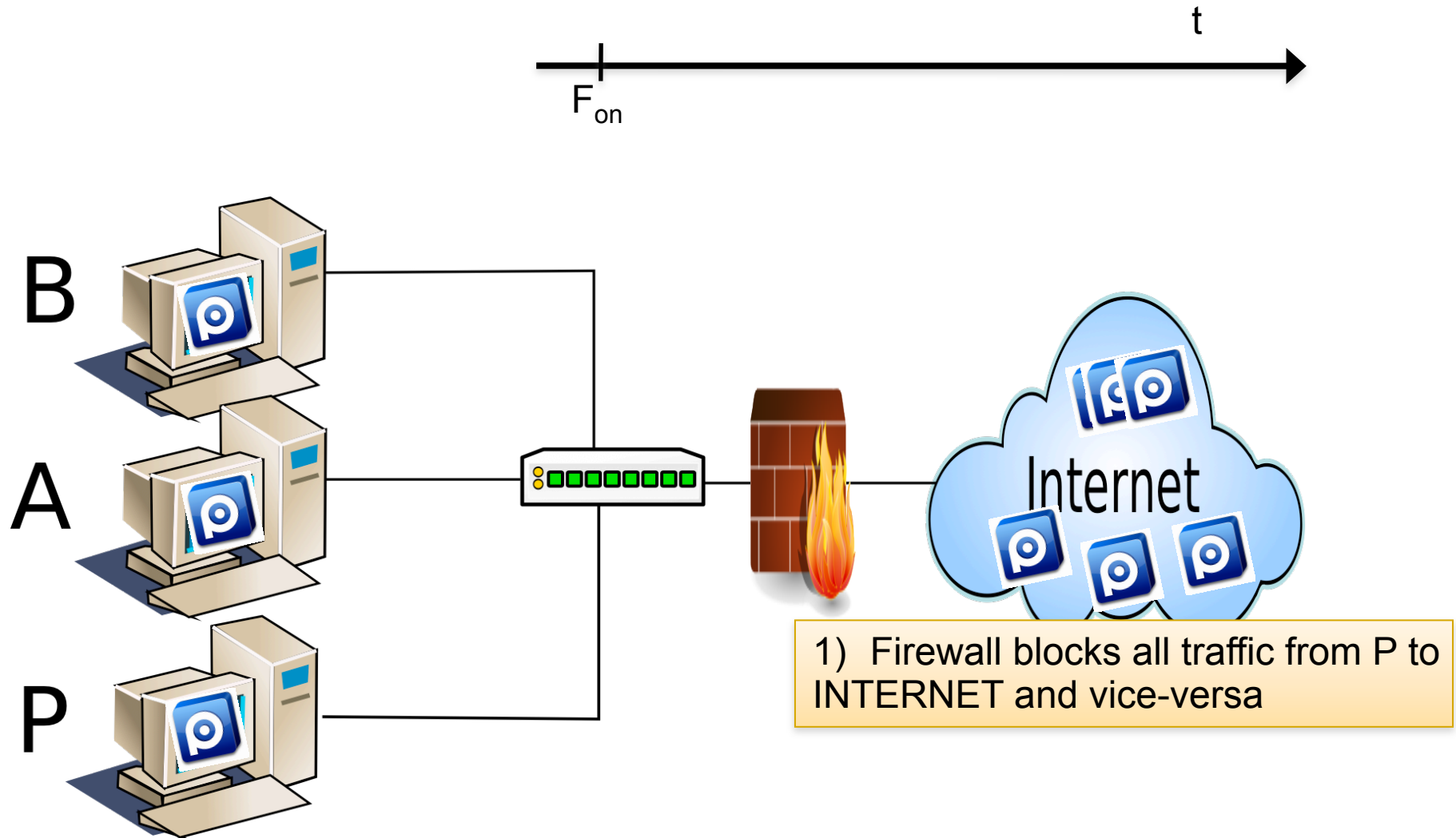
Measuring Network Awareness [4]



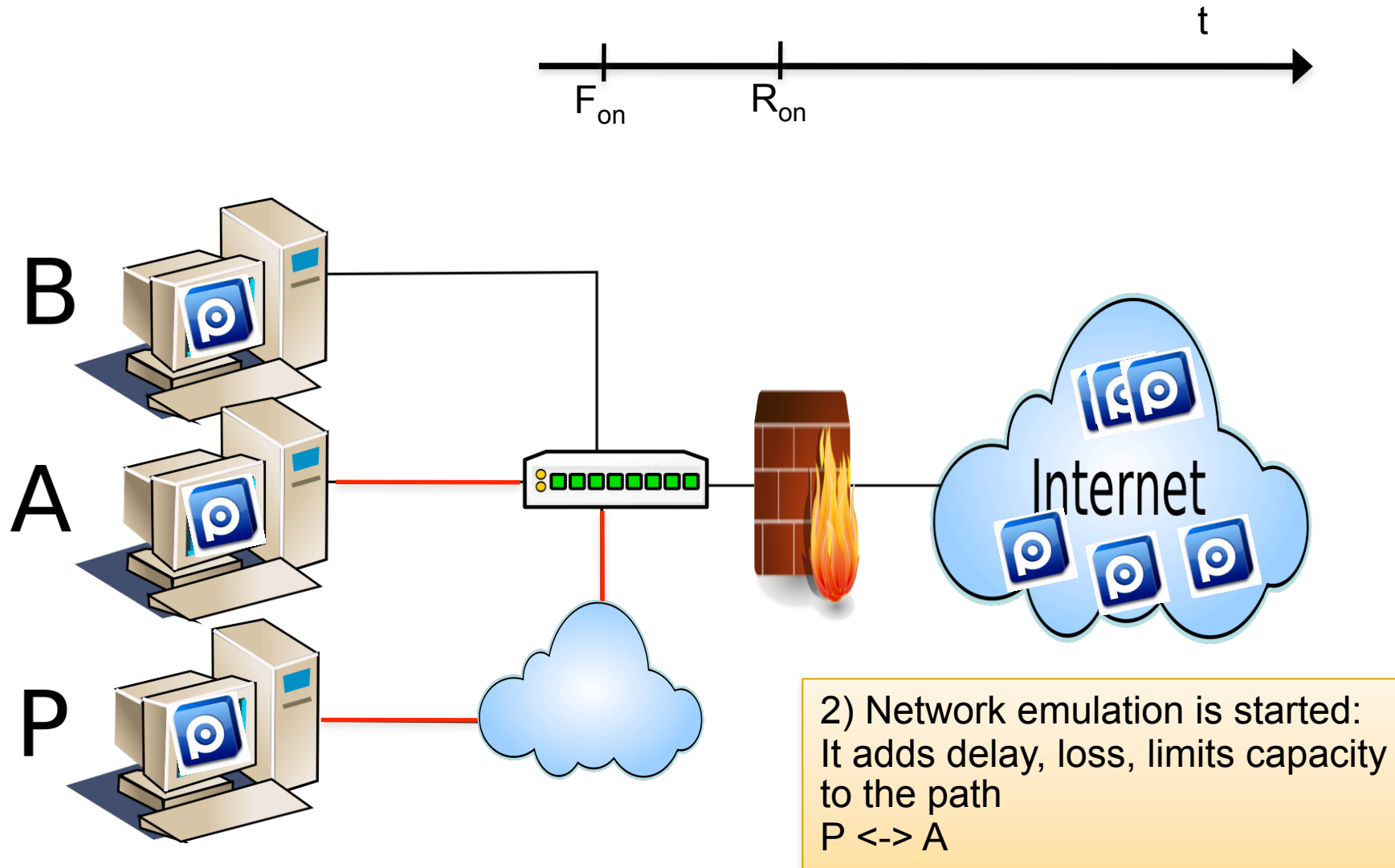
Test-bed Setup



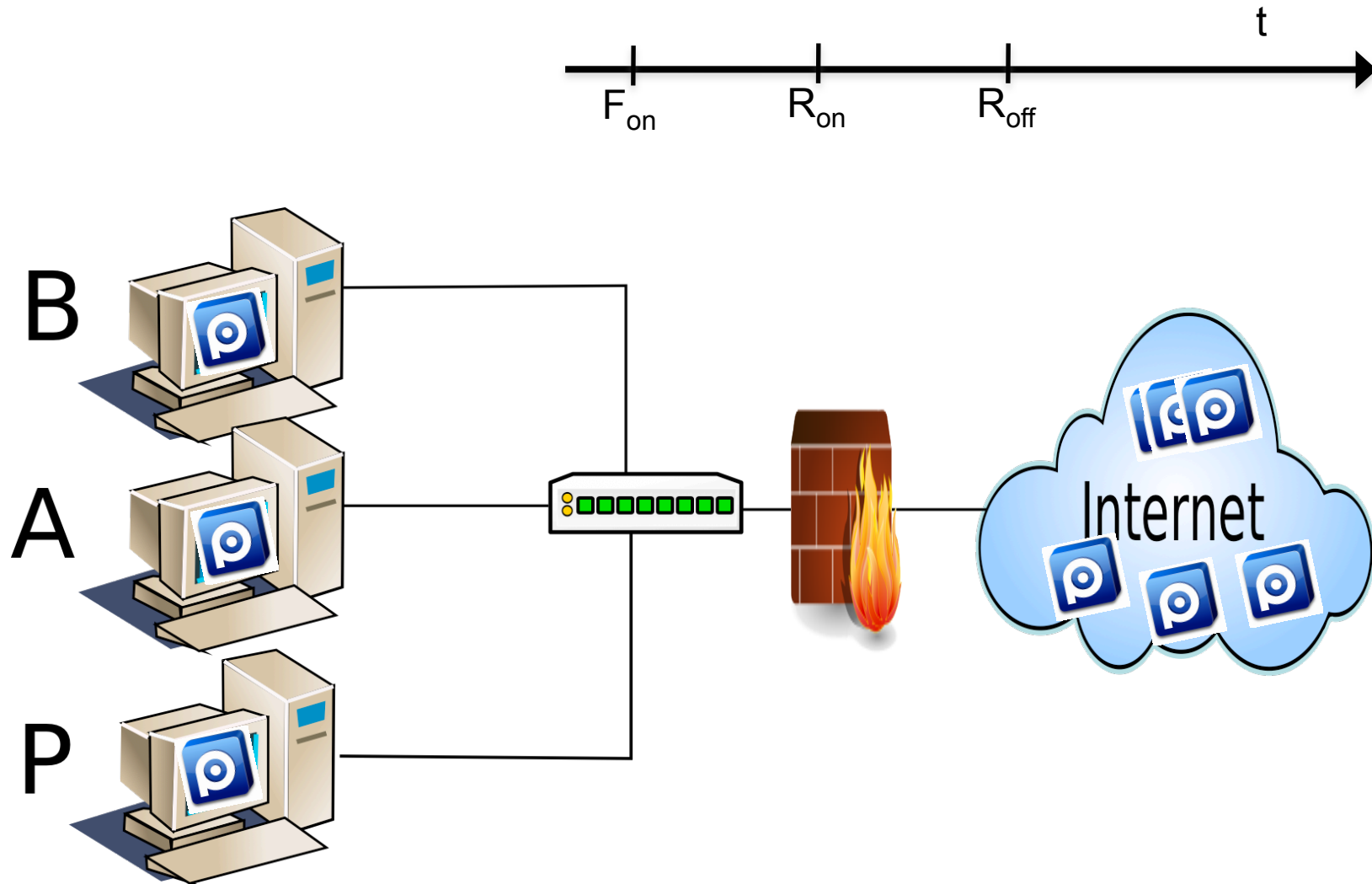
Test-bed Setup



Test-bed Setup

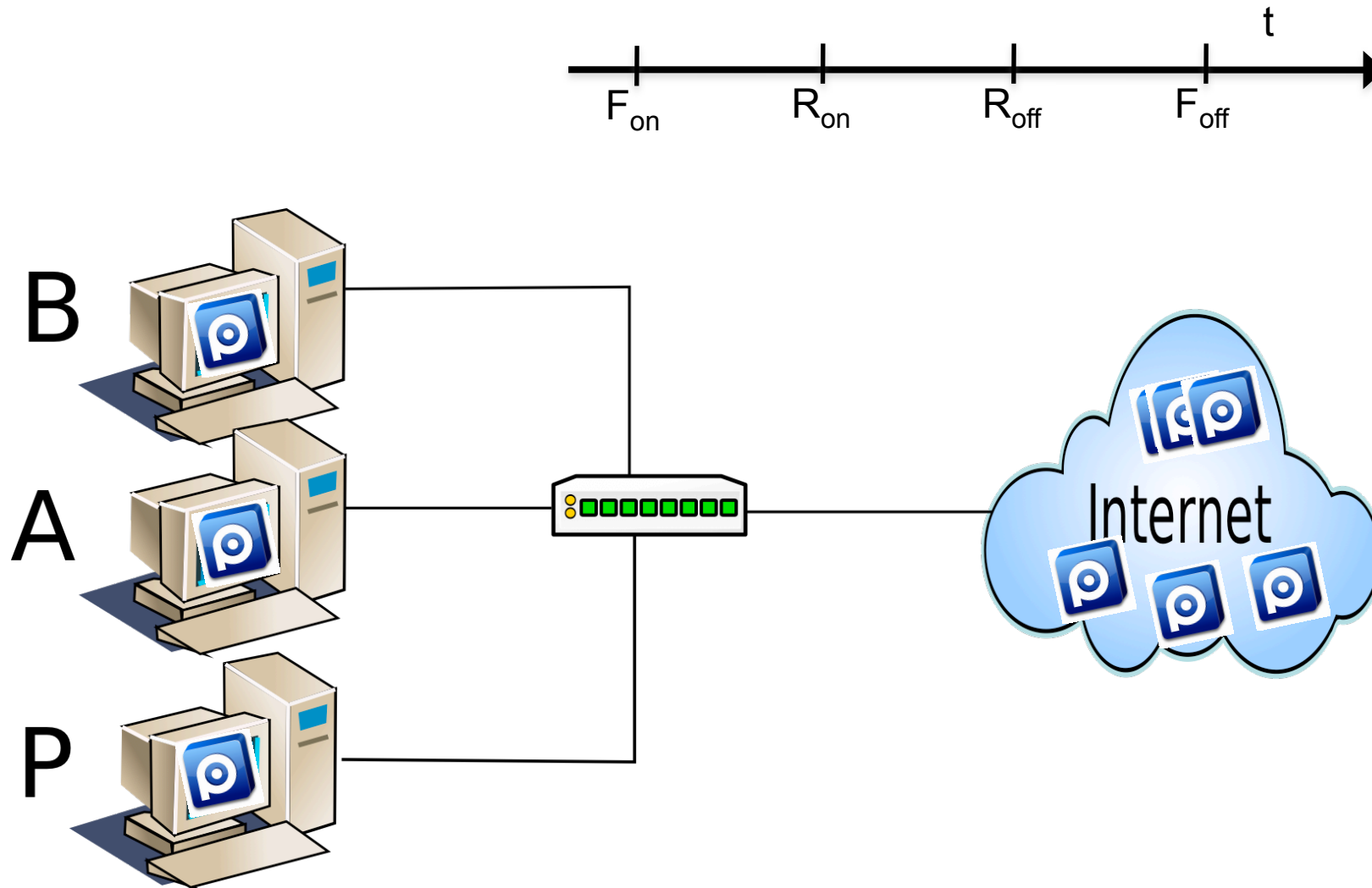


Test-bed Setup



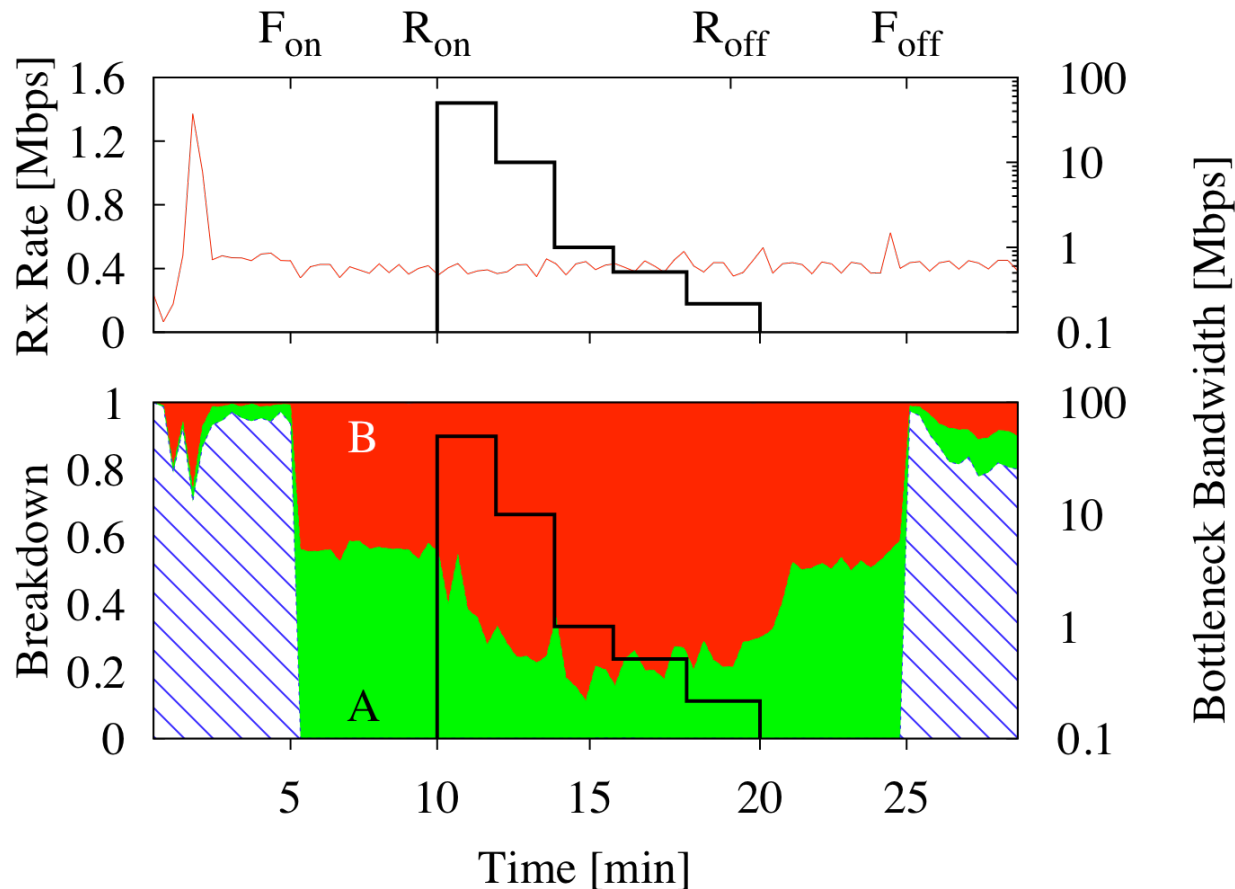
3) Network emulation is stopped

Test-bed Setup



4) Firewall allows all traffic

Capacity Limitation



- Bitrate and Breakdown of the video stream.
- Bandwidth limitation never goes under the streaming threshold
- PPLive is very reactive towards bandwidth change!

