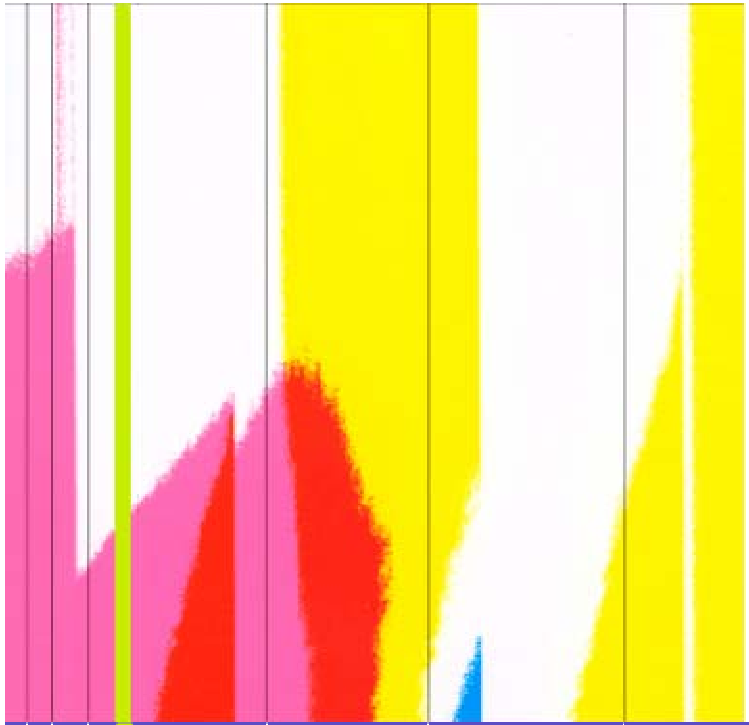


Fairness issues when transferring large volume of data on high speed networks with router-assisted transport protocols

11/5/2007, Anchorage, Alaska

Laurent Lefèvre - INRIA RESO – LIP (UMR CNRS, ENS, INRIA, UCB), France
Laurent.Lefevre@ens-lyon.fr

Dino M. López P. INRIA RESO – LIP (UMR CNRS, ENS, INRIA, UCB), France
Congduc Pham LIUPPA – Université de Pau, France

Introduction

We want to transport large volume of data in managed network infrastructures (High Speed Grids)

Need of a transport protocol :

- aggressive
- reactive

Using router assisted transport protocol (XCP)

2 cases :

- adding TCP flows among already transported XCP flows
- inserting XCP flows among TCP concurrent streams

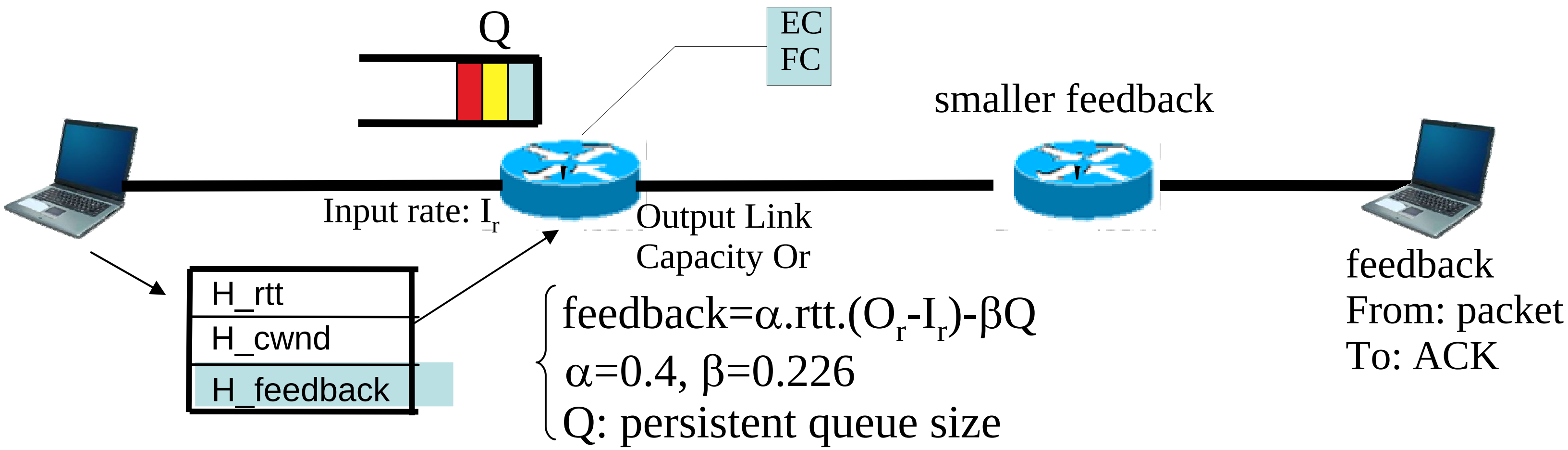
Context : long live flows, (not Internet)

The eXplicit Control Protocol XCP)

40 ans
la révolution de l'information

INRIA
RHÔNE-ALPES

XCP: eXplicit Control Protocol [Katabi02]



Q

EC
FC

smaller feedback

Input rate: I_r

Output Link
Capacity Or

H_rtt

H_cwnd

H_feedback

$feedback=\alpha.rtt.(O_r-I_r)-\beta Q$

$\alpha=0.4,\ \beta=0.226$

Q: persistent queue size

feedback
From: packet
To: ACK

Approach based on the routers assistance.

- Generalizes ECN.

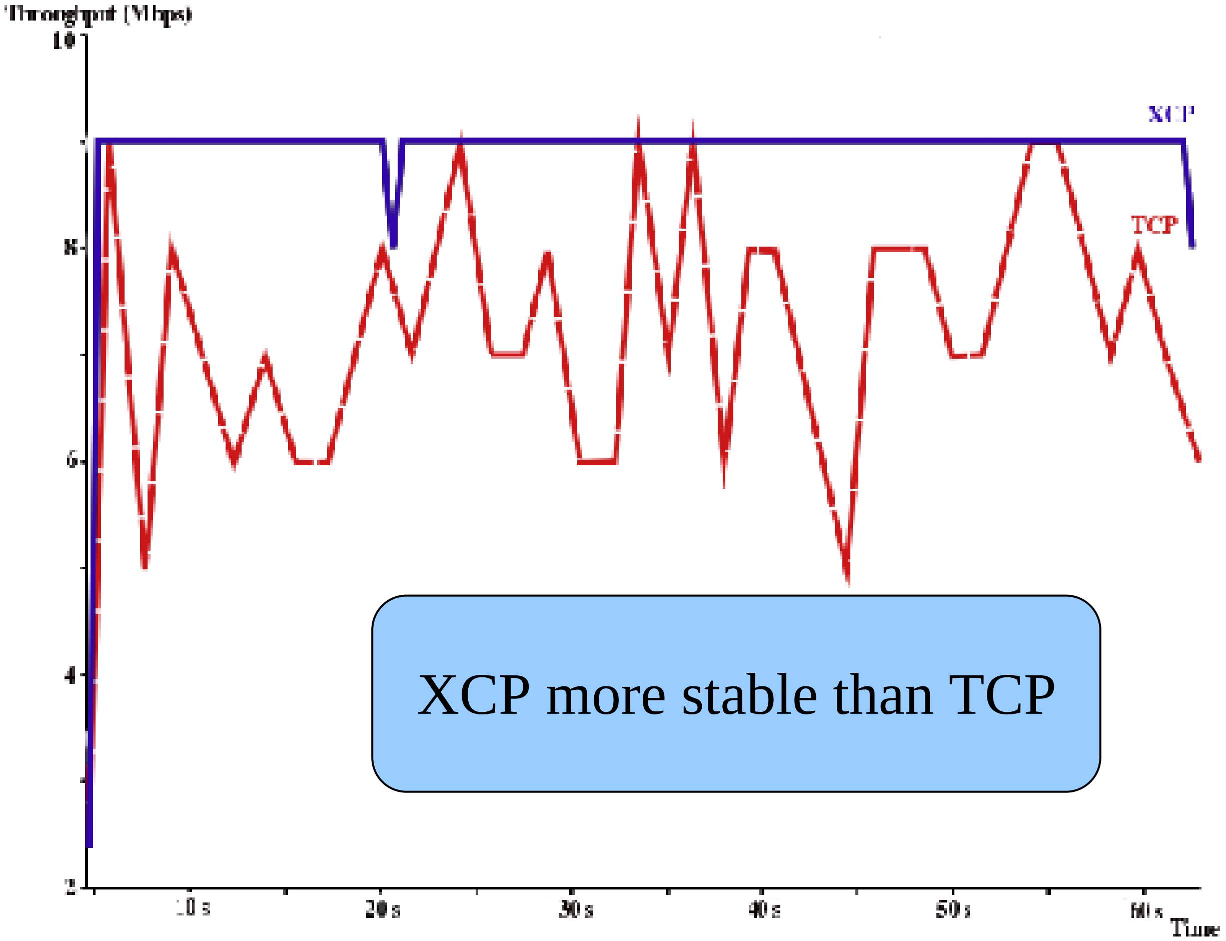
XCP router computes the available bandwidth (feedback)

- By monitoring the input traffic rate, the output link capacity and the persistent queue size.

The feedback is sent in the ACK to the sender

TCP and XCP (10Mbps – Random losses)

Real XCP and TCP non concurrent tests



XCP more stable than TCP

[Master Report. Anne-Cecile Orgerie. ENS Lyon]

Limits of XCP (& others router-assisted approaches)

Losses of ACK packets = Loss of network state information.

No interoperability between equipments

- Bad performance when it is used in combination with IP routers.

No interoperability between protocols.

- Low fairness when the resources are shared with end-to-end protocols

Limits of XCP (& almost all the assisted routers approaches)

Losses of ACK packets = Loss of network state information.

No interoperability between equipments

- Bad performance when it is used in combination with IP routers.

No interoperability between protocols.

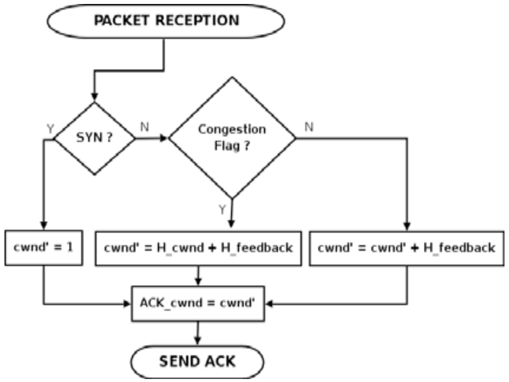
- Low fairness when the resources are shared with end-to-end protocols

Impossible to think about an incremental deployment of XCP over currents networks!!

XCP-r : Sender protocol stack off-load to avoid important feedback losses

In assisted routers protocols, the senders always need the collaboration of the receivers:

- Transfer feedback from data to ACK packets.



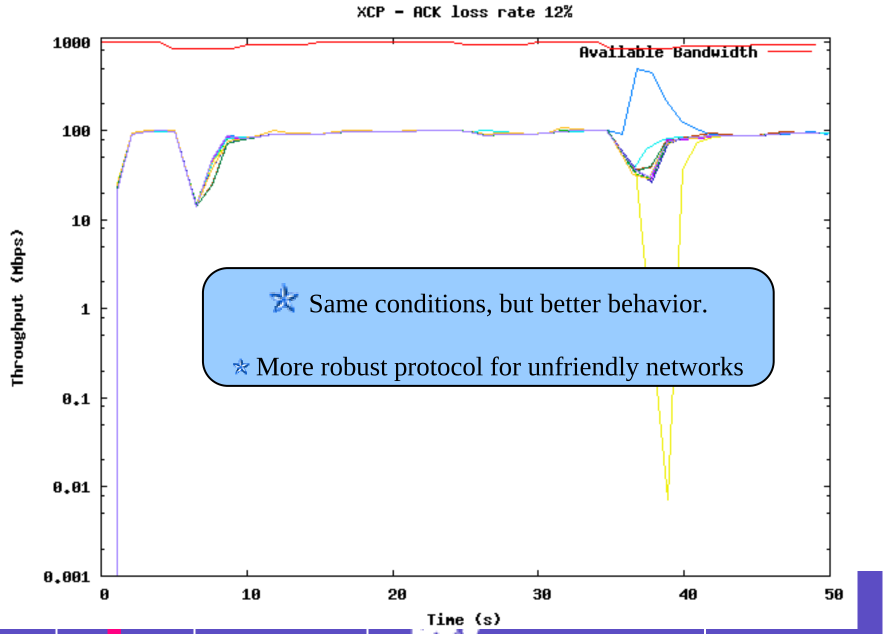
New algorithm in the receiver side

To avoid loss of information (feedback contained in ACK) the receiver can compute the appropriate *cwnd* value for the sender.