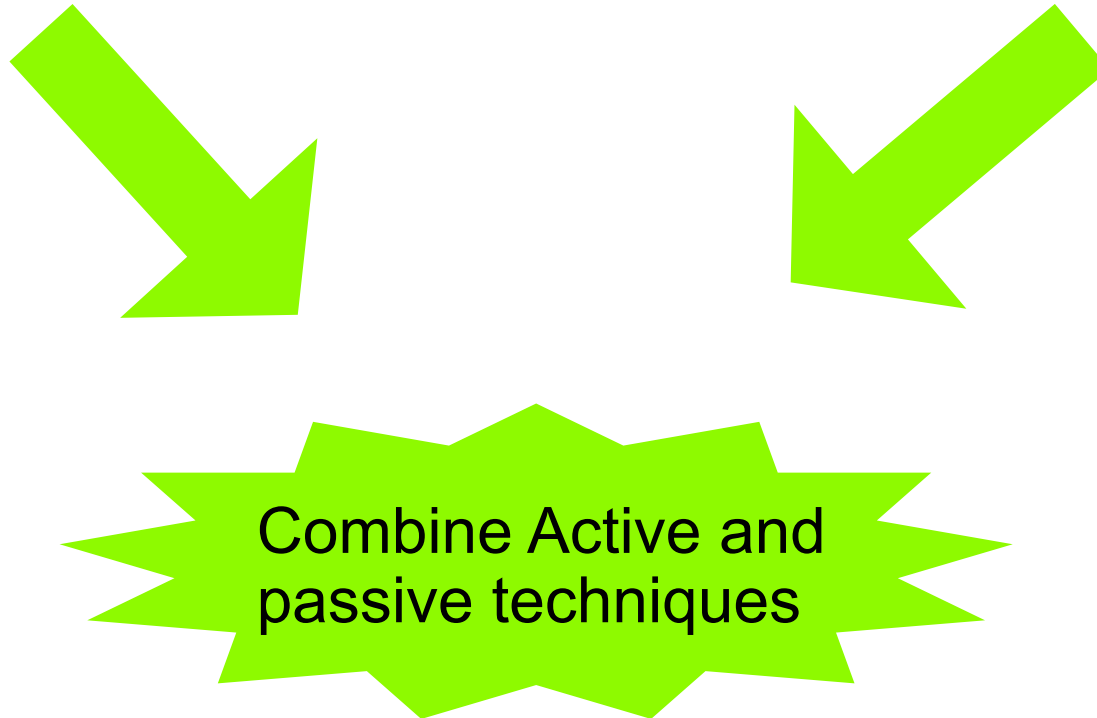
Problematics

Passive analysis

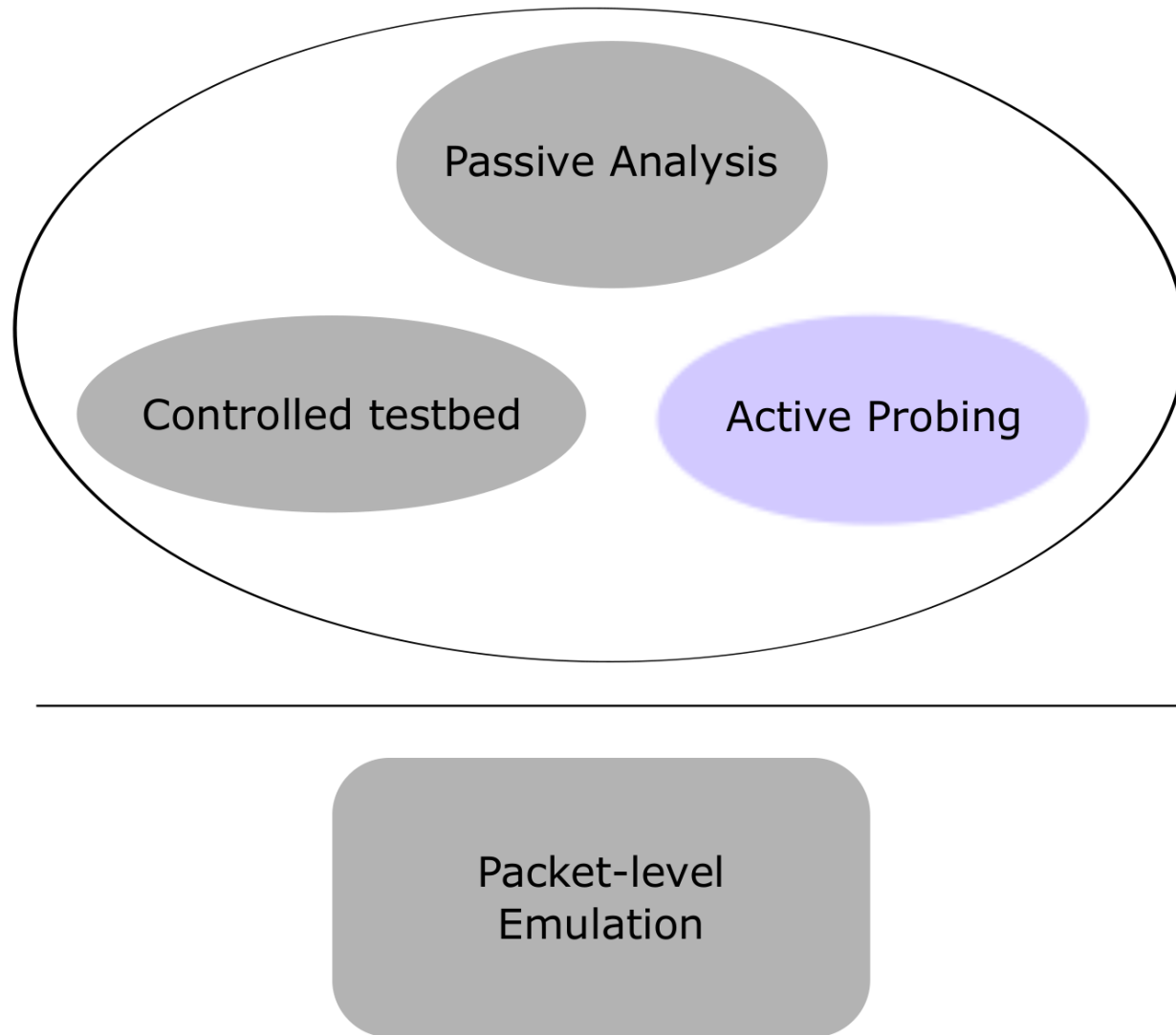
- bias
- peer-wise

Active testbed

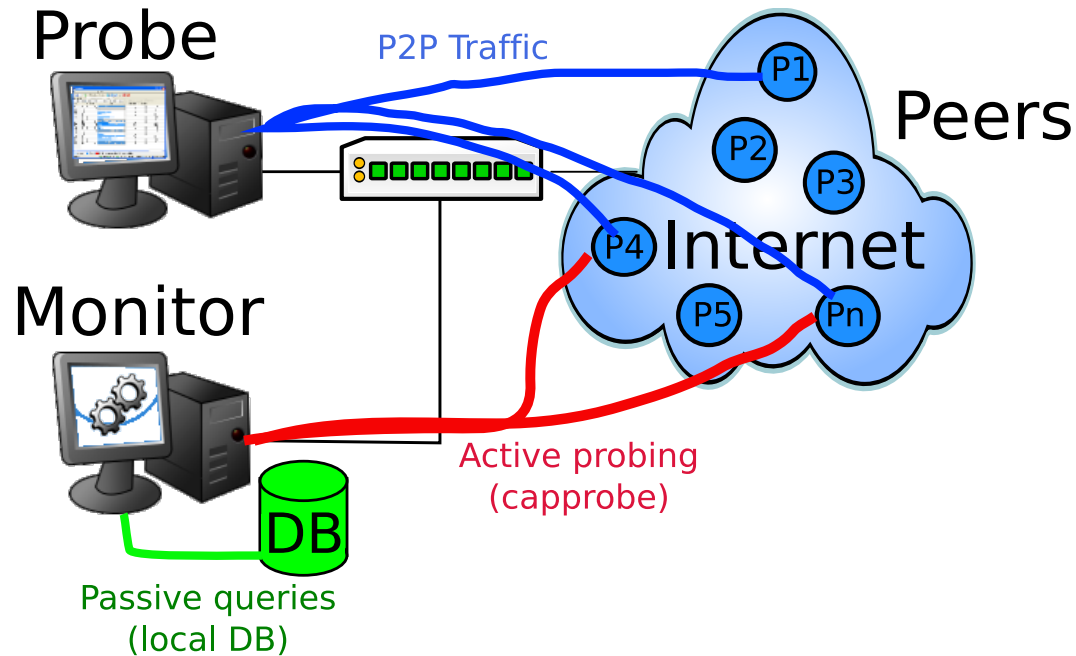
- path-wise
- testbed



Measuring Network Awareness [3]

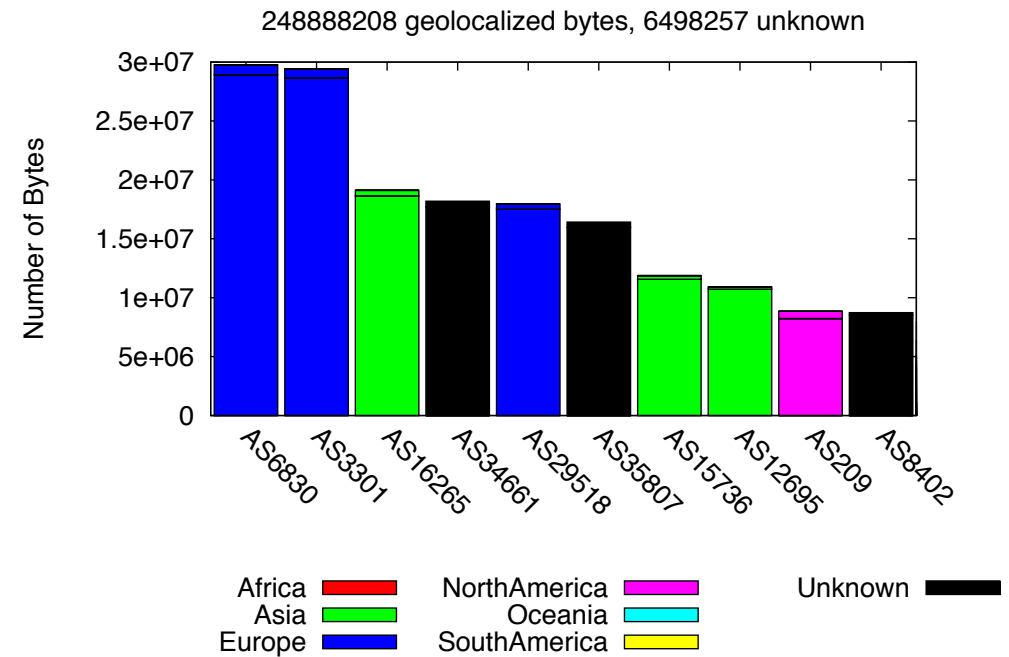
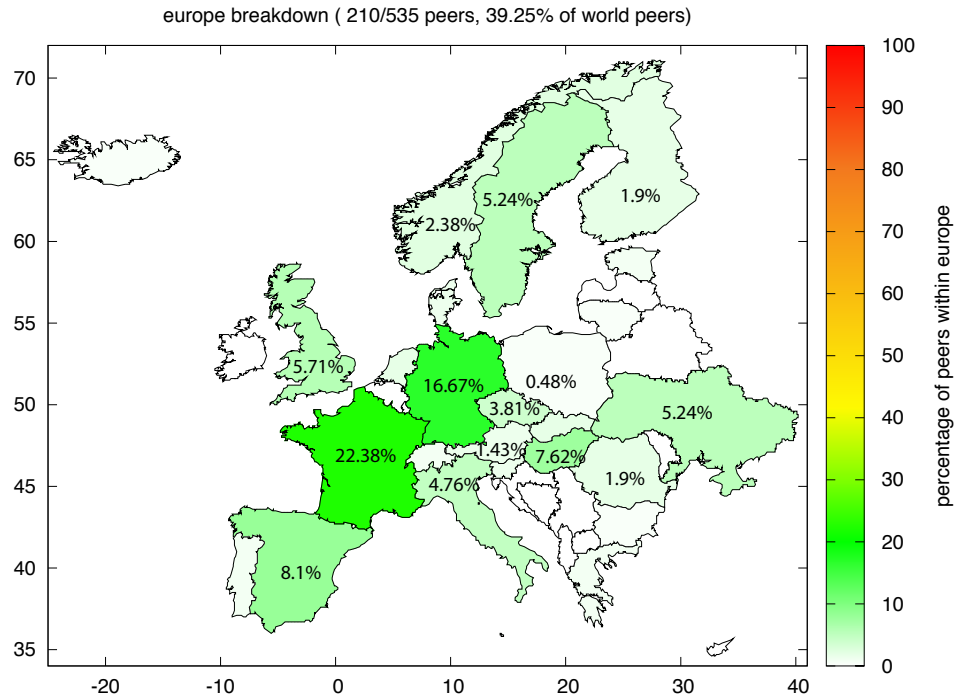


P2PGauge



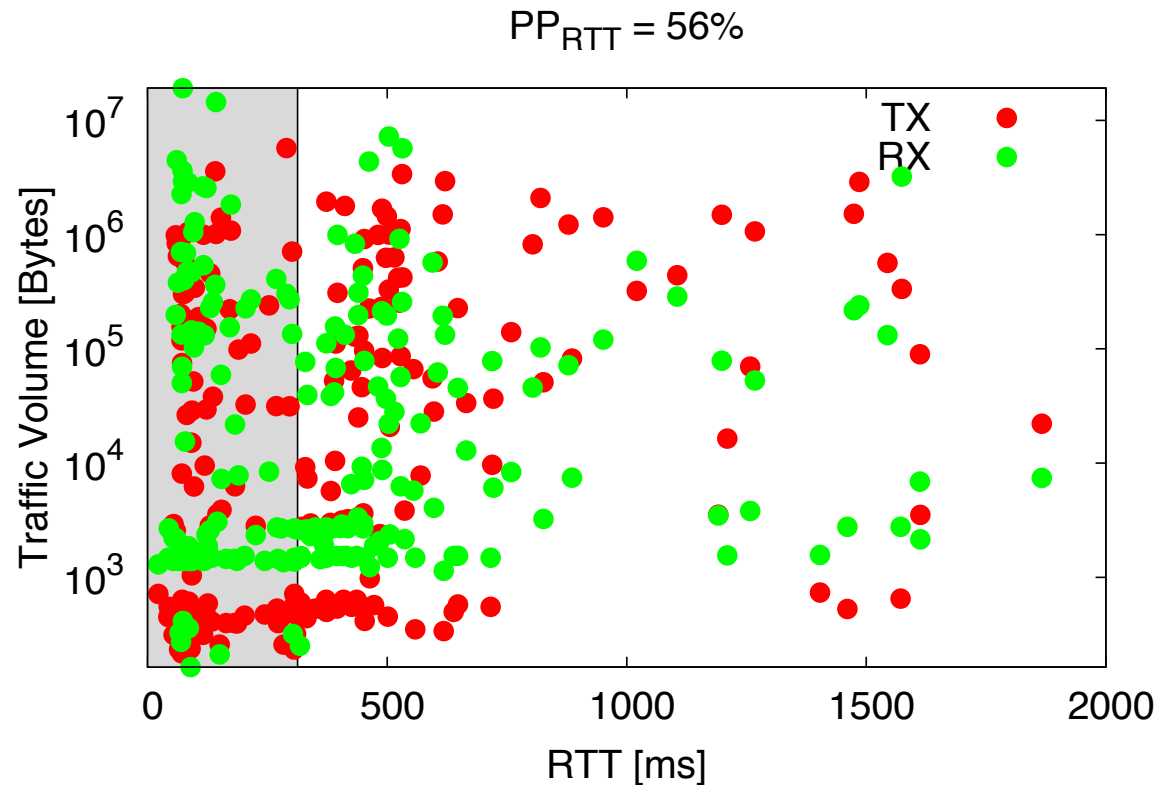
- Probe
 - unmodified P2P application
- Monitor captures traffic
 - measures (active and passive)
 - analyzes and visualizes results

Passive characterization



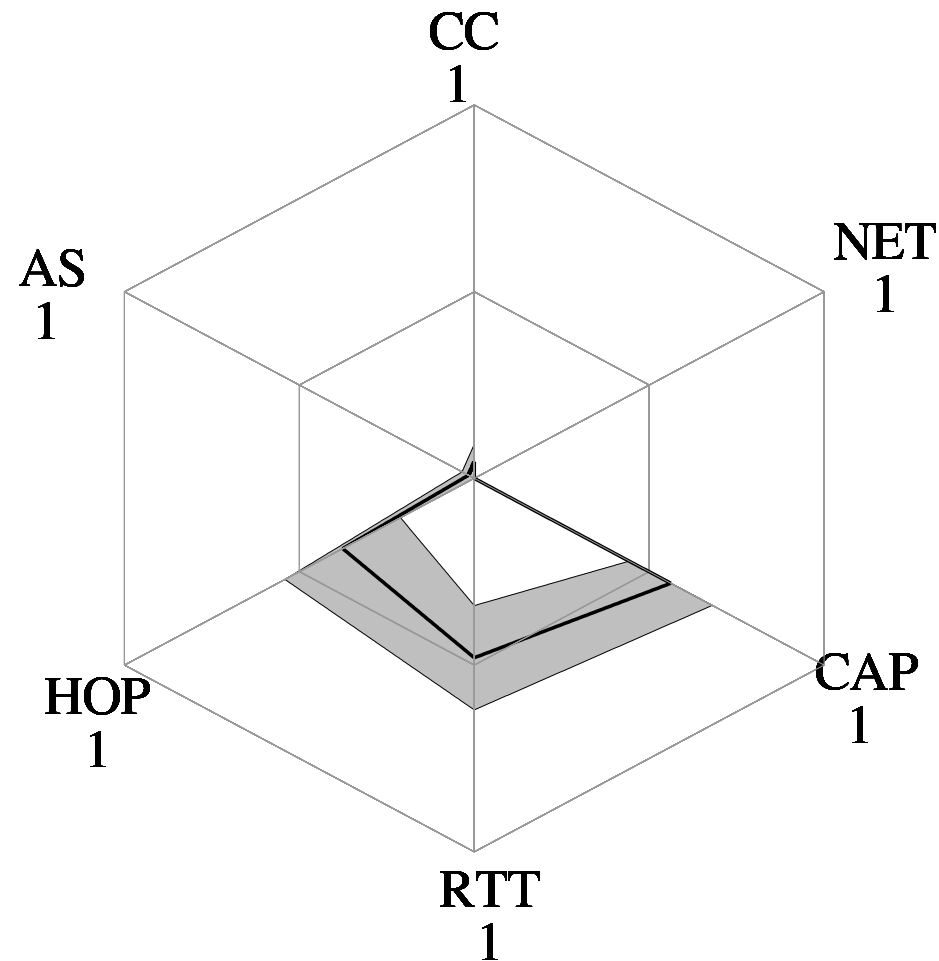
- breakdown of traffic between countries, AS
- How to represent network awareness “at a glance”?

Metrics



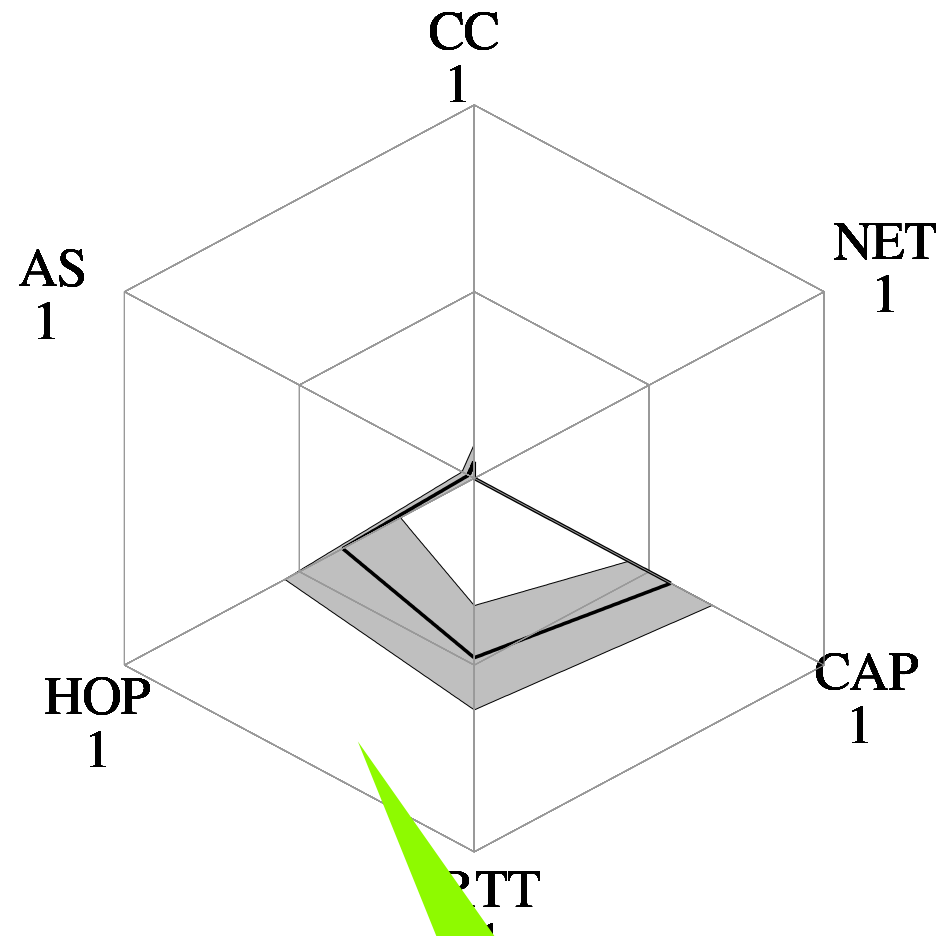
- Compact information into a single value
 - Multiple metrics are possible (Preferential Partition, Correlation, Kullback-Leibler distance, Bhattacharya distance)

Kiviati chart (Preferential Partition)



- Compact view
- Mean and variance
- Flexible

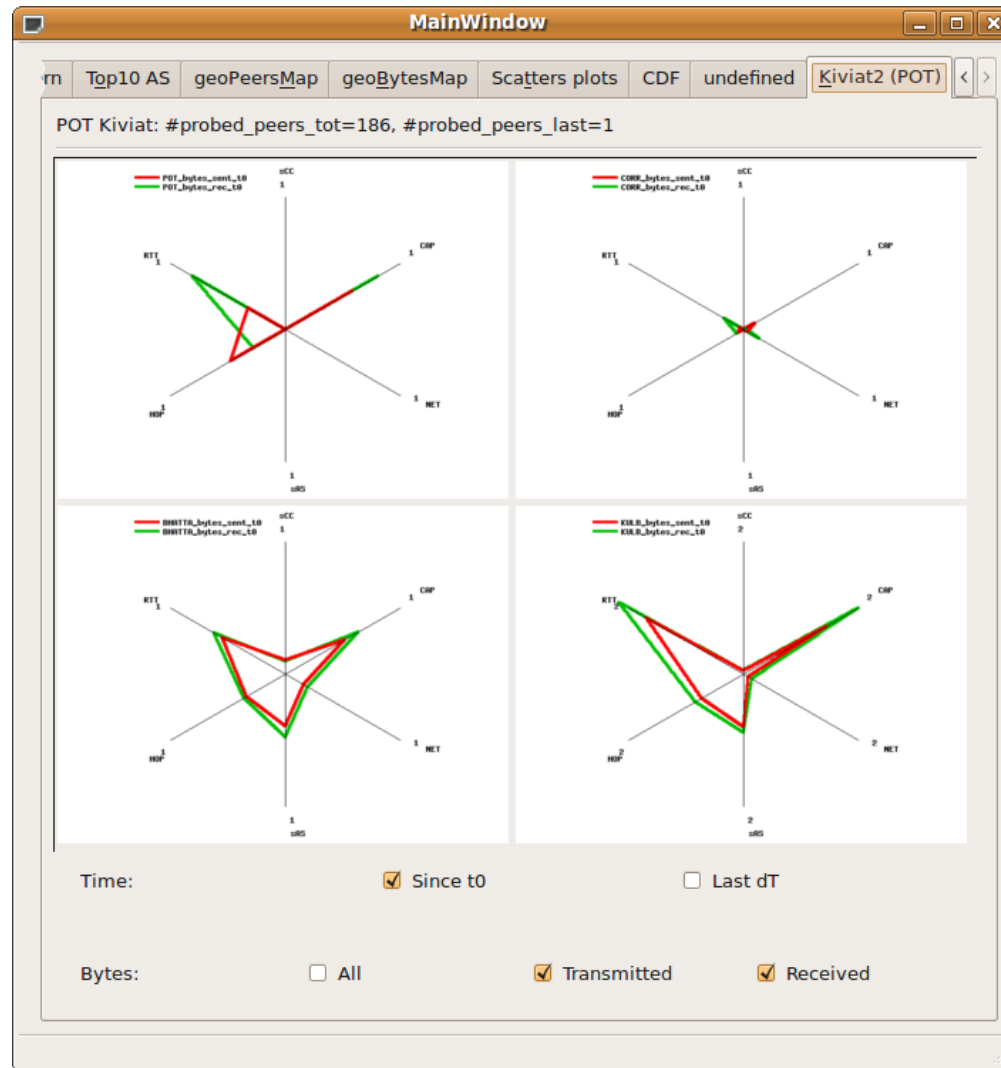
Kiviat chart (Preferential Partition)



- Compact view
- Mean and variance
- Flexible

- Slight Preference towards high capacity peers
- No preference for close peers (RTT, HOP)

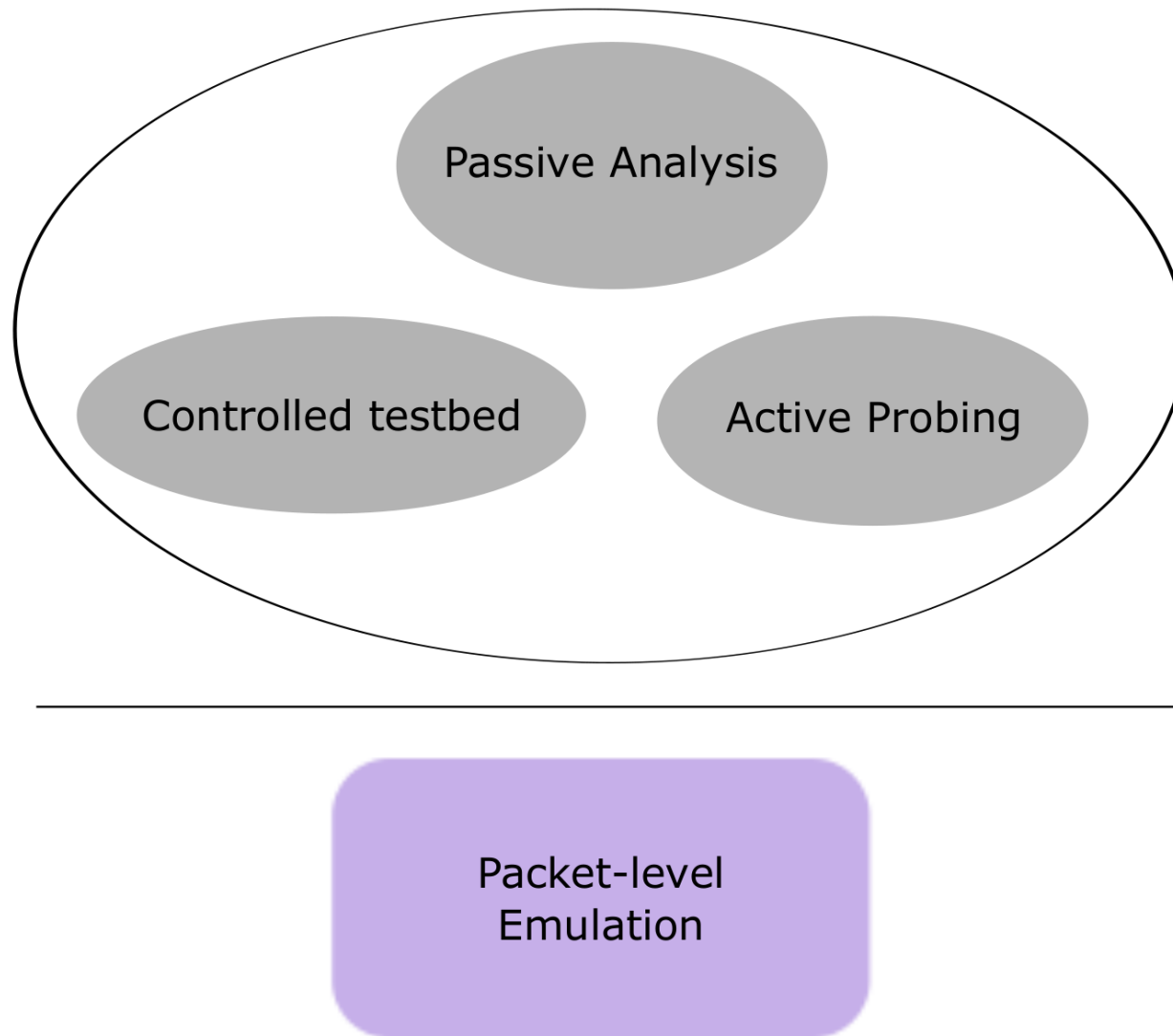
P2PGauge [1,2]



[1] <http://www.infres.enst.fr/~drossi/index.php?n=Software.P2PGauge>

[2] D. Rossi, E. Sottile, S. Valenti and P. Veglia, Gauging the network friendliness of P2P applications . In ACM SIGCOMM, Demo Session, Barcelona, Spain, August 2009.

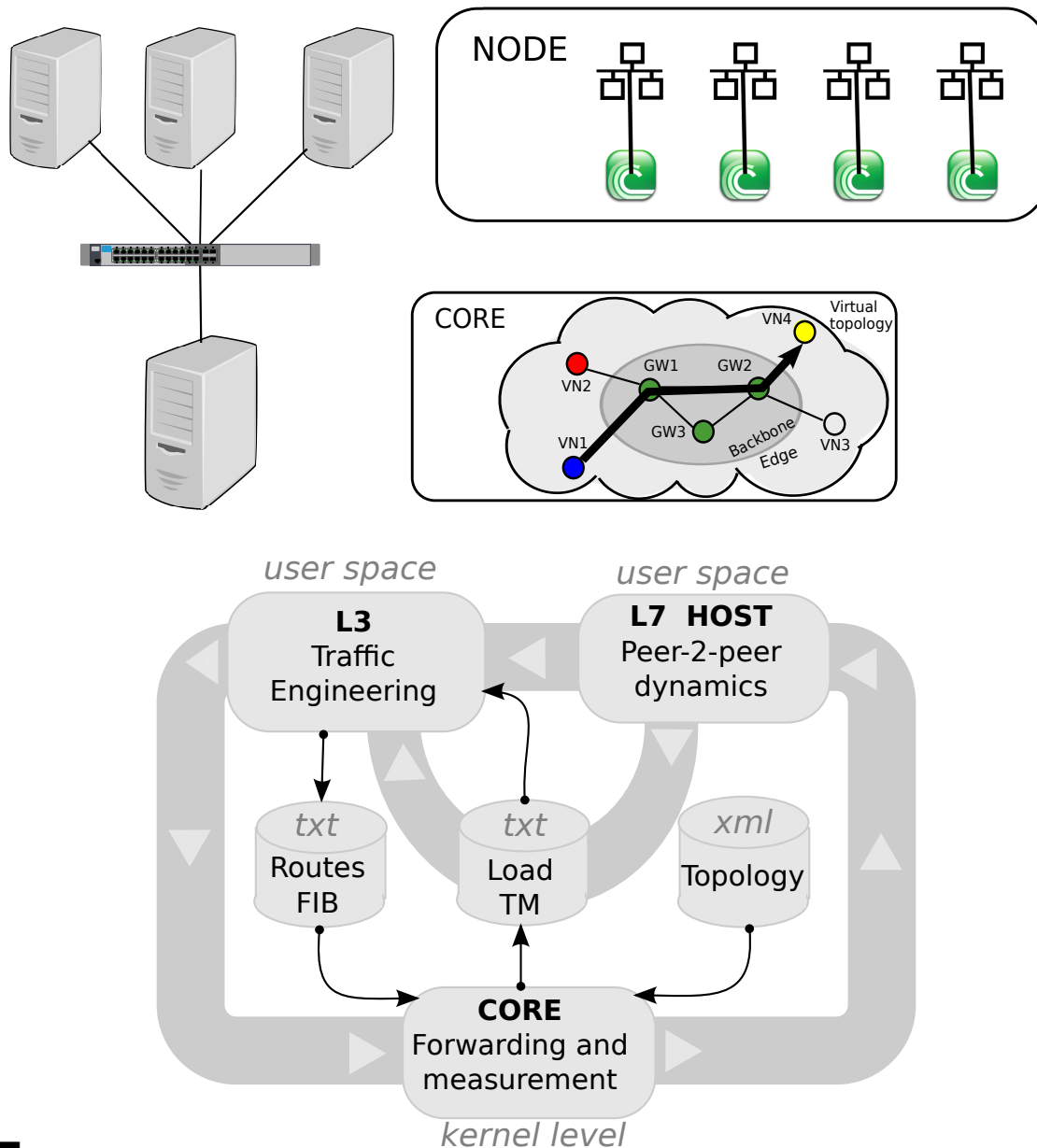
L3/L7 Interaction [9]



Motivations

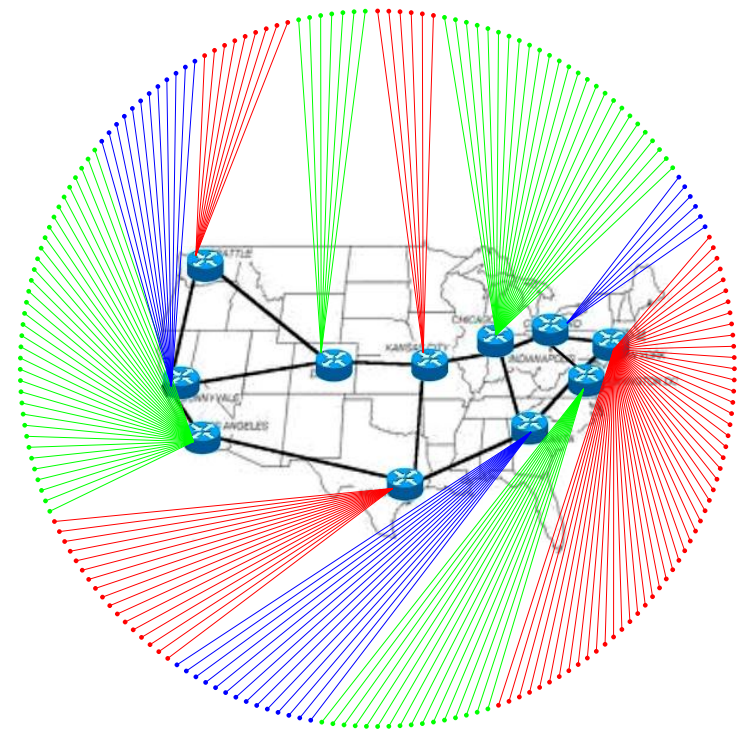
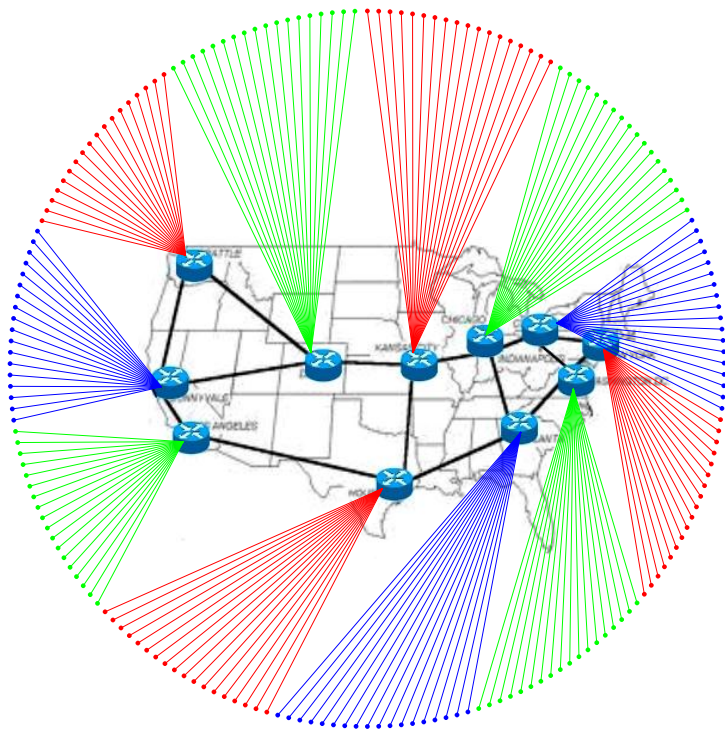
- Provide a tool to test P2P applications
- Complementary to Simulation and large scale testbed as PlanetLab
 - test **real** application
 - Full control on the network (topology, routing, load balancing, etc.)
 - Reproducibility
- Study the interaction of L7 P2P trading logic with L3 traffic engineering
 - especially important for network aware applications

Modelnet-TE



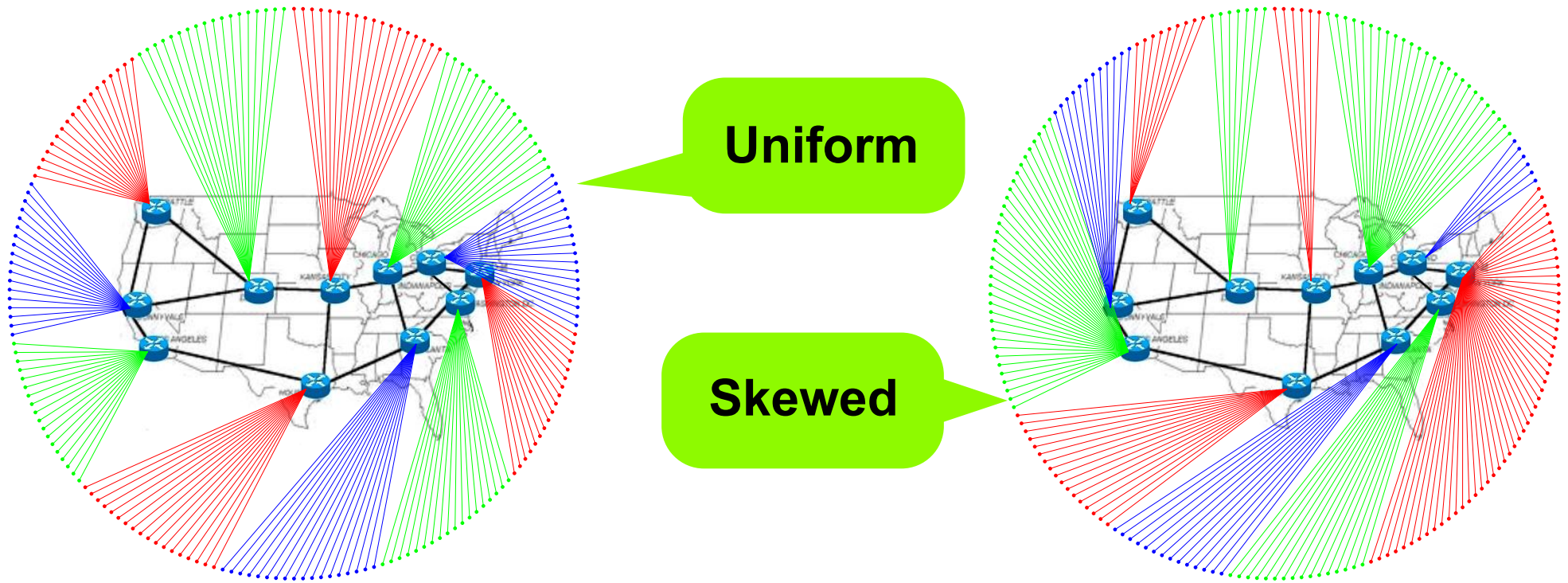
- Single core machine emulating topologies (capacities, latencies, etc)
- Nodes running multiple instances of applications binded to virtual interfaces
- All the traffic is forced through the Core
- ModelnetTE periodically performs traffic engineering according to traffic matrix

Scenarios



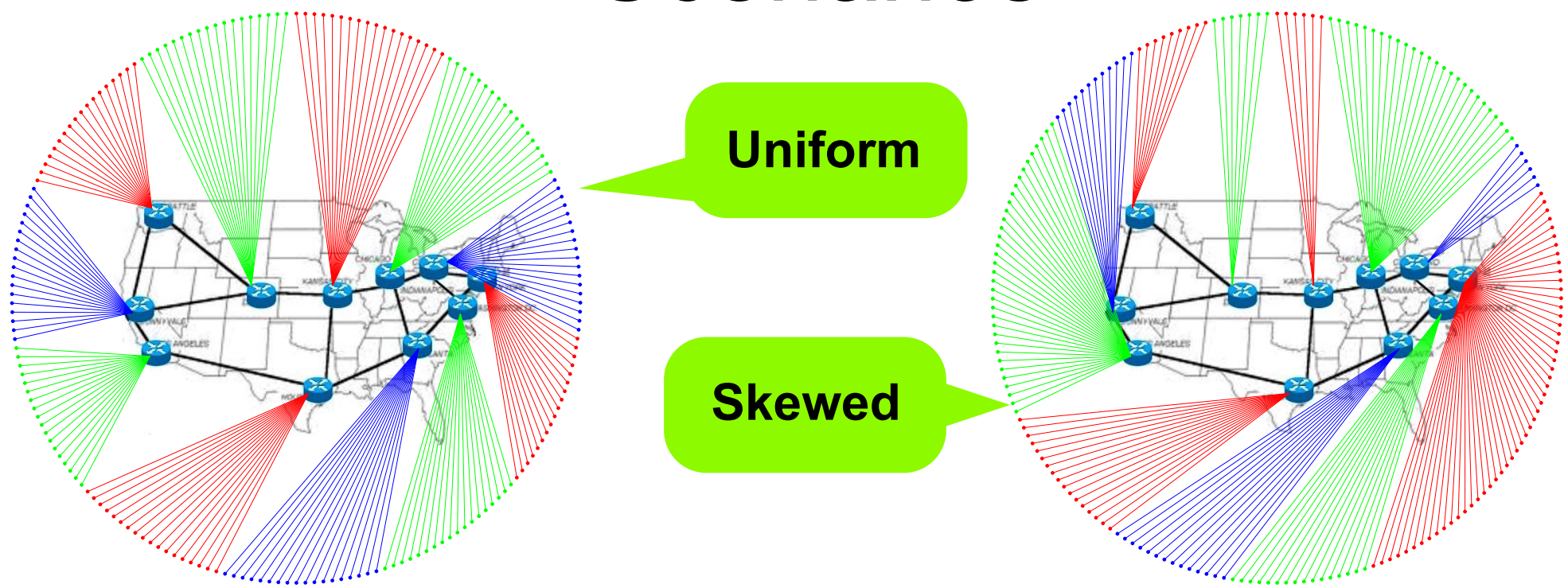
- Abilene topology (Internet 2)
- Uniform VS skewed population models
- IP (shortest path) VS Traffic engineering
- 200 peers

Scenarios



- Abilene topology (Internet 2)
- Uniform VS skewed population models
- IP (shortest path) VS Traffic engineering
- 200 peers

Scenarios



Uniform

Skewed

- Abilene topology (Internet 2)
- Uniform VS skewed population mode
- IP (shortest path) V
- 200 peers

Scalability issue!