We install the protocol in the receiver *(XCP-r)*
`[Dino Lopez & Congduc Pham. ICN 2006]`

- For every connection accepted a *cwnd mirror (cwnd')* must be created by the receiver

- If the sender decides to modify the congestion window size arbitrary (e.g. after a congestion problem), the sender must notify this change to the receiver



- Same conditions, but better behavior.
- More robust protocol for unfriendly networks

Towards interoperability of XCP

XCP -r

~~Losses of ACK packets = Loss of network state information.~~

No interoperability between equipments

- Bad performance when it is used in combination with IP routers.

No interoperability between protocols.

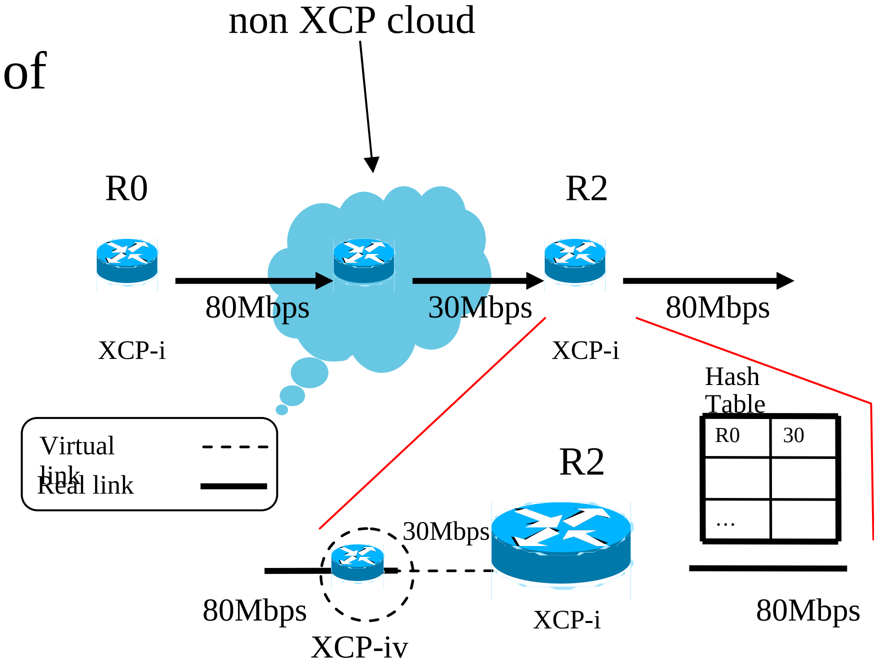
- Low fairness when the resources are shared with end-to-end protocols

XCP-i : Getting a new interoperable XCP protocol over XCP – IP networks

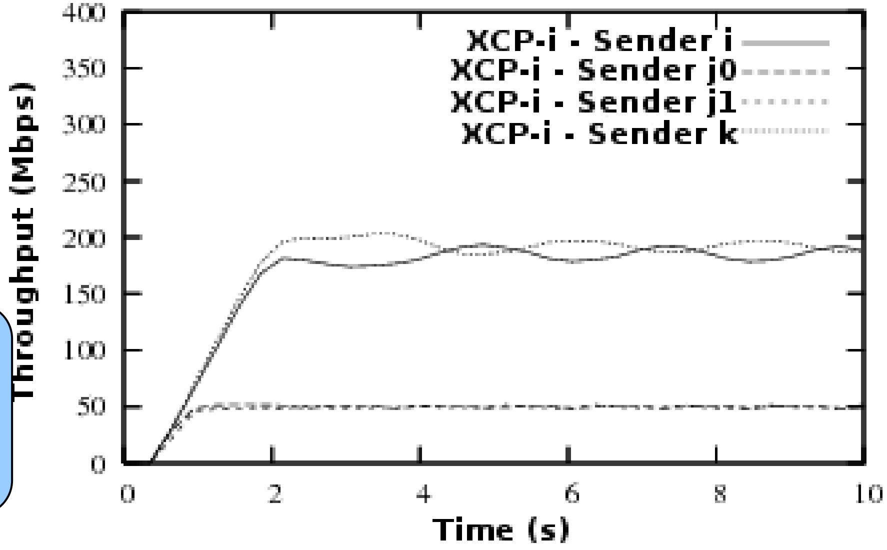
To solve this problem, we propose the addition of a new algorithm in the XCP routers :

- Keeping the XCP algorithm as in the original model.
- Reducing as much as possible the use of memory resources.

[D.Lopez, L.Lefèvre & C.Pham. Globecom 2006]



- *Detect the non-XCP clouds*
- *Estimate the ABW between RO and R2*
- *Replace every non XCP cloud by a XCP virtual router.*



Adapted performance in a not fully XCP network

Good fairness between XCP flows.

Good stability

Towards interoperability of XCP

~~Losses of ACK packets = Loss of network state information.~~

XCP-r

~~No interoperability between equipments~~

- ~~Bad performance when it is used in combination with IP routers.~~

XCP-i

No interoperability between protocols.