Ref	Testbed	Nodes	Nodes/machine
R6	Grid5000	10000	100
R7	PlanetLab	400	1
R8	Modelnet	320	32
R9	Grid5000	300	100
	ModelnetTE	200	35
R8	PlanetLab	160	1

Applications



BitTorrent

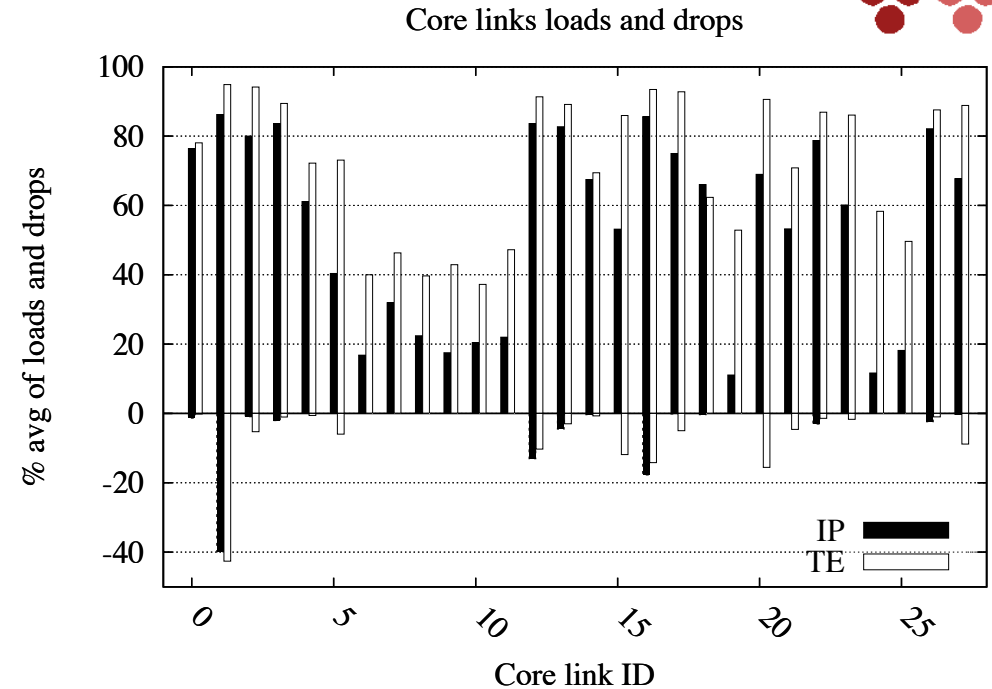
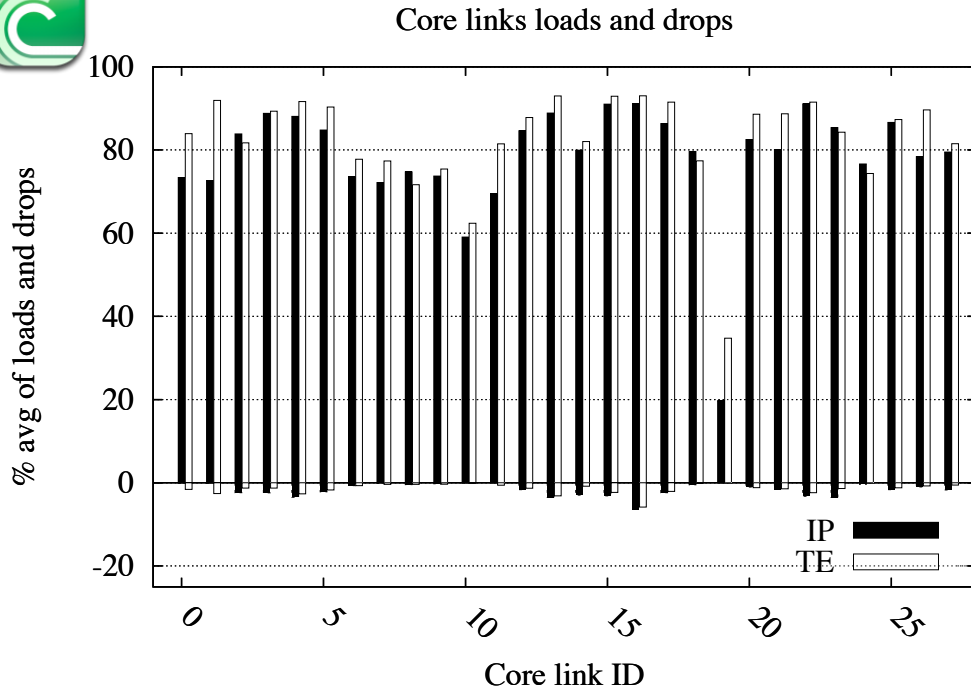
- Filesharing
- TCP
- elastic traffic
- rarest-first chunk trading logic



WineStreamer

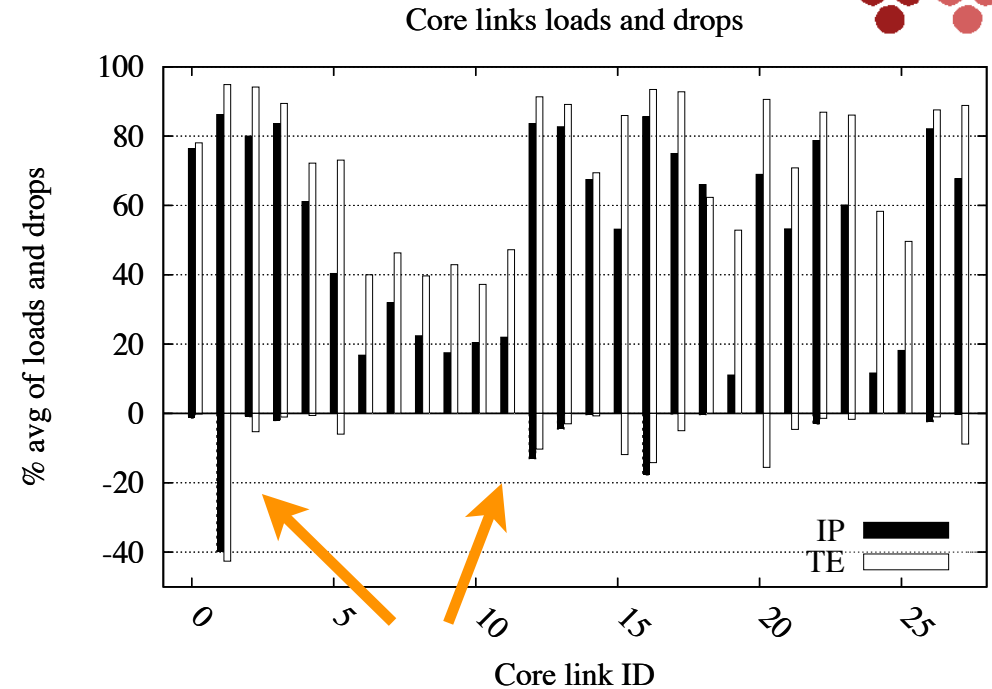
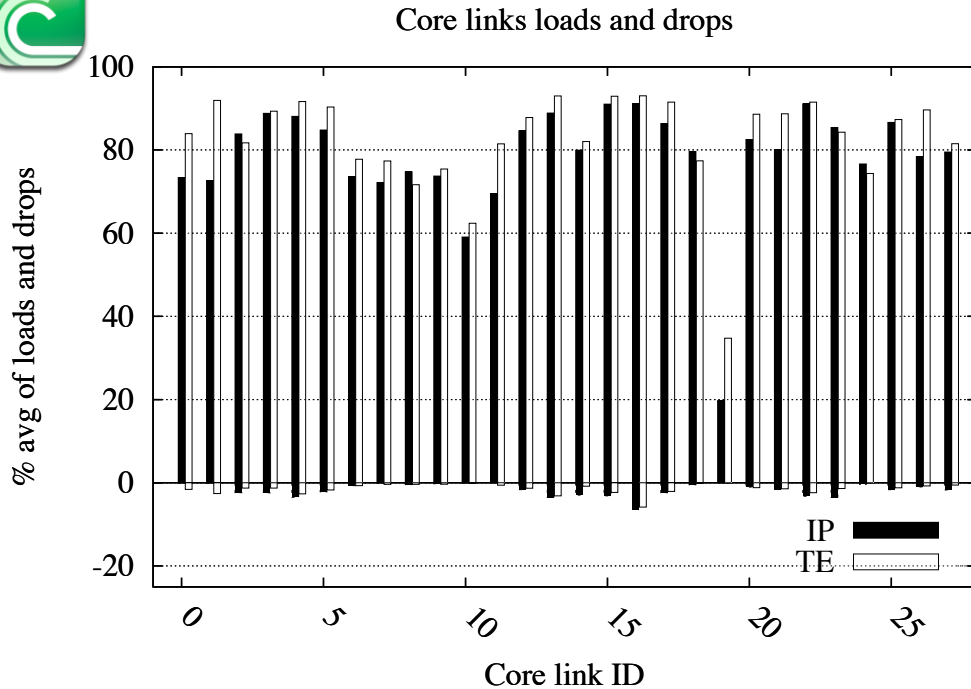
- Live Network Aware P2P-TV
- NapaWine
- UDP
- Minimum rate
- Playout deadline based

Link load



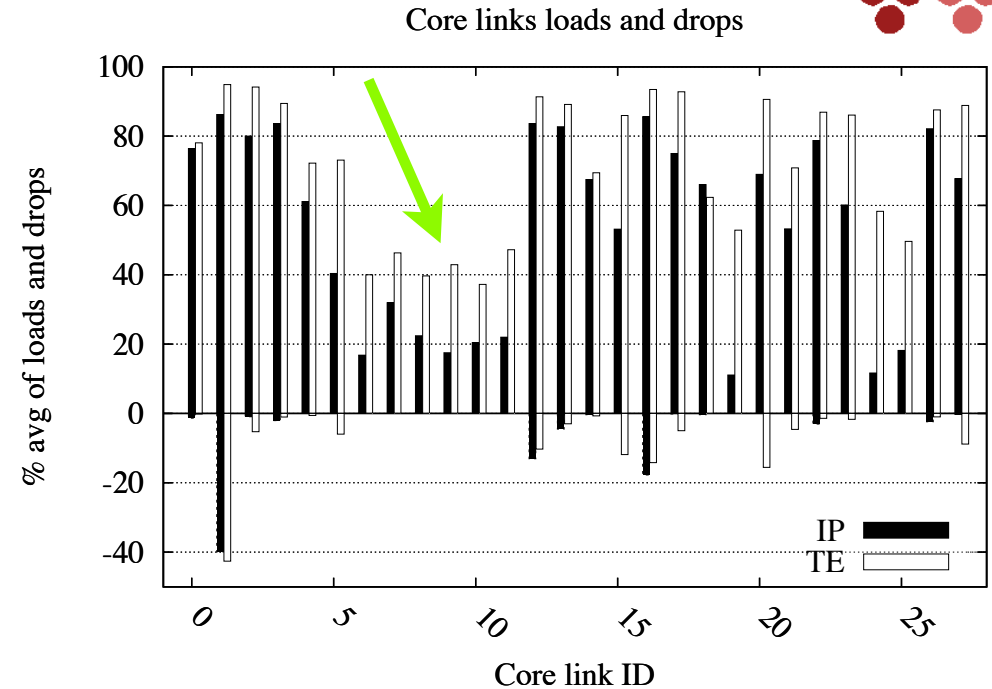
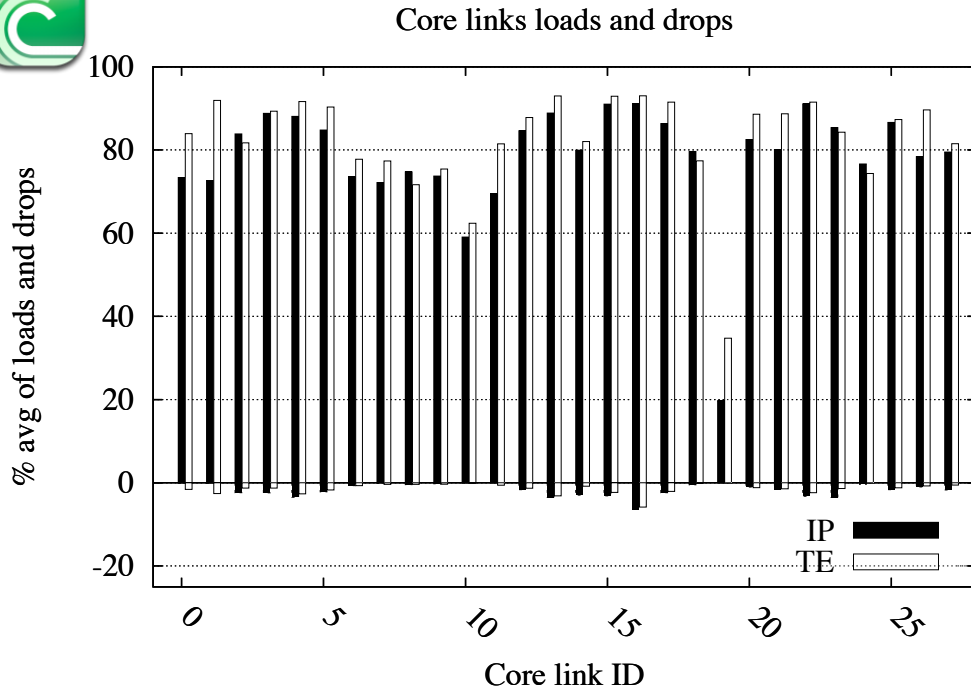
- BitTorrent and TCP fully exploit network capacity (5mpbs each core link)
- Almost no drops for BitTorrent
- WineStreamer heavily congests **some links**
- Traffic Engineering **improves** utilisation on some links
- But can **create congestion** on otherwise unloaded links!

Link load



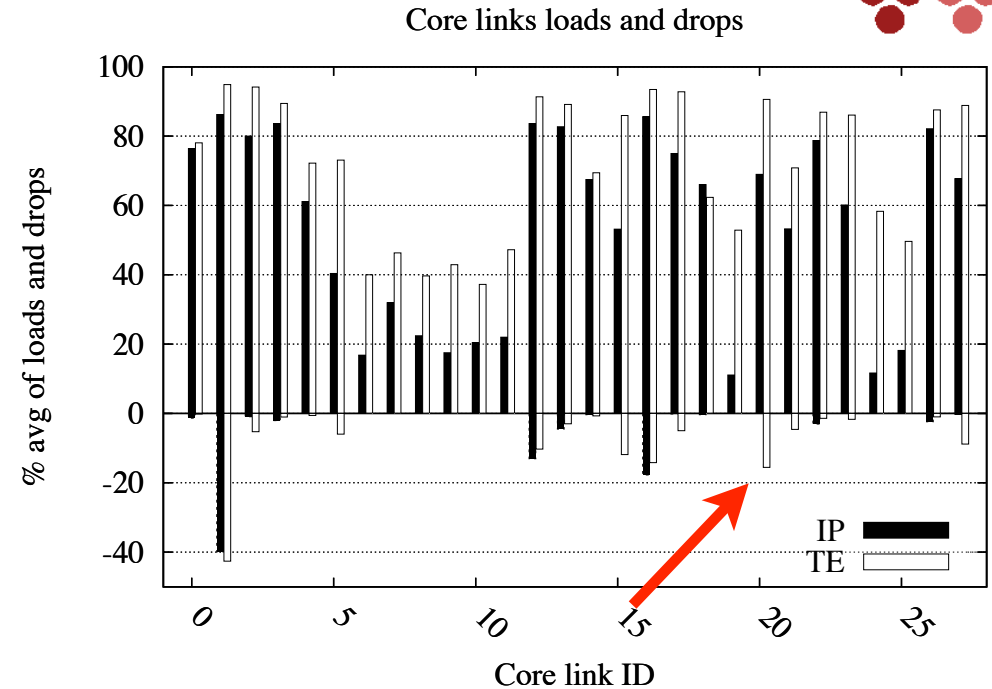
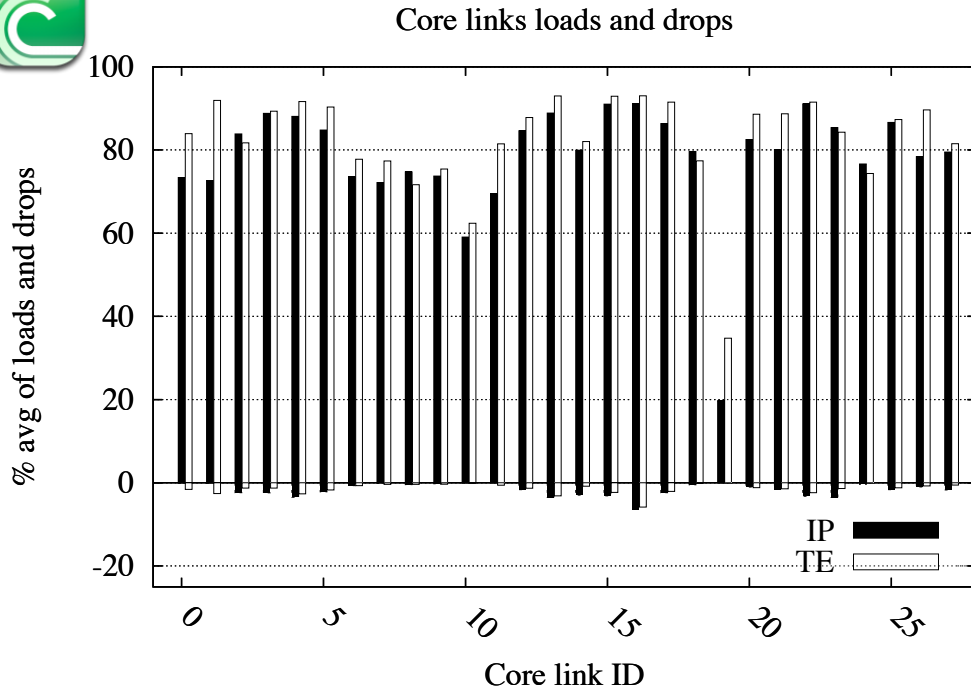
- BitTorrent and TCP fully exploit network capacity (5mpbs each core link)
- Almost no drops for BitTorrent
- WineStreamer heavily congests **some links**
- Traffic Engineering **improves** utilisation on some links
- But can **create congestion** on otherwise unloaded links!

Link load



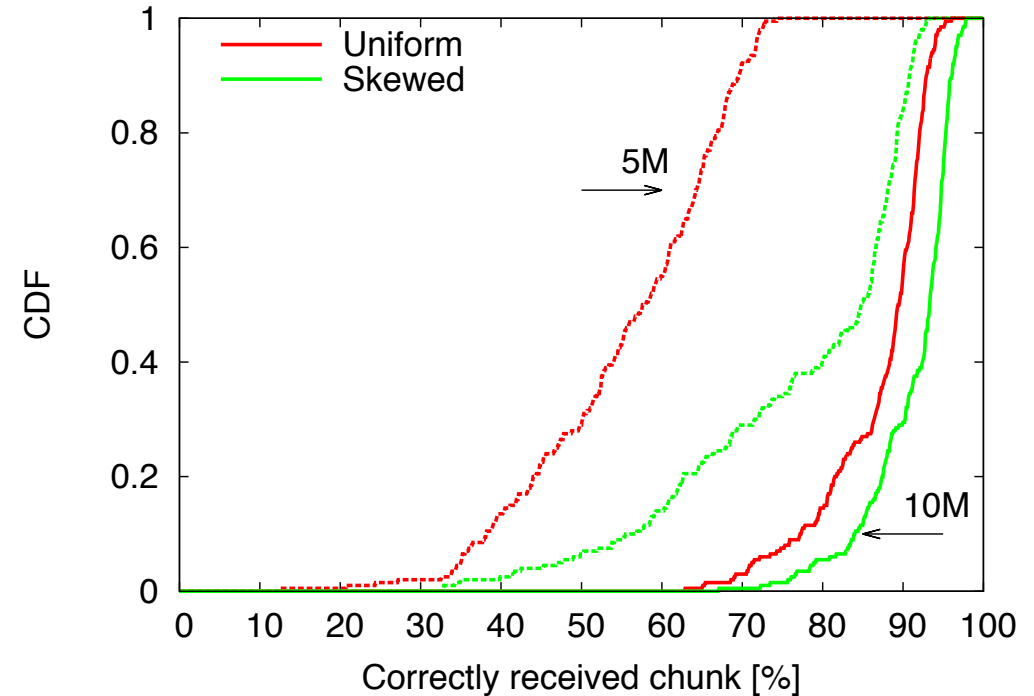
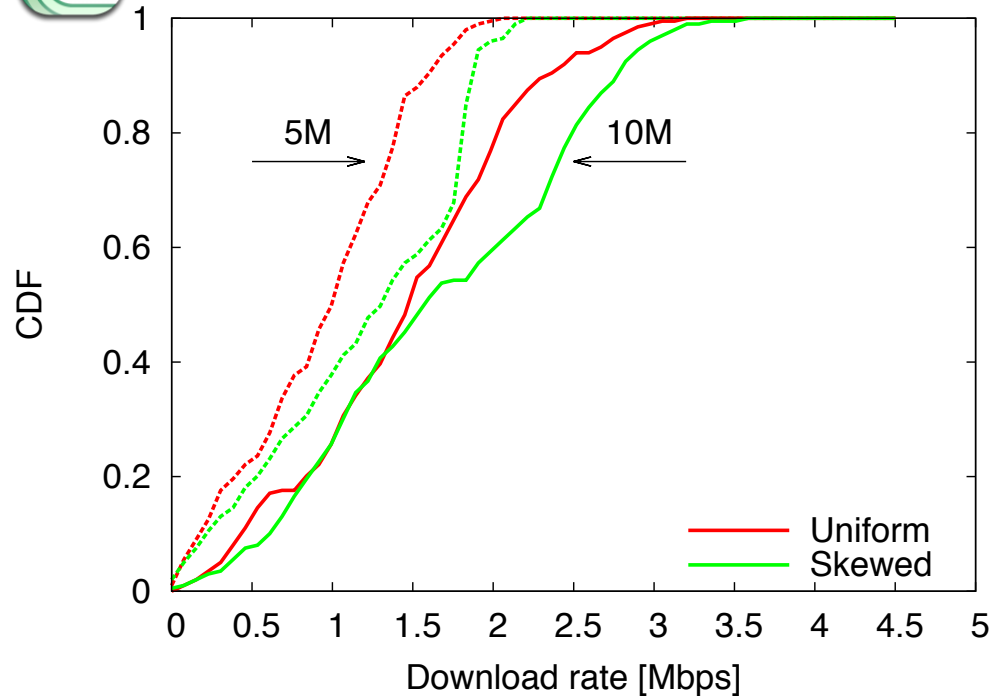
- BitTorrent and TCP fully exploit network capacity (5mpbs each core link)
- Almost no drops for BitTorrent
- WineStreamer heavily congests **some links**
- Traffic Engineering **improves** utilisation on some links
- But can **create congestion** on otherwise unloaded links!

Link load



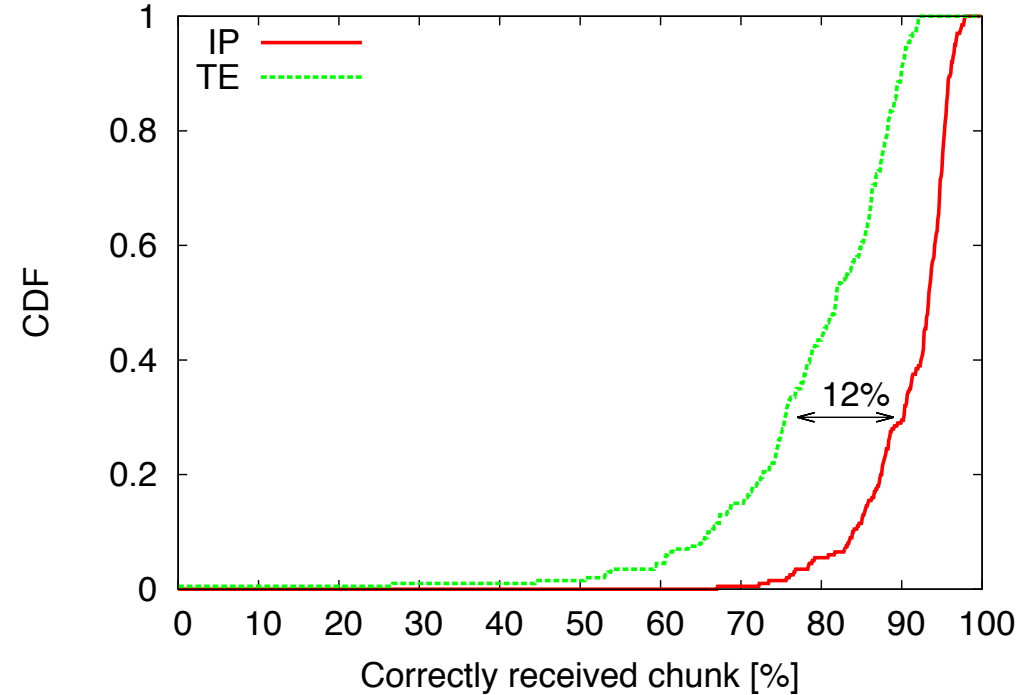
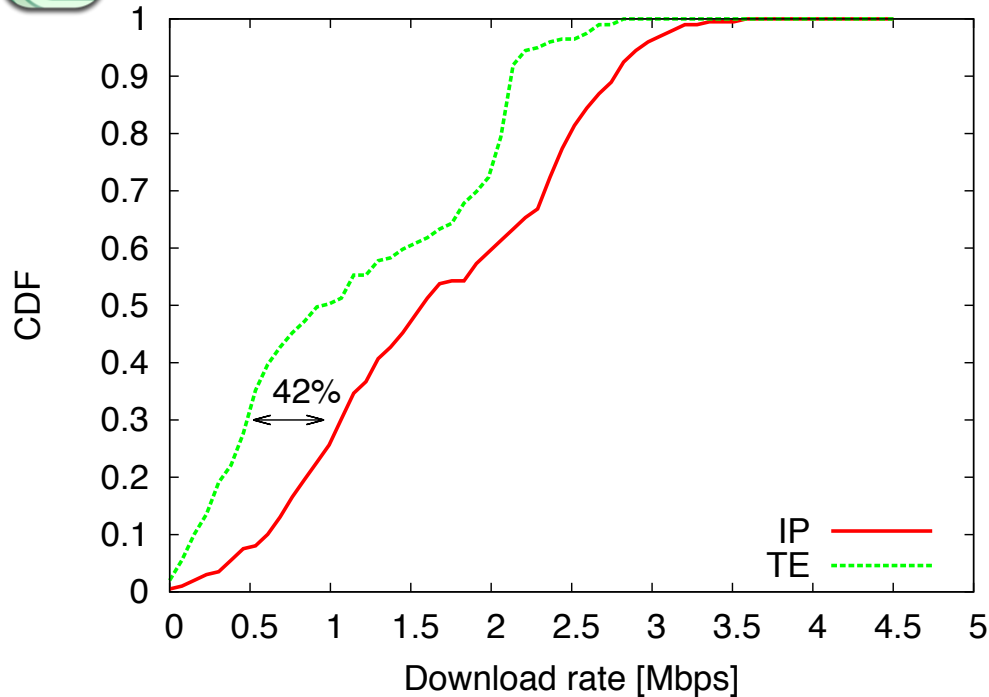
- BitTorrent and TCP fully exploit network capacity (5mpbs each core link)
- Almost no drops for BitTorrent
- WineStreamer heavily congests **some links**
- Traffic Engineering **improves** utilisation on some links
- But can **create congestion** on otherwise unloaded links!

Population model and network capacity



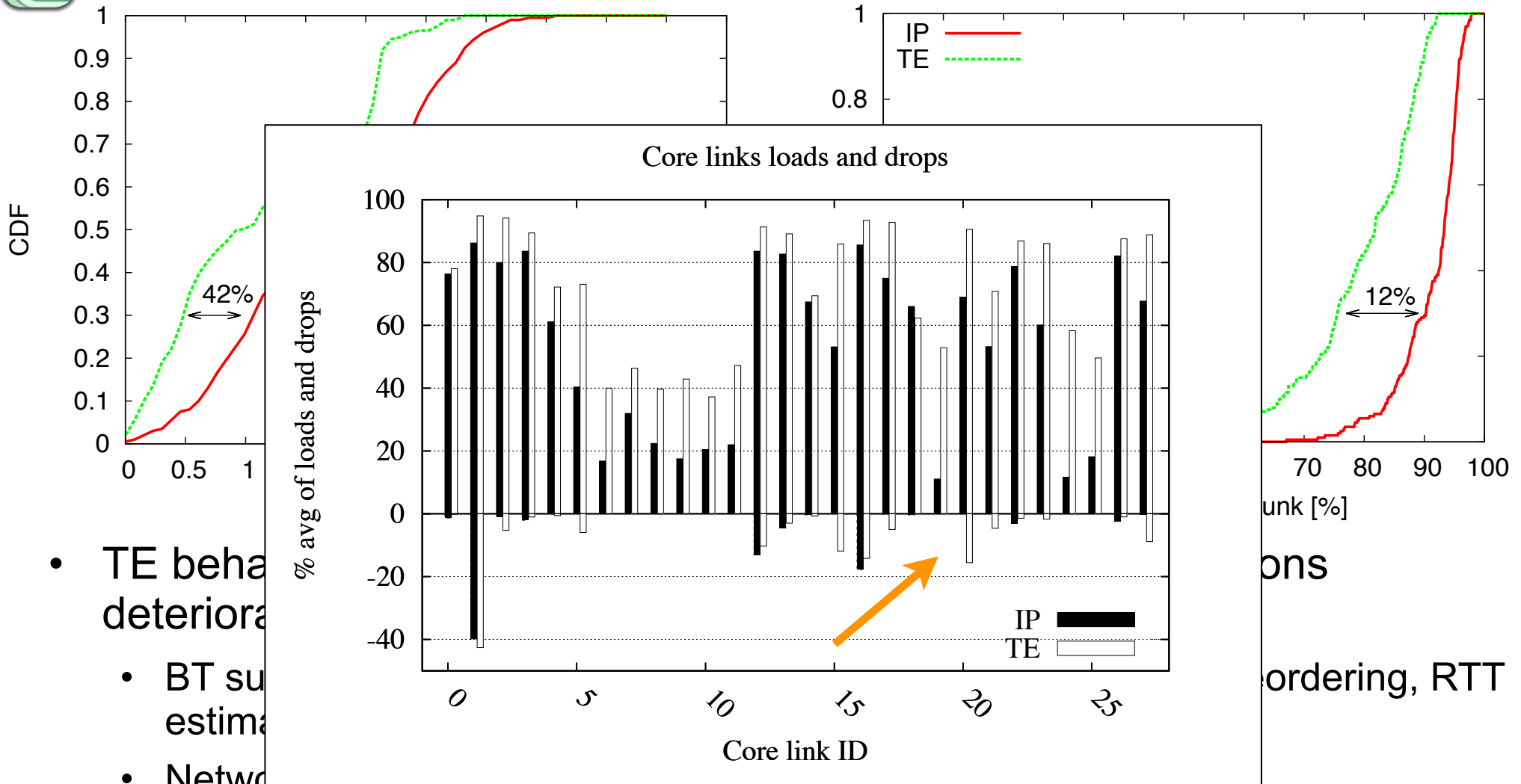
- Skewed population model is beneficial for both applications
 - Higher probability to choose peers behind the same router
 - TCP performs better
 - NA applications can **explicitly** choose close peers to avoid narrow links

Traffic Engineering



- TE behavior counterintuitive: performance of both applications deteriorates
 - BT suffers longer paths and interaction of TE and TCP is bad (reordering, RTT estimation)
 - Network Awareness and TE do not play well together
 - Abilene network is a hard scenario for TE algorithms since there is no path-diversity

Traffic Engineering



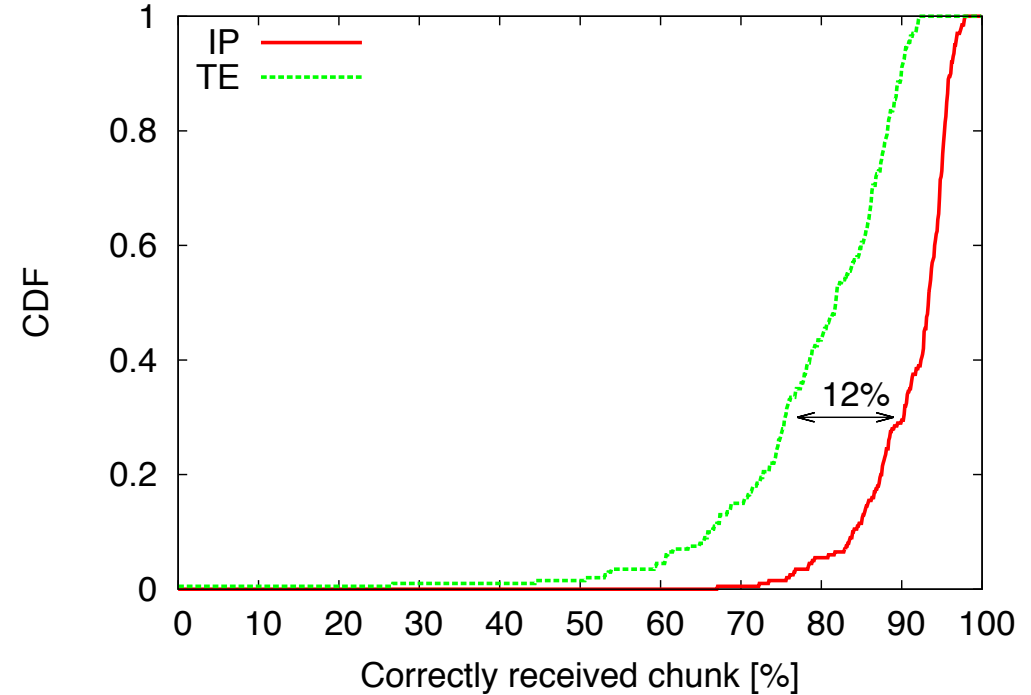
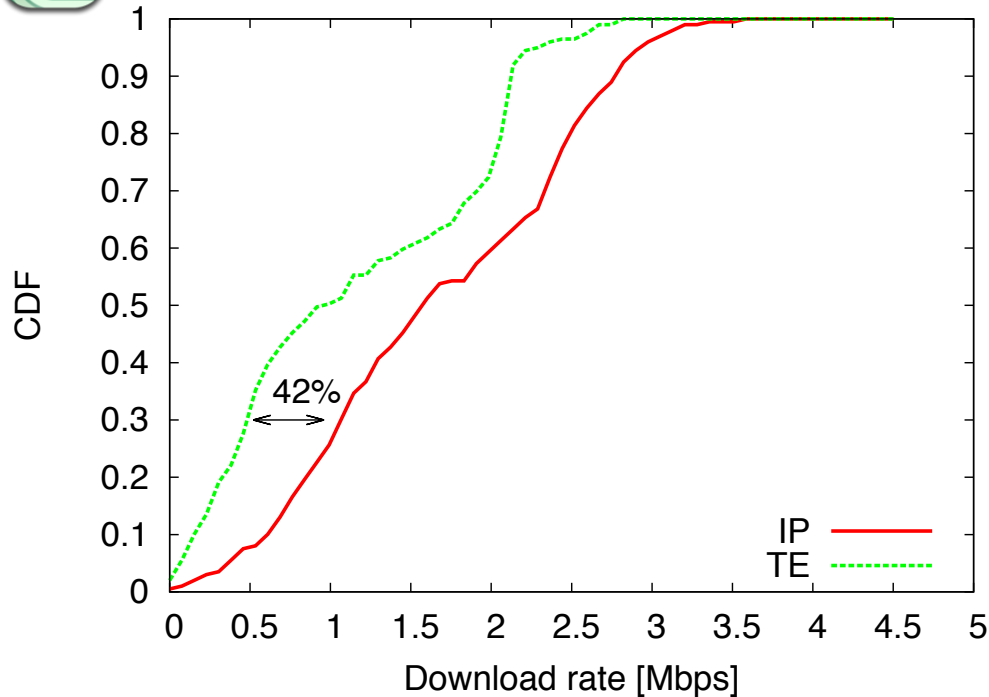
- TE behavior deteriorates

- BT success estimation
- Network awareness and TE do not play well together

- Abilene network is a hard scenario for TE algorithms since there is no path-diversity

ons
ordering, RTT

Traffic Engineering



- TE behavior counterintuitive: performance of both applications deteriorates
 - BT suffers longer paths and interaction of TE and TCP is bad (reordering, RTT estimation)
 - Network Awareness and TE do not play well together
 - Abilene network is a hard scenario for TE algorithms since there is no path-diversity

Contributions summary

- We provided procedures to test P2P applications' Network Awareness
 - Through passive measures (dataset with 40 nodes, 3 applications)
 - Controlled testbed to measure preference towards path-wise metrics
 - P2PGauge tool which exploits active and passive measures to show at a glance applications'NA
- Developed procedures and tools to test interaction of P2P application with L3 network
 - Simulation campaign to test NA algorithms in realistic scenarios
 - ModelnetTE which allows the study of the interaction of L7 routing over reactive networks

Contribution summary (Lessons learned)

- Measurement
 - application mostly bandwidth aware
- Simulation
 - Bandwidth aware is a good choice for QoE (in ADSL chunk TX time \gg propagation delay)
- ModelnetTE
 - packet level interaction (load balancing, TCP, overlay routing) may have unwanted effects

Future work

- Use our tools to test new generation applications
- Parallel Modelnet core
- New topologies and TE algorithm
 - understand which TE algorithms/topologies perform better
- Web integration
 - implement P2P logic into browsers
- Signaling dynamics
 - better study the trade-offs between signaling overhead and outdated knowledge

References

- [R1] P4P <http://cs-www.cs.yale.edu/homes/yong/p4p.html>
- [R2] Application Layer Traffic Optimization <http://datatracker.ietf.org/wg/alto/charter/>
- [R3] Bindal et Al. “Improving Traffic Locality in BitTorrent via Biased Neighbor Selection”. In *ICDCS’06*.
- [R4] Silva et Al. “A bandwidth-Aware scheduling Strategy for P2P-TV Systems”. In *P2P’08*.
- [R5] Ren et Al. “On Reducing Mesh Delay for Peer-to-Peer Live Streaming”. In *INFOCOM’08*.
- [R6] Blond et Al. “Pushing BitTorrent locality to the limit”. In *Elsevier Computer Networks*, 55(3), 2011.
- [R7] Dale et Al. “Evolution and enhancement of BitTorrent network topologies”. In *IWQoS’08*.
- [R8] Picconi et Al. “Isp friend or foe? Making P2P live streaming isp-aware”. In *ICDCS’09*.
- [R9] Rao et Al. “Can realistic BitTorrent experiments be performed on clusters?”. In *P2P’10*.