- Low fairness when the resources are shared with end-to-end protocols

Towards interoperability of XCP

XCP-r

~~Losses of ACK packets = Loss of network state information.~~

XCP-i

~~No interoperability between equipments~~

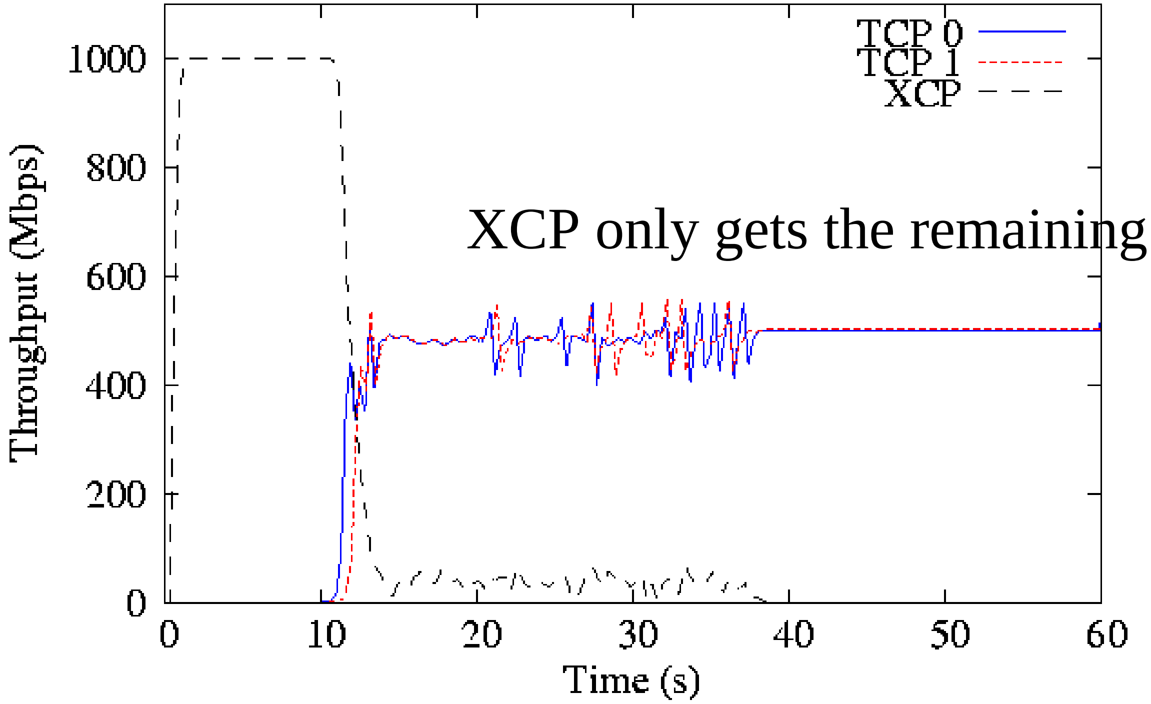
- ~~Bad performance when it is used in combination with IP routers.~~

No interoperability between protocols.

- Low fairness when the resources are shared with end-to-end protocols

Sharing the resources : XCP vs TCP

$$H\_feedback = \alpha.rtt.(O - I) - \beta.Q$$



XCP only gets the remaining bandwidth

Improving the fairness

Obtaining fairness on a link :

- Hard work!!
- Need to estimate the resources needed by XCP and non-XCP flows (TCP)
- Need to know the number of XCP and non-XCP active flows crossing a router.

Some approaches ;

- Record the ID of all flows crossing a router = Impossible
- Apply Bloom filters (e.g. NRED [Li-Su05]) = Hard in terms of processing. Accuracy in estimation (according to the authors)
- SRED-like mechanisms = Lightweight in terms of process time and used memory. Good accuracy ?

Estimation of the number of active XCP and non-XCP flows

We recycle the active flow estimation algorithm as described in SRED:

1. After filling (id flow) a Zombie list of 1000 packets

2. Each arriving packet p compared with randomly chosen zombie

If (!hit) with probability "r", overwrite the flow identifier of zombie with arrived packet.

3. Update the hit frequency estimator P(t)

$$P(t) = (1-\alpha)P(t-1) + \alpha \,.\, Hit$$

P(t)-1 is an estimation of the effective number of active flows

Resources needed by XCP flows

- After having an idea about the number of XCP & non-XCP flows, we can estimate the Bandwidth needed by XCP flows

XCP_BW = # XCP flows * Link_Capacity / (# XCP flows + # non-XCP flows)

If the estimation is reliable, how to improve fairness ?

- We have estimated the number of XCP and TCP flows. We DON'T know the exact number.

- To get a fairness, XCP routers must drop TCP packets with a certain probability.

- If our value of flows estimation is not accurated, this new probability should amortize the error.

Actions to execute after estimating XCP_BW

```
If (XCP input traffic rate > XCP_BW)
        Decrease the probability of dropping non-XCP packets
        Pdrop = Pdrop * Ddrop;
else if (XCP input traffic rate < XCP_BW)
                Increase the probability of dropping non-XCP packets
                Pdrop = Pdrop * Idrop;
        else
                Do nothing;
```

For TCP New Reno :

- $0.99 < Ddrop < 1$
- $1.01 > Idrop > 1$

Discussion / Limits

- We do not monitor the queue occupancy of the routers. We monitor the XCP_BW needed.