Publications

Published

- (1) Assessing the impact of signaling on the QoE of push-based P2P-TV diffusion algorithms. D. Rossi and P. Veglia. In IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, Feb. 2011.
- (2) Network Awareness of P2P Live Streaming Applications: A Measurement Study. D. Ciullo, M. A. Garcia, A. Horvath, E. Leonardi, M. Mellia, D. Rossi, M. Telek and P. Veglia. IEEE Transactions on Multimedia, Volume 12, Issue:1, 54-63, Jan. 2010.
- (3) Black-box analysis of Internet P2P applications. D. Rossi, E. Sottile and P. Veglia. Peer-to-Peer Networking and Applications, Volume 4, Number 2, 146-164, Jun. 2010.
- (4) A Hybrid Approach to Assess the Network Awareness of P2P-TV Applications. D. Rossi and P. Veglia. International Journal of Digital Multimedia Broadcasting, Volume 2010, Article ID 826351, 11 pages, Nov. 2010.
- (5) Network Awareness of P2P Live Streaming Applications. D. Ciullo, M. Garcia, A. Horvath, E. Leonardi, M. Mellia, D. Rossi, M. Telek and P. Veglia. In HotP2P'09, Rome, Italy, May 2009.
- (6) Gauging the network friendliness of P2P applications. D. Rossi, E. Sottile, S. Valenti and P. Veglia. In SIGCOMM demo session, Barcelona, Spain, Aug. 2009.
- (7) Pictures from the Skype. D. Rossi, S. Valenti, P. Veglia and D. Bonfiglio. ACM Performance Evaluation Review, Sep. 2008.

Under review

- (8) P. Veglia, D. Rossi, Performance evaluation of P2P-TV diffusion algorithms under realistic settings, submitted to Springer Peer-to-Peer Networking and Applications.
- (9) P. Veglia, D. Rossi, ModelNet-TE: An emulation tool for the study of P2P and Traffic Engineering interaction dynamics, submitted to Springer Peer-to-Peer Networking and Applications.

Thank you for your attention!

Question time

Oops (guard slide)

NapaWine induced bias

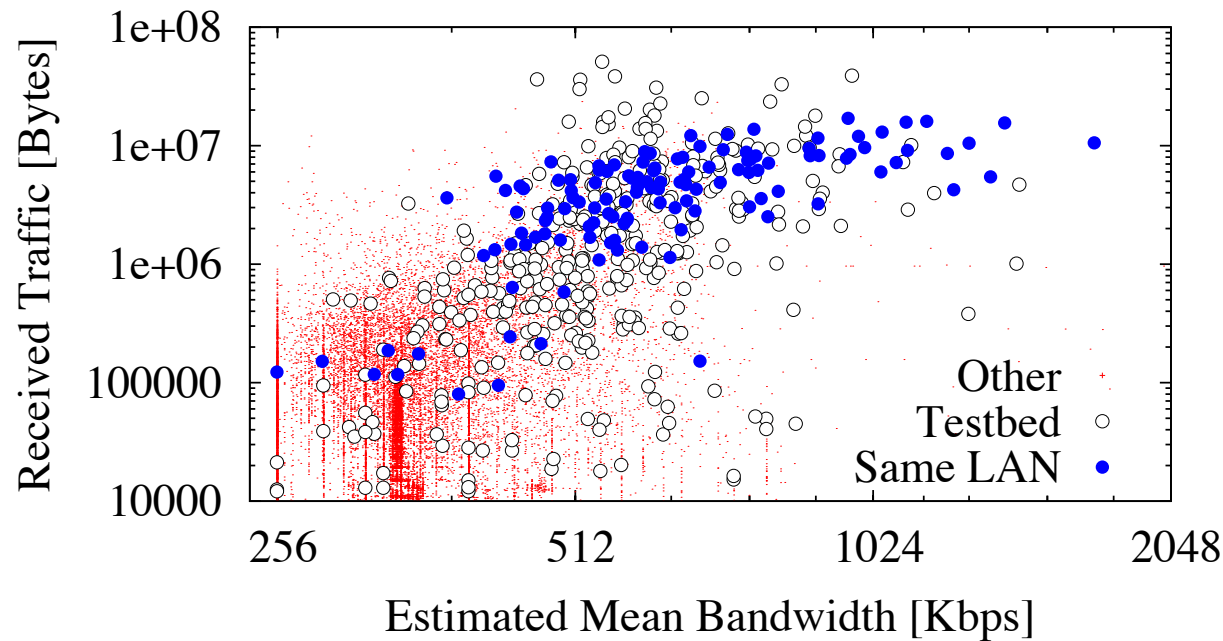
Table 3.2: NAPA-WINE induced Bias

App	Contributors		All-peers	
	Peer%	Bytes%	Peer%	Bytes%
PPLive	0.95	3.54	0.10	3.33
SopCast	10.25	17.71	4.60	19.45
TVAnts	29.82	56.31	15.56	56.06

ToN complete table

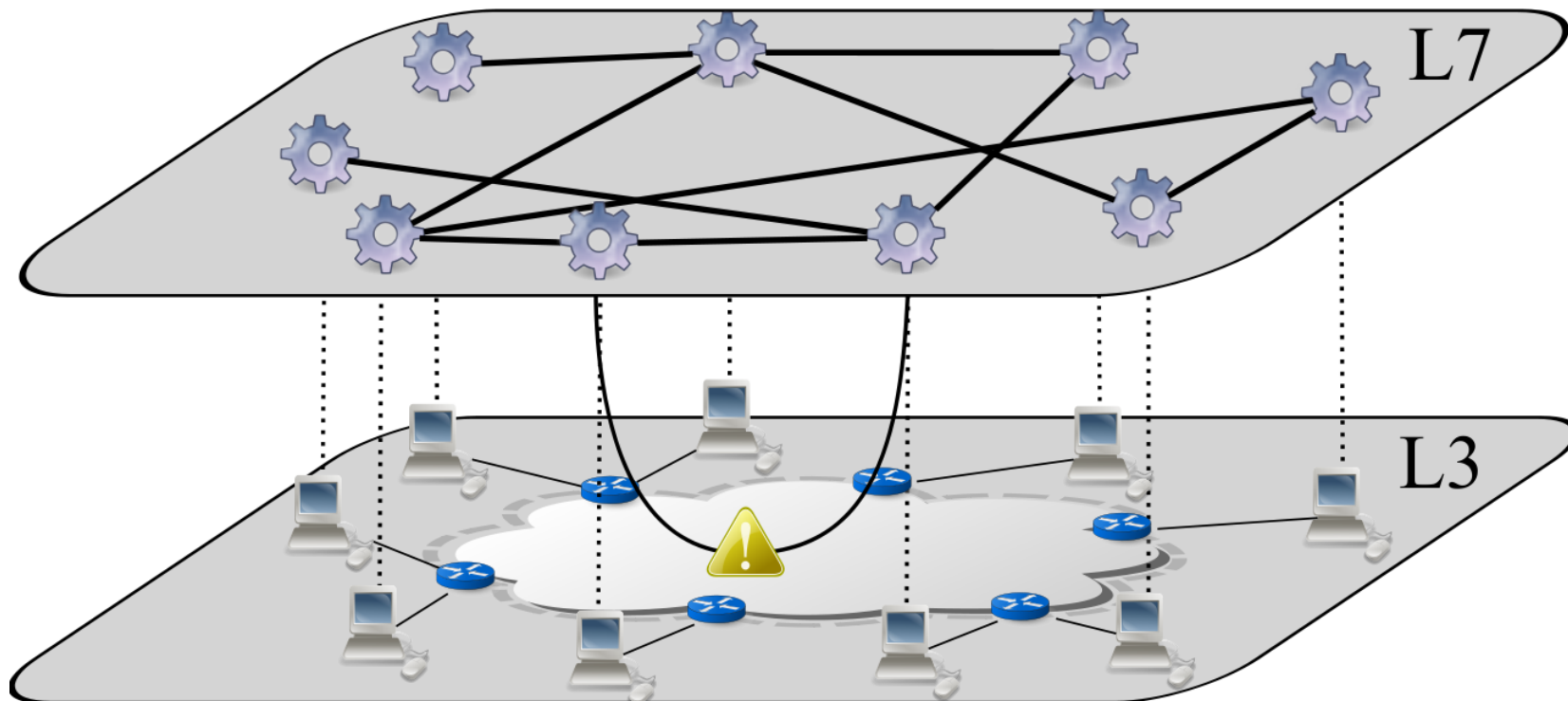
		Download				Upload			
		Non-Napa		All Contributors		Non-Napa		All Contributors	
Net	App	B'_D %	P'_D %	B_D %	P_D %	B'_U %	P'_U %	B_U %	P_U %
AS	PPLive	6.5	0.6	12.8	1.3	0.8	0.2	1.8	0.5
	SopCast	0.6	0.7	3.5	3.9	1.7	0.7	6.4	3.9
	TVAnts	7.3	3.3	32.0	13.5	11.6	1.8	30.1	9.6
CC	PPLive	6.5	0.6	13.1	1.4	1.1	0.3	2.1	0.6
	SopCast	0.6	0.8	4.0	4.4	1.7	0.8	7.2	4.4
	TVAnts	7.6	4.0	37.9	16.3	14.3	3.1	37.7	12.5
NET	PPLive	-	-	9.9	0.8	-	-	1.4	0.3
	SopCast	-	-	2.0	2.6	-	-	3.5	2.6
	TVAnts	-	-	18.1	6.7	-	-	18.1	5.4
HOP	PPLive	42.2	41.1	51.4	42.4	30.4	40.4	31.7	41.0
	SopCast	29.0	40.7	37.9	48.0	45.9	43.0	56.9	49.8
	TVAnts	62.1	55.0	81.1	71.9	57.8	53.0	78.9	67.2
SYM	PPLive	3.3	4.8	4.3	5.0	-	-	-	-
	SopCast	6.7	13.0	7.8	14.2	-	-	-	-
	TVAnts	12.4	10.9	20.0	14.3	-	-	-	-

Passive BW correlation

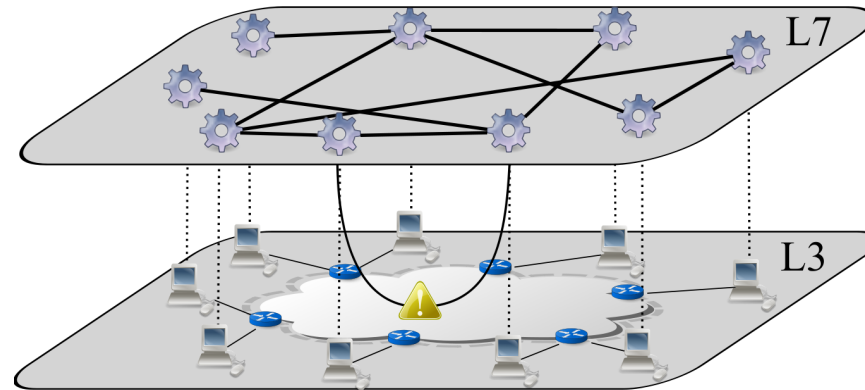


$X :$		$Y :$ $BW_{\Delta T}$		BW_N	
		mean	p_{99}	mean	p_{99}
AS		0.28	0.44	0.27	0.29
CC		0.27	0.42	0.26	0.26
RX	all	0.43	0.54	0.45	0.53
	LAN	0.62	0.75	0.51	0.61
	!LAN	0.33	0.40	0.37	0.45

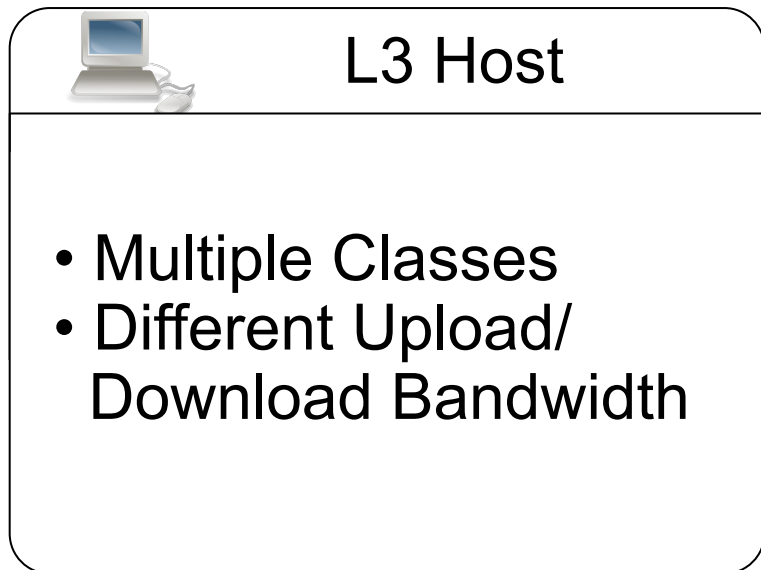
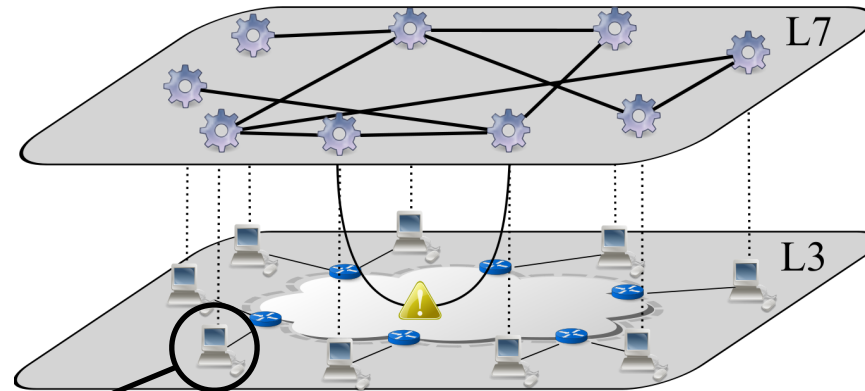
The big picture



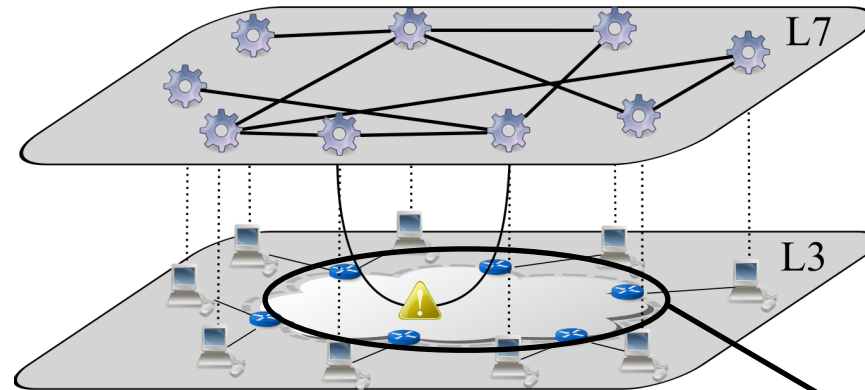
Layer 3 components




Layer 3 components




Layer 3 components



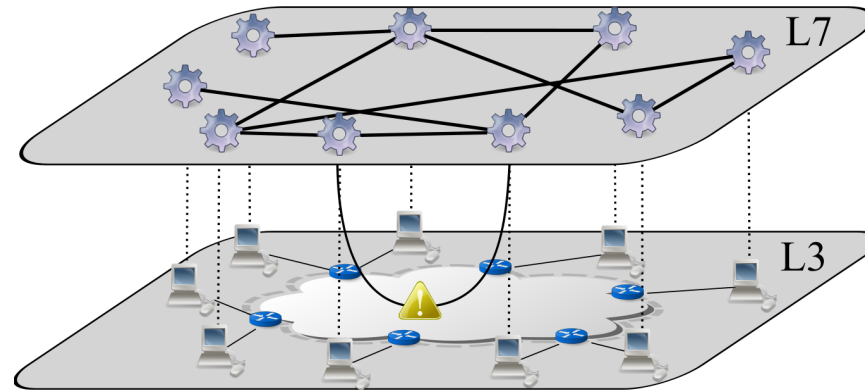
 **L3 Host**

- Multiple Classes
- Different Upload/Download Bandwidth

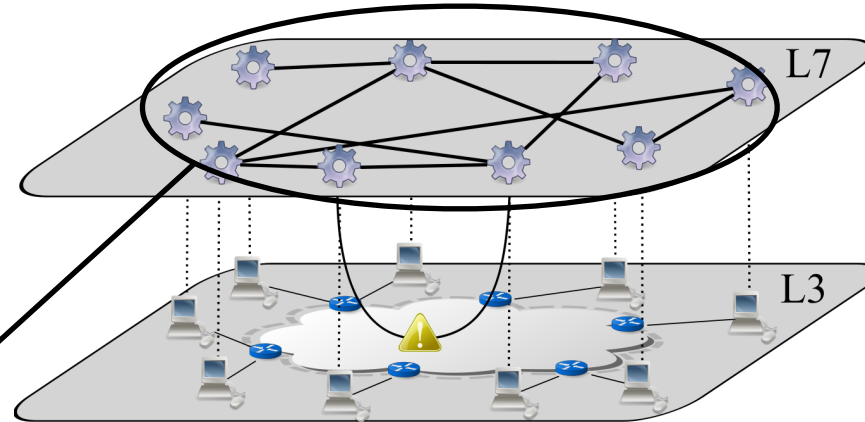
L3 Network 

- Static_{ext}
 - Overlay only (latency = 0)
 - Constant delay
 - **Meridian latency matrix**
- Dynamic

Layer 7 components



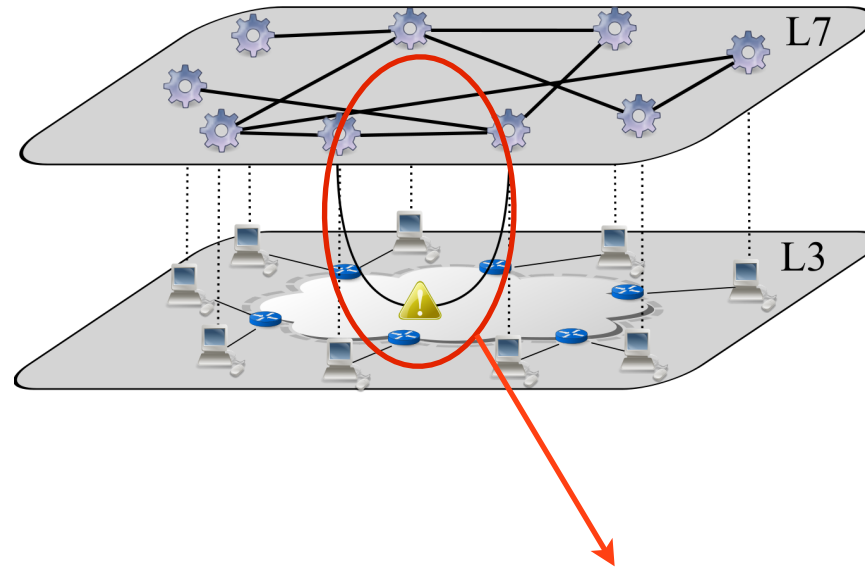
Layer 7 components



L7 Peer and Overlay

- Chunk Schedulers
 - $\{ru/r, lu/r, lu/la, lu/ba, lu/pa\}$
- Topologies:
 - $G(n, M), G(n, p)$
 - Topology Manager

Layer 7 components



L7 Peer and Overlay

- Chunk Schedulers
 - $\{ru/r, lu/r, lu/la, lu/ba, lu/pa\}$
- Topologies:
 - $G(n, M)$, $G(n, p)$
 - Topology Manager