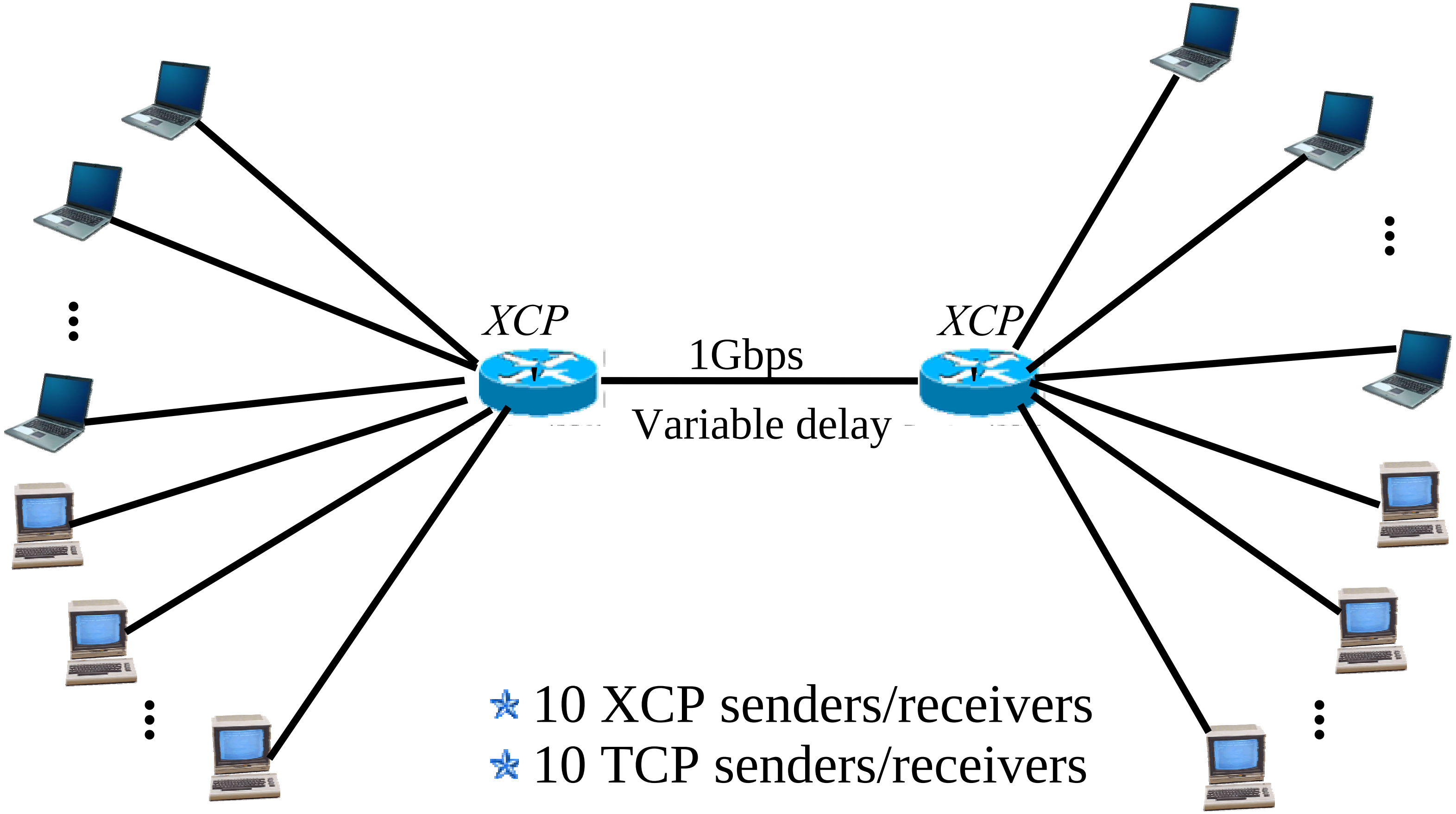
- This mechanism is executed only when XCP & non-XCP flows are detected by a XCP router.

- This mechanism is executed only when the the Total Input Traffic rate is bigger than 97% of the Output Link Capacity of a router.

- We never discard XCP packets.

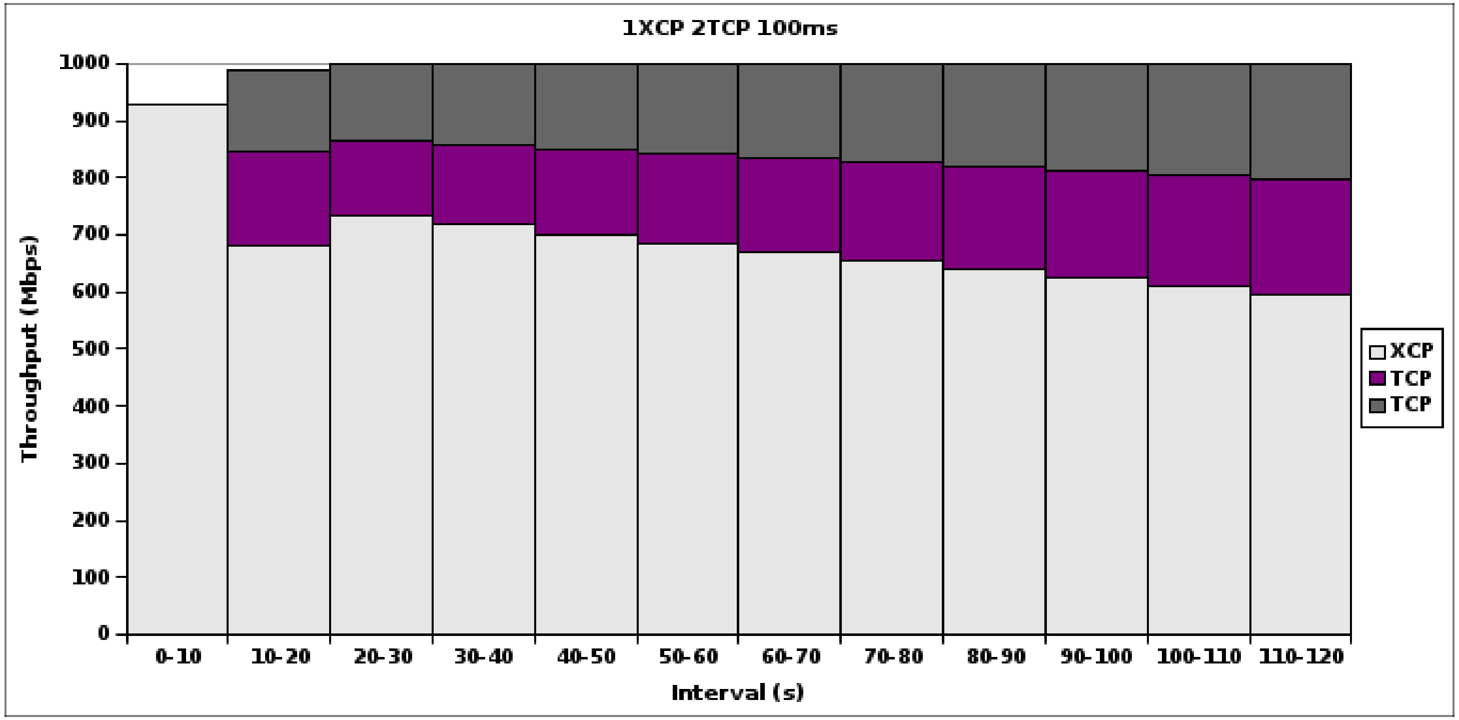
- The probability of dropping non-XCP packets is updated at every interval control of XCP (= average of all RTT values given by the XCP packets to the routers).

Topology for testing the XCP-TCP fairness mechanism



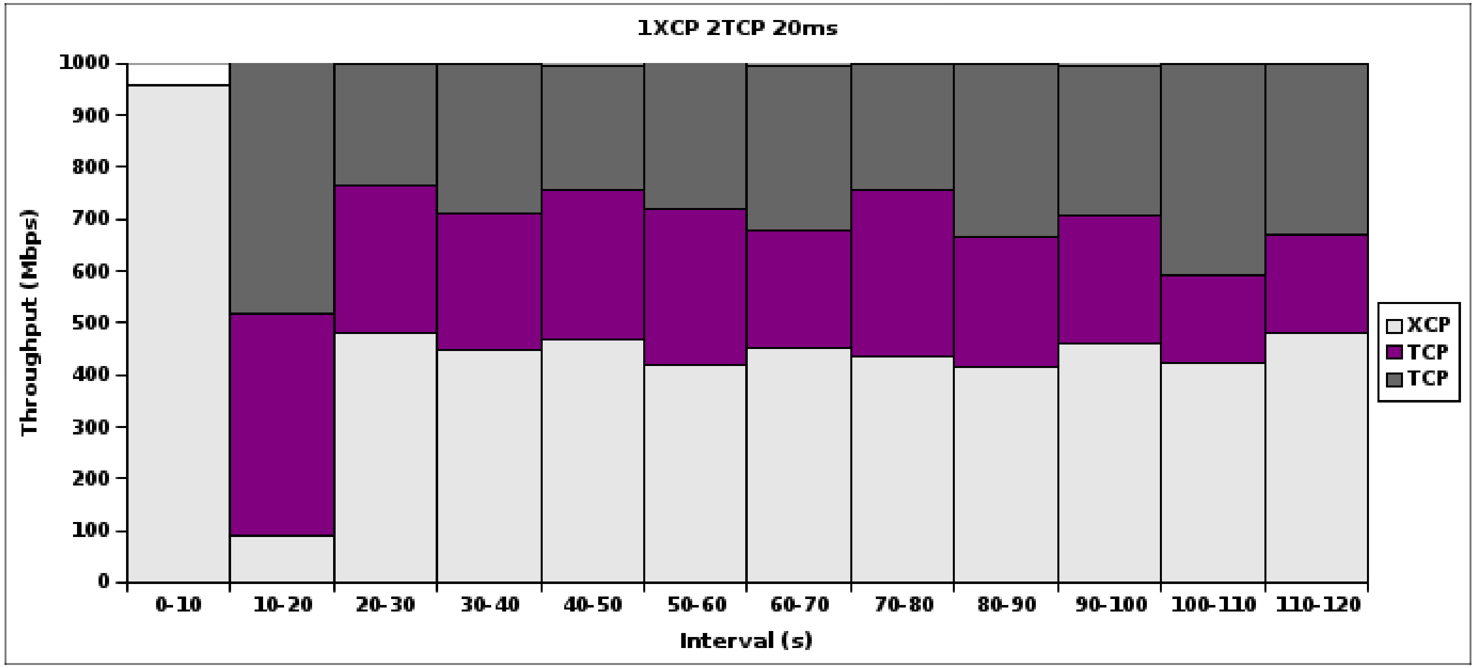
- 10 XCP senders/receivers
- 10 TCP senders/receivers

1XCP & 2TCP flows - 100ms of RTT



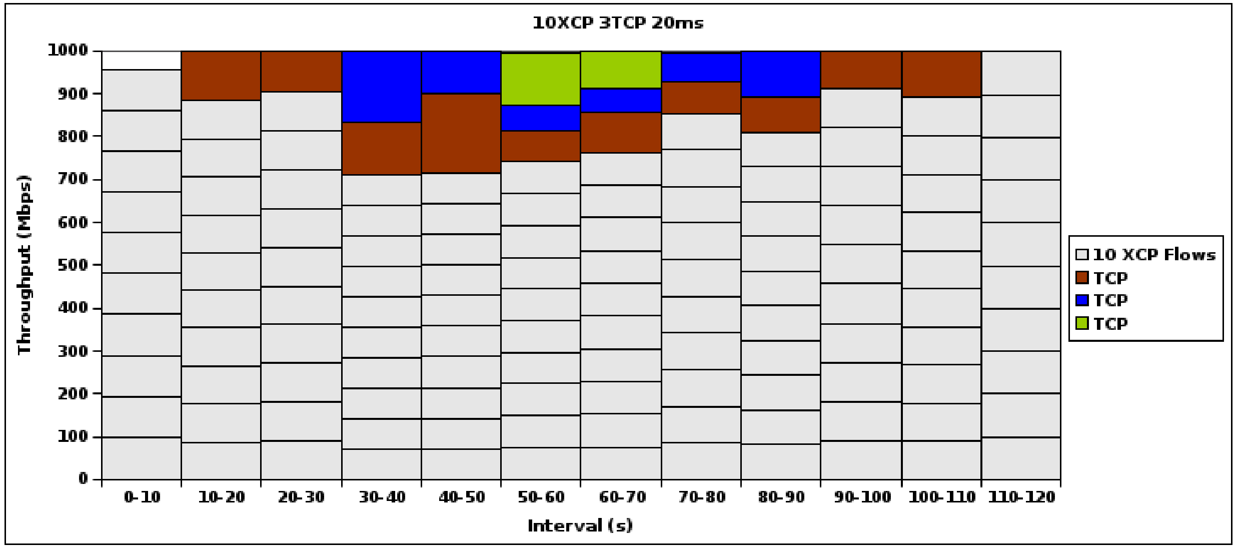
- XCP obtains (a little too much) bandwidth
- Drops at beginning of experiment
- Due too long RTTs, TCP flows not enough reactive