L3/L7 Interaction

- ~~Text~~ **Signaling errors**
- Latency/capacity estimation errors

Simulation settings

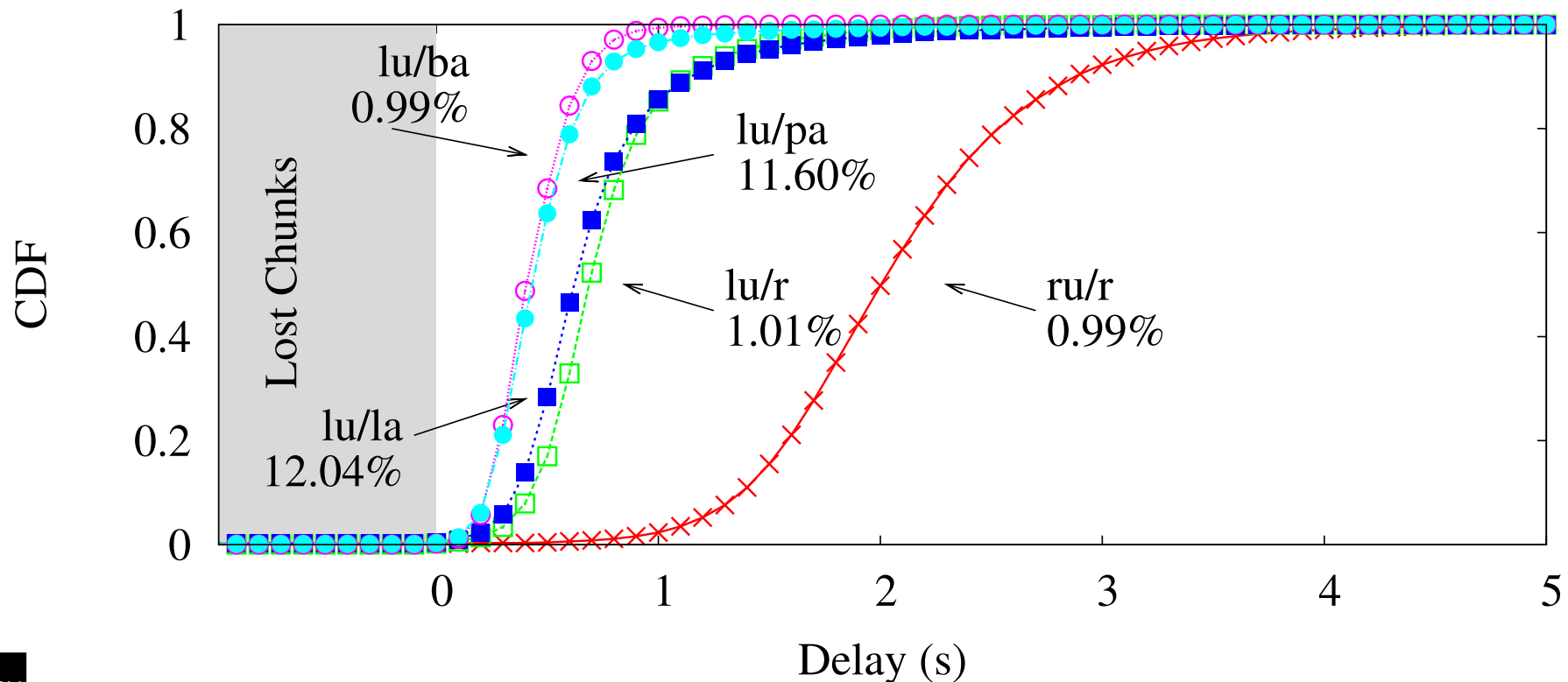
- 2000 peers
- Mean *out-degree* = 10
- 1500 fixed size chunks (100kbit) for a duration of 150 seconds
- Video stream at 1Mbps
- Experiment averaged on 10 simulation run with different overlays (GNR, GNP) and random seeds
- First 500 chunks ignored to avoid initial transient

Performance indexes

- Overlay performance indexes
 - Global **chunk delay** (CDF, 99 percentile,...)
 - **Loss rate** (chunks lost, or arrived after deadline)
- User perceived quality
 - **PSNR**
- Operator-wise link utilization
 - **Locality P%**: Amount of traffic exchanged at the access

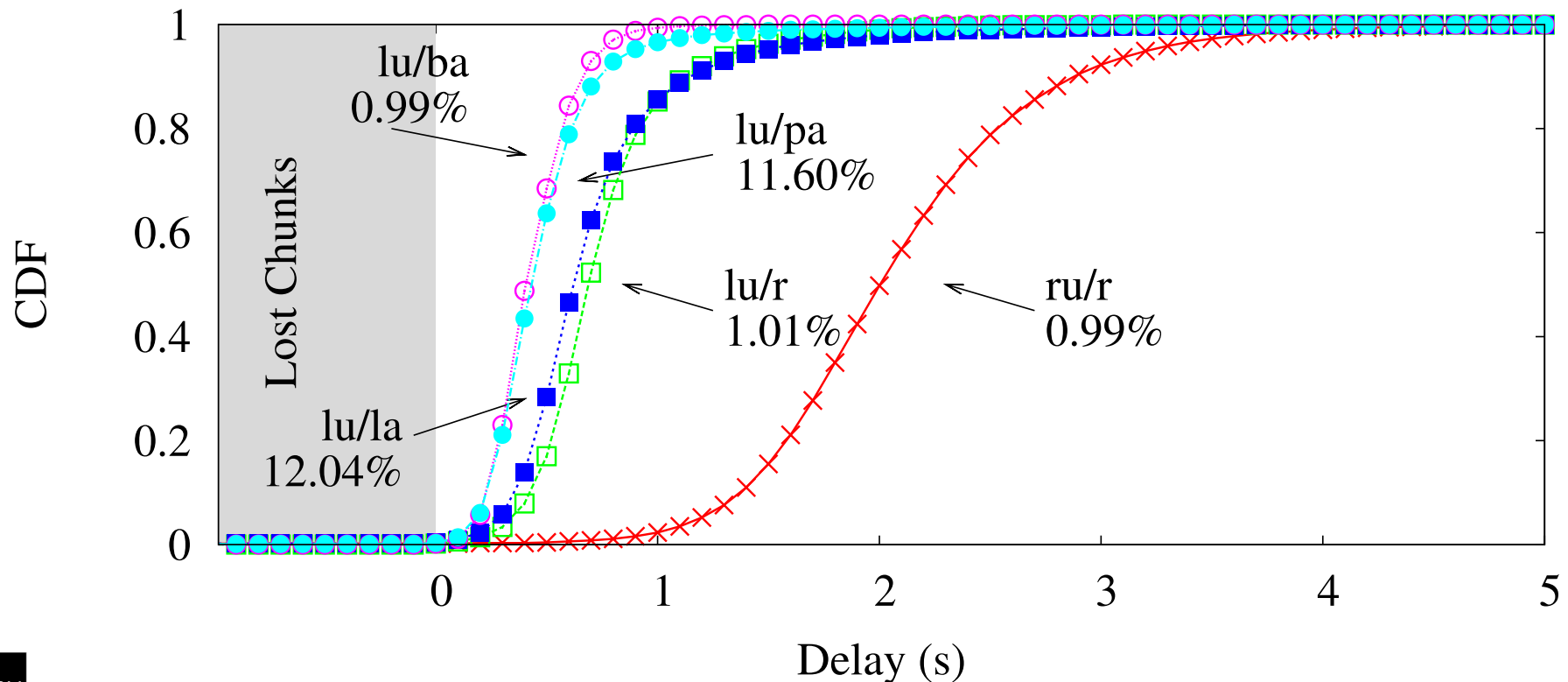
Algorithms comparison

- We compare **delay distributions** of 5 well-known chunk diffusion algorithms
 - *ru/r*, *lu/r*, *lu/la*, *lu/ba*, *lu/pa*
- Along with traffic **Locality** index P%



Algorithms comparison

- We compare **Latency Aware** distributions of 5 well-known chunk replication algorithms
 - ru/r , lu/r , lu/la , lu/ba , lu/pa
- Along with traffic **Locality** index P%

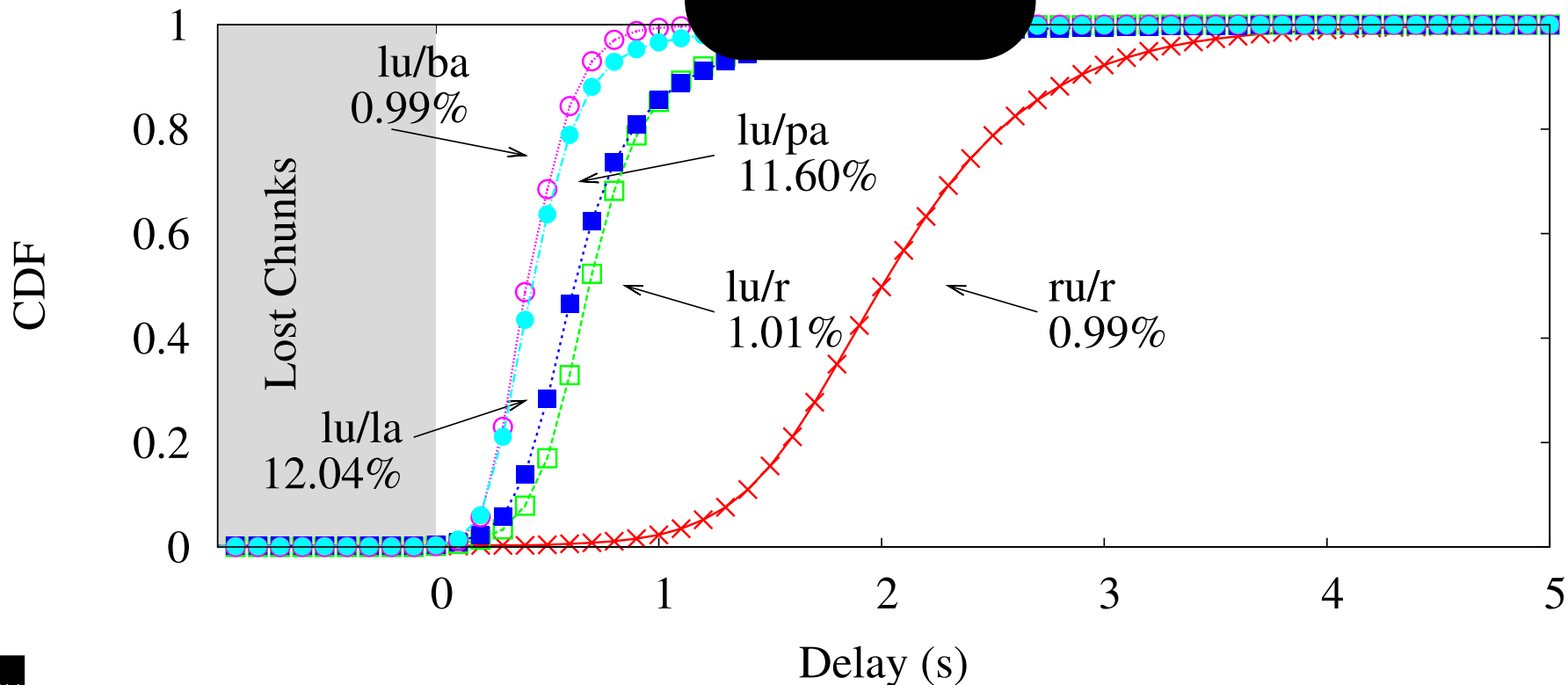


Algorithms comparison

- We compare **Latency Aware** distributions of 5 well-known chunk distribution algorithms

- *ru/r, lu/r, lu/la, lu/ba, lu/pa*

- Along with traffic Loss Index P%

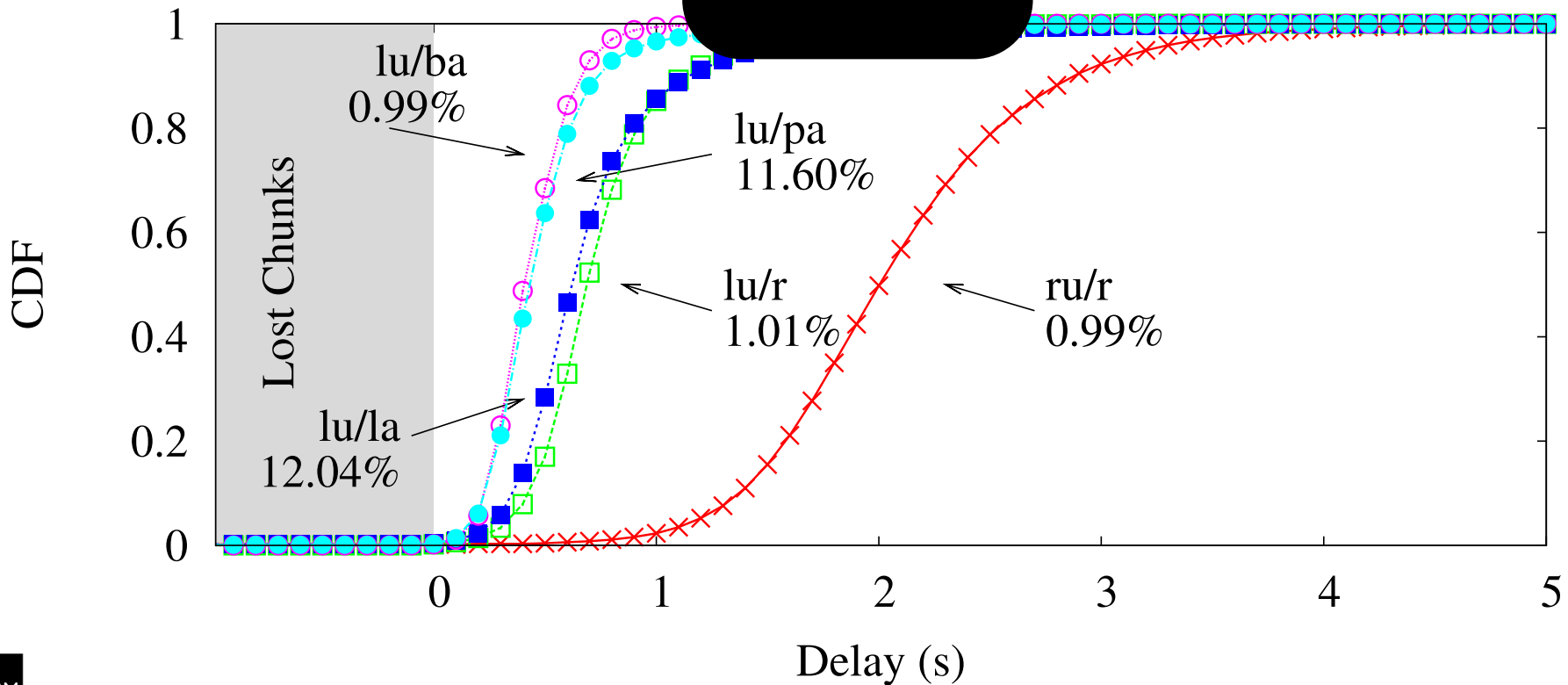
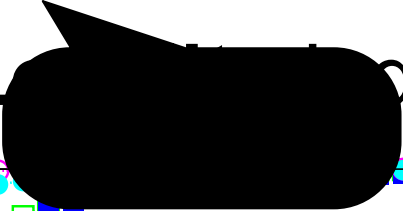


Algorithms comparison

- We compare known chunk distribution algorithms
 - *ru/r*, *lu/r*, *lu/la*, *lu/ba*, *lu/pa*
- Along with traffic Loss Index P%

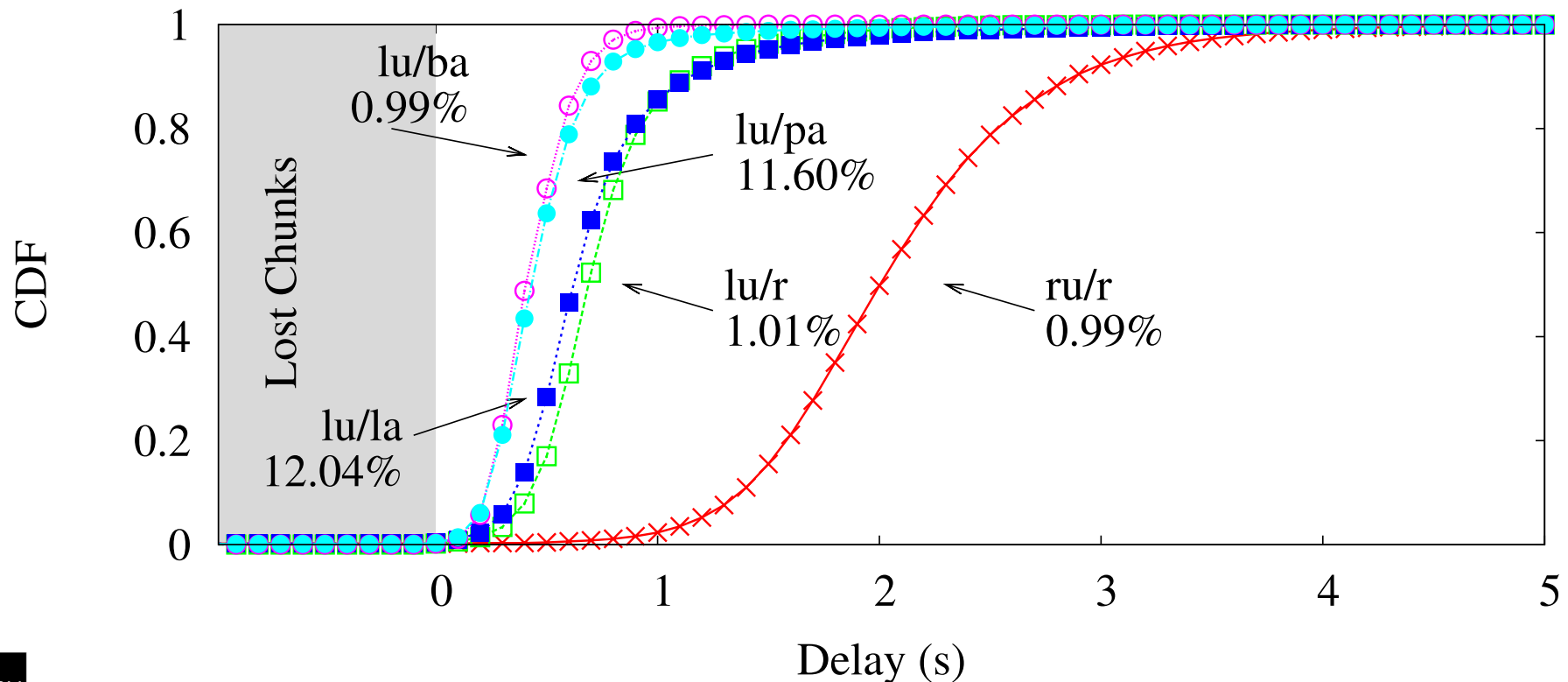
Latency Aware

Power Aware
BW/Latency



Algorithms comparison

- We compare **delay distributions** of 5 well-known chunk diffusion algorithms
 - *ru/r*, *lu/r*, *lu/la*, *lu/ba*, *lu/pa*
- Along with traffic **Locality** index P%



L3/L7 Interaction

- All schedulers base their decision on chunk “usefulness”.
- Each peer must have an up-to-date knowledge of neighborhood buffermaps.
- Control information can be lost at L3 (if control information is sent over UDP).
- P2P-TV applications may wish to limit signaling traffic lowering the rate of control information.
- **These factors induce a distorted view of the system state.**



Chunk collisions

- We want to assess the impact of signaling errors without being bound to a specific signaling algorithm.
- We resort to an high-level abstraction
 - we model errors due to loss or out-of date information as error on buffer-maps
- **With a probability P_{err} , a peer send a chunk to a neighbor despite it already received it.**



Perfect knowledge



Out-of-date knowledge

Chunk collisions

- We want to assess the impact of signaling errors without being bound to a specific signaling algorithm.
- We resort to an high-level abstraction
 - we model errors due to loss or out-of date information as error on buffer-maps
- **With a probability P_{err} , a peer send a chunk to a neighbor despite it already received it.**



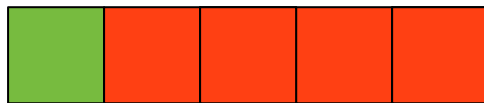
Perfect knowledge



Out-of-date knowledge

Chunk collisions

- We want to assess the impact of signaling errors without being bound to a specific signaling algorithm.
- We resort to an high-level abstraction
 - we model errors due to loss or out-of date information as error on buffer-maps
- **With a probability P_{err} , a peer send a chunk to a neighbor despite it already received it.**



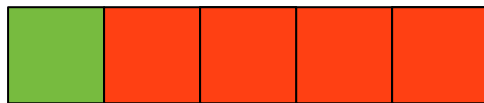
Perfect knowledge



Out-of-date knowledge

Chunk collisions

- We want to assess the impact of signaling errors without being bound to a specific signaling algorithm.
- We resort to an high-level abstraction
 - we model errors due to loss or out-of date information as error on buffer-maps
- **With a probability P_{err} , a peer send a chunk to a neighbor despite it already received it.**