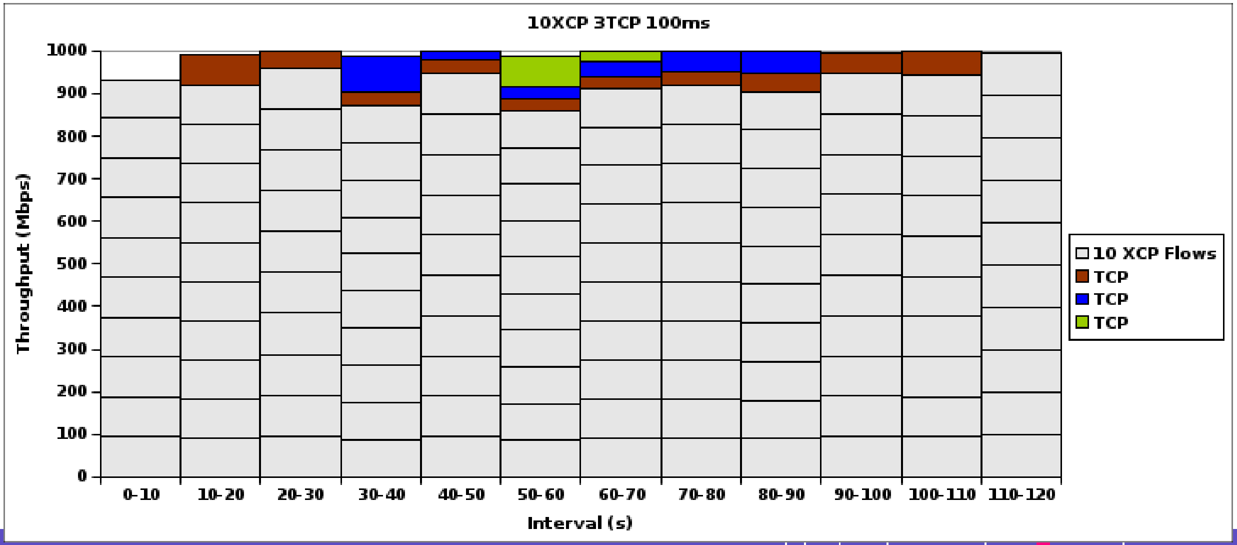
1XCP & 2TCP flows - 20ms of RTT



- We observe slow start effect
- TCP flows reactive : « good » fairness

10XCP & 3TCP flows – 20 and 100 ms



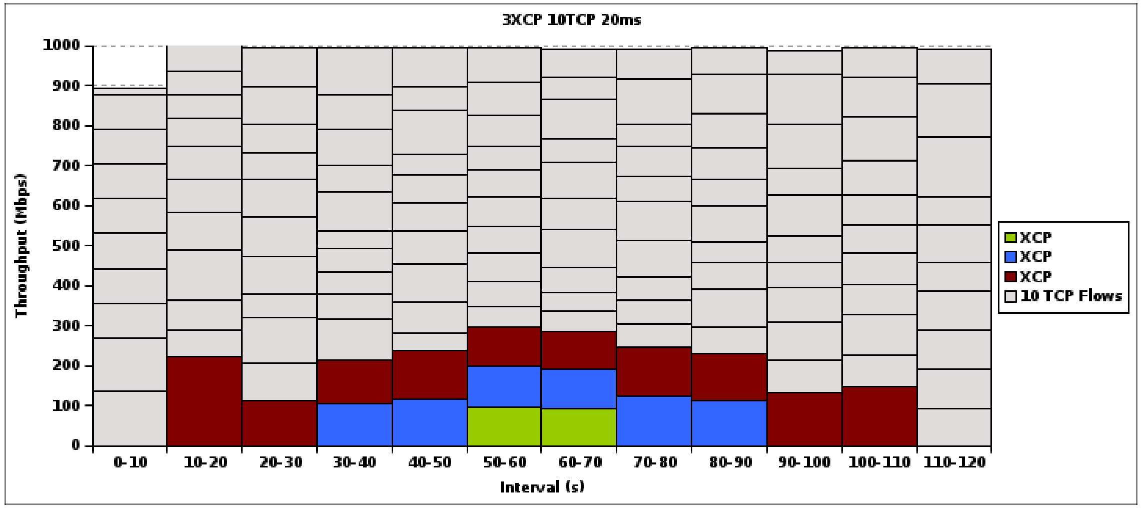


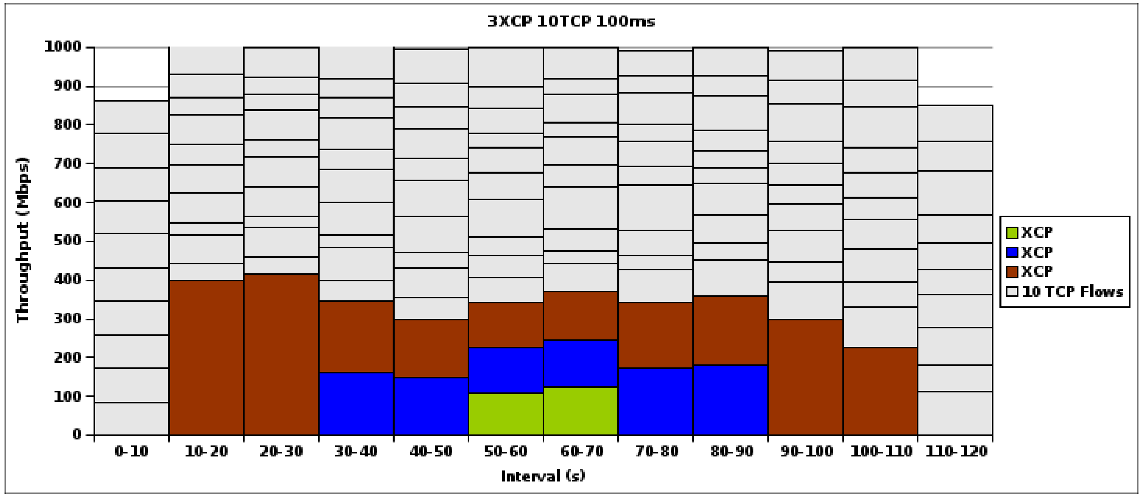
TCP flows arriving at sec 10, 30, 50

- •TCP slow start effect
- •Stability at sec 60

- •TCP not enough reactive to a drop (RTTs)
- • 3 slow start steps -> drop increase
- •XCP regains bandwidth when TCP flows leave

3XCP & 10TCP flows – 20 and 100 ms





XCP flows arriving at sec 10, 30, 50

- We observe good stability periods

- Need time to converge

Modifying the SRED-like mechanism

In all our studies, the zombie method for estimating the number of active flows analyzes all the incoming packets. This mechanism joined with the XCP controls can be expensive for the router.