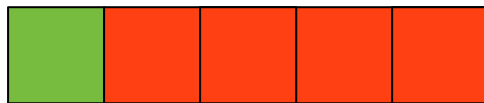
Perfect knowledge



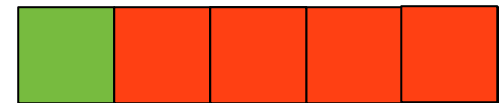
Out-of-date knowledge

Chunk collisions

- We want to assess the impact of signaling errors without being bound to a specific signaling algorithm.
- We resort to an high-level abstraction
 - we model errors due to loss or out-of date information as error on buffer-maps
- **With a probability P_{err} , a peer send a chunk to a neighbor despite it already received it.**



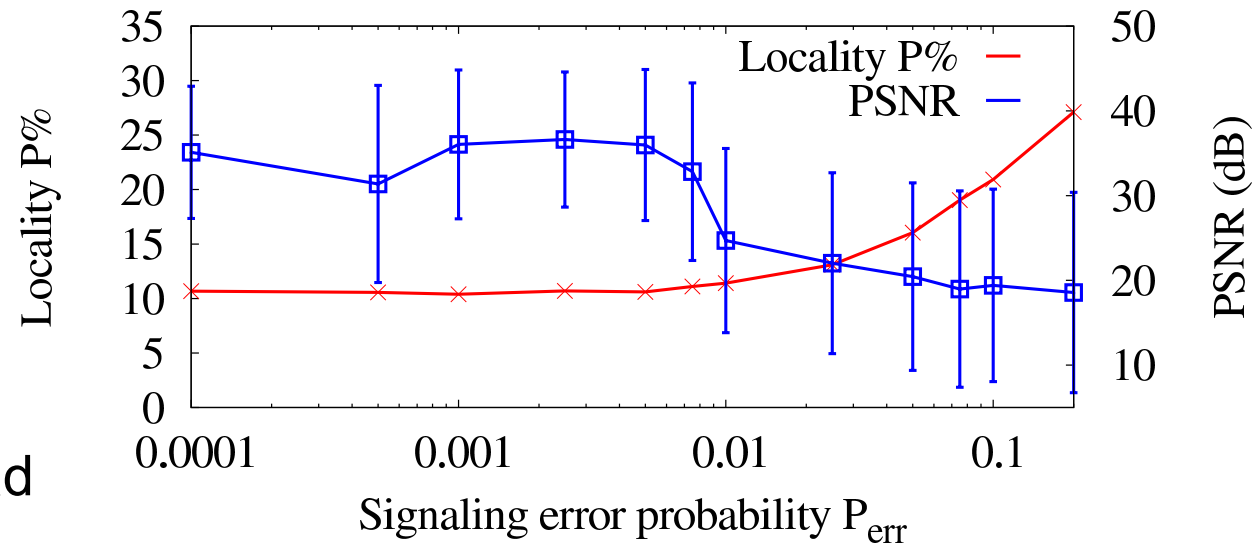
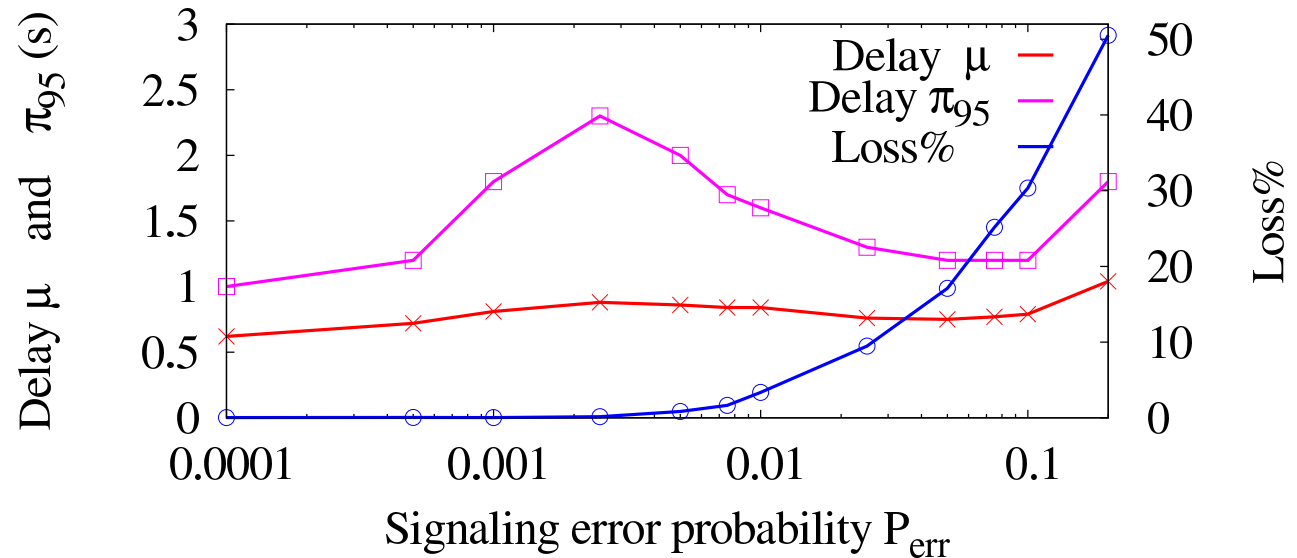
Perfect knowledge



Out-of-date knowledge

Signaling errors

- Global overlay properties
 - delay μ and π_{95}
 - loss%
- {User,operator}-related quality indexes
 - Locality P%
 - PSNR
 - EvalVid
 - log scale
 - stratified sampling
 - 28dB is the good/bad quality threshold



Wrap-up & Future work

- We want to compare these algorithms under **realistic settings** such as network latencies and heterogeneous peer classes.
- **Signaling errors** can *heavily impact* overlay performance.
- Future work:
 - Improve signaling mechanism
 - Consider other sources of errors
 - More sophisticated core network
 - Next-generation video codec

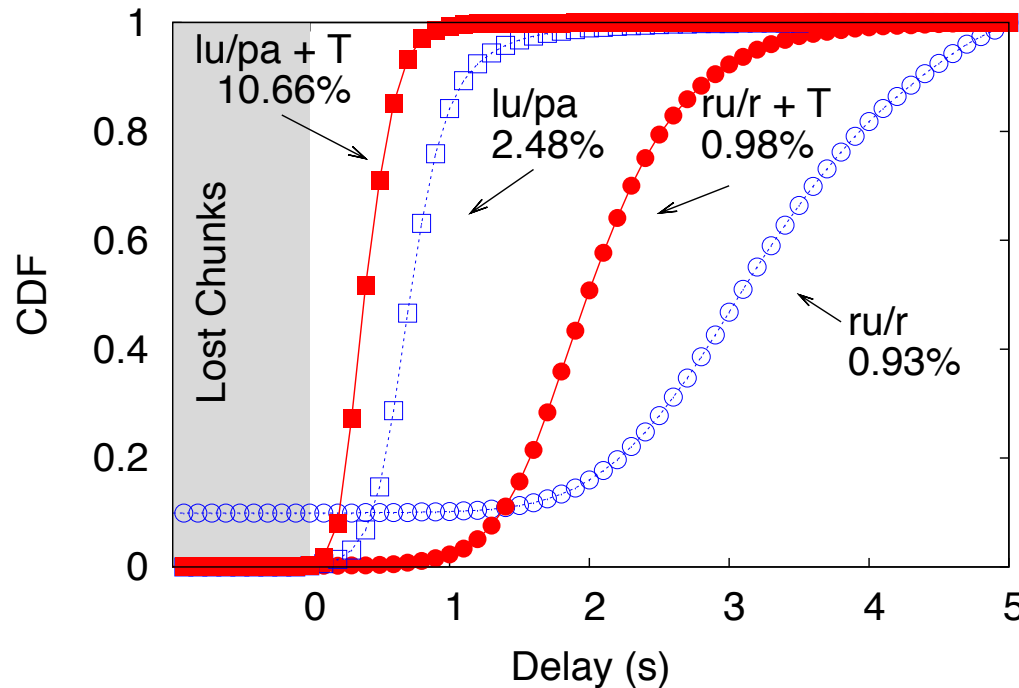
Signaling errors (cntd)

class	Ideal			Realistic			μ	π_{95}	L
	μ	π_{95}	L	μ	π_{95}	L			
1	0,41	0,6	2e-3	0,8	1,3	0,2	2,0	2,2	92,2
2	0,59	1	1e-2	0,8	1,1	4,4	1,4	1,1	263,9
3	0,69	1	1e-2	0,8	1,2	8,5	1,2	1,2	1421,3
4	0,92	1,3	0,2	1,1	1,4	33,9	1,2	1,1	198,6
FF	2,24	2,17	68,2	1,3	1,1	146,9			

- Breakdown of performance indexes among peer classes
- Intra-class degradation factor (i.e. $\mu_{ideal}/\mu_{realistic}$)
 - performance loss from ideal to realistic settings.
- Inter-class degradation factor (i.e. μ_1/μ_4)
 - quantifies the fairness of the results among classes

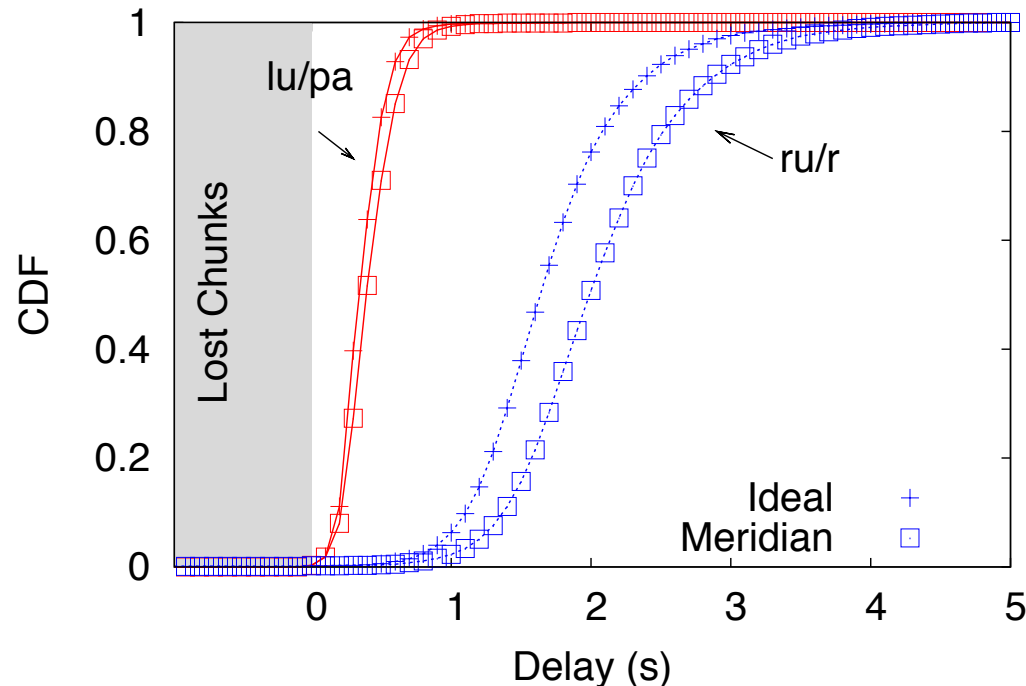
Topology Manager impact

- Subsets of algorithms (best - worst)
 - $ru/r, lu/pa$
- Topology manager improves performance as well as traffic localization!



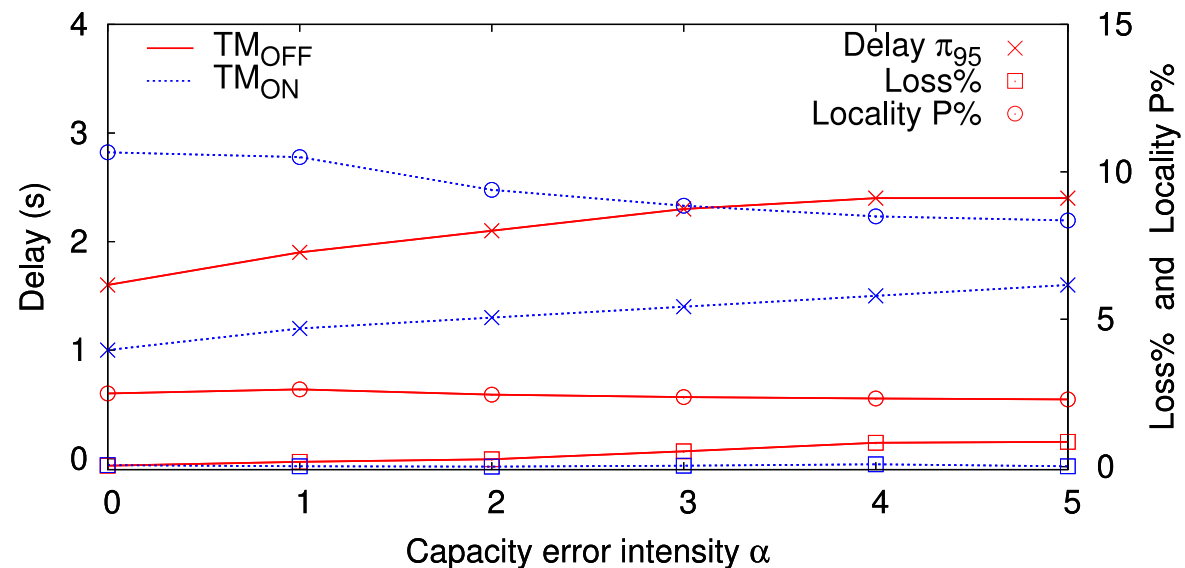
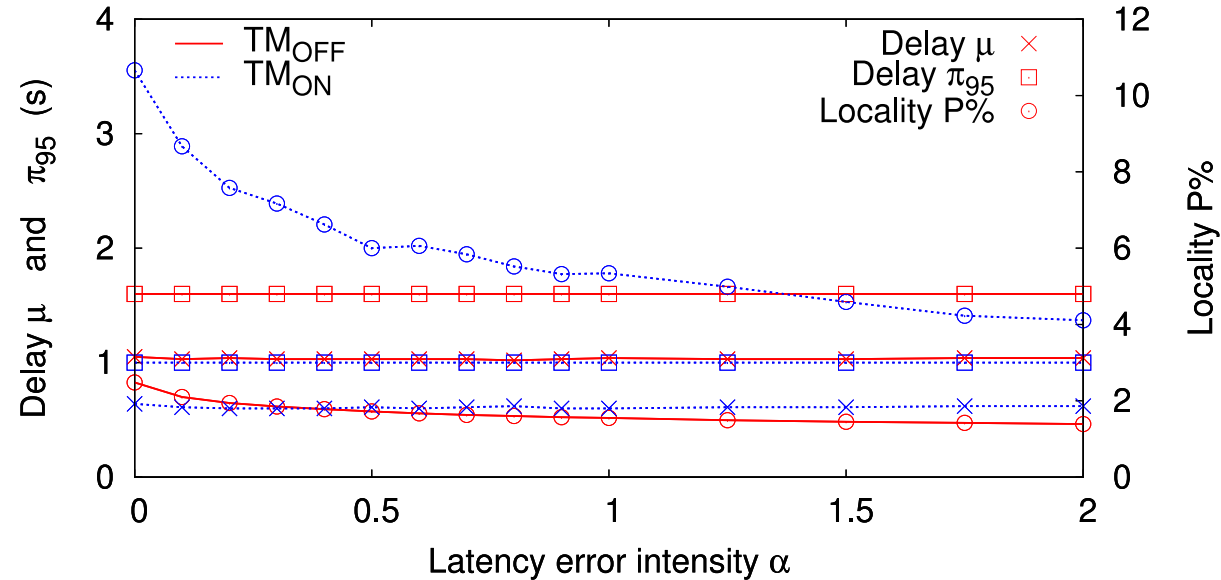
Network topologies impact

- Algorithms:
 - ru/r, lu/pa
- Reaction to different network topologies (i.e., latencies between router follow pre defined distributions)
 - Ideal, Constant, Meridian, Dynamic

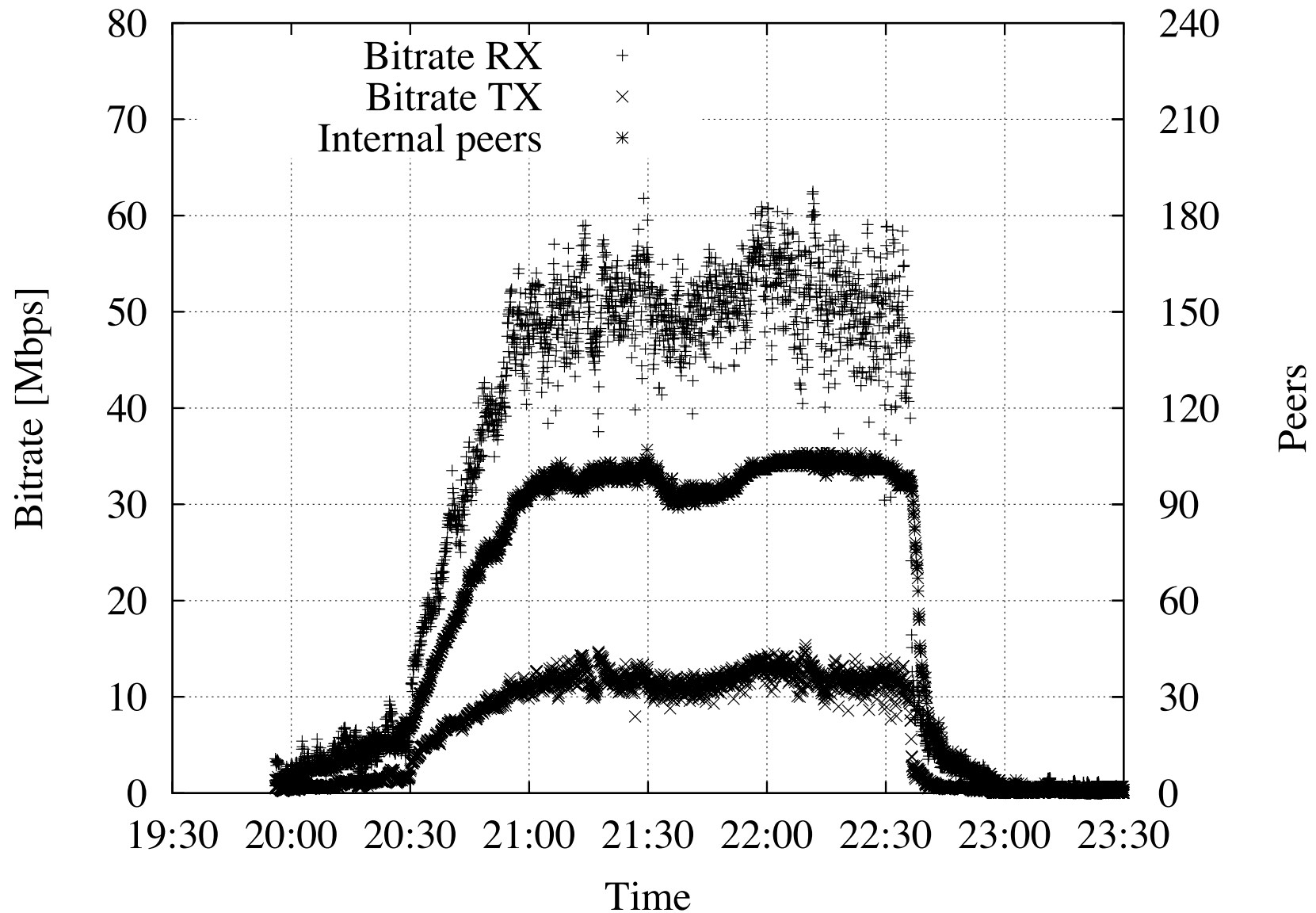


Measurements errors

- Latency estimation error
 - delays are little altered
 - traffic locality gets penalized
- Capacity estimation error
 - global delay increase



Churning



PSNR

- **Peak Signal-to-Noise Ratio**
- Logarithmic scale
- Measures the quality of the received video
- *EvalVid* tool used to evaluate PSNR
- Computed from lost chunk list
- Calculating PSNR for each peer is **CPU intensive!**
- We make a sampling among the peer and we use the mean value

```
Lost Chunks: 487
Mean delay: 1,28s
PSNR: 12,1dB
```

```
Lost Chunks: 45
Mean delay: 1,18s
PSNR: 16,1dB
```