We reduce the number of analysed packets :

- $P(t) = (1-\alpha)P(t-1) + \alpha .hit$: probability to find the *i* flow in the zombie table when we analyze 100% of incoming packets

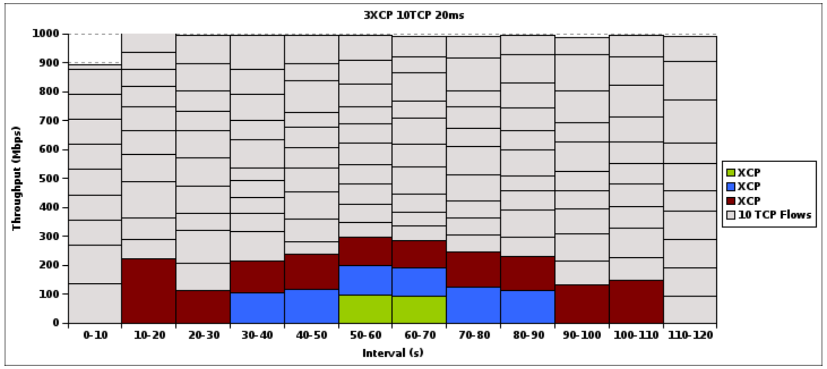
- $P(t) = (1- \alpha)P(t-1)Pa + \alpha .hit.Pa$: probability to find the *i* flow in the zombie table when we analyze the incoming packets, with a *Pa* probability.

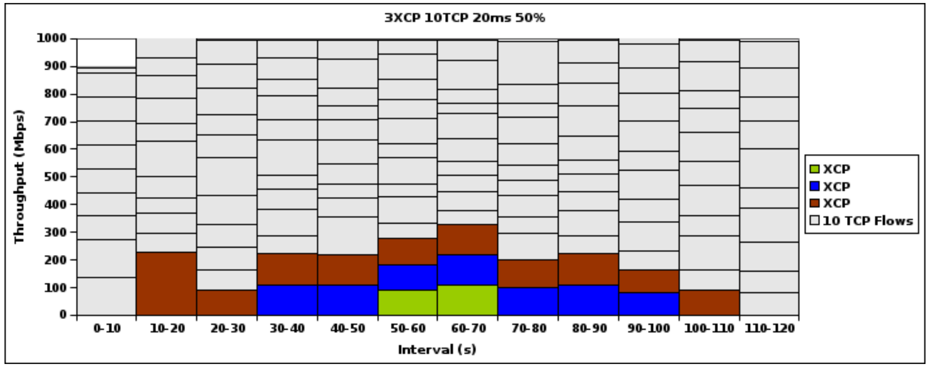
Reducing impact per router

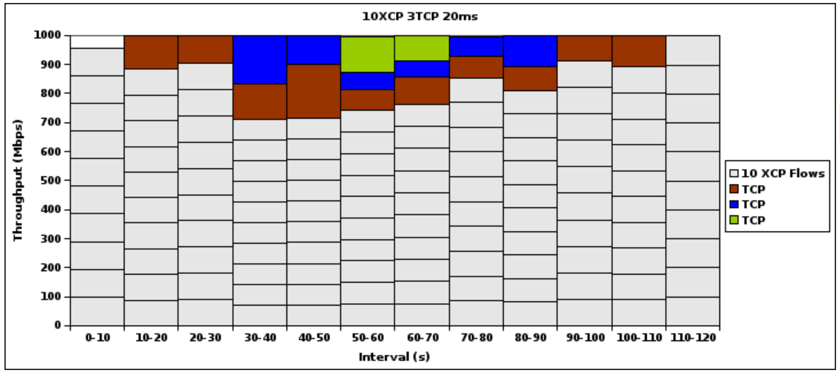
Avoiding the analysis of 100% packets

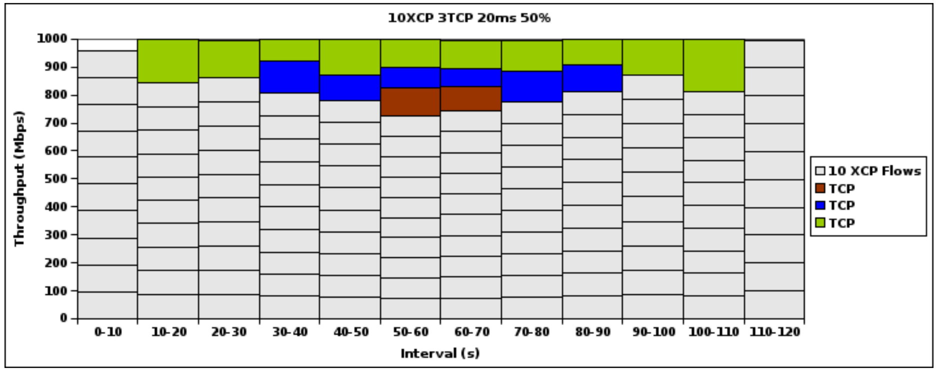
100% packets

50% packets









Conclusions and current works

- Last step for interoperability of XCP with external world (losses, heterogeneous equipments, heterogeneous protocols)

- Scalable and lightweight in terms of routers CPU / memory usage

- Appropriate for high bandwidth * delay product networks running long live flows.

- Limit of simulation tools (ns-2). We want to validate on real emulated XCP routers : Developing an XCP implementation for large scale validation (Grid5000 platform)

- More information on :

http://www.ens-lyon.fr/LIP/RESO/Projects/XCP

Questions ?

http://www.ens-lyon.fr/LIP/RESO/Projects/XCP

40 ans
la révolution de l'information

INRIA
RHÔNE-ALPES