XCP-i: eXplicit Control Protocol for heterogeneous inter-networking of high-speed networks

Dino LOPEZ [1] C. PHAM [2] L. LEFEVRE [3]

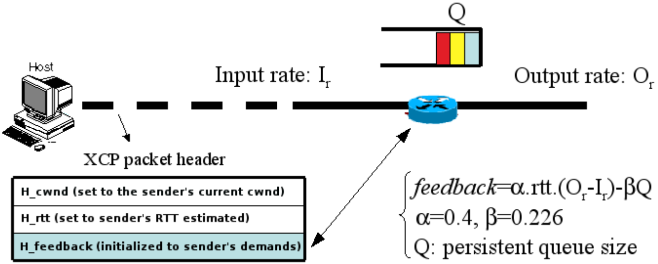
November 28th, 2006

[1]INRIA RESO/LIP, France dmlopezp@ens-lyon.fr
[2]LIUPPA, University of Pau, France Congduc.Pham@univ-pau.fr
[3]INRIA RESO/LIP, France Laurent.Lefevre@ens-lyon.fr

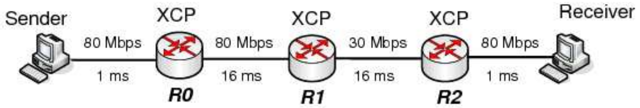
Index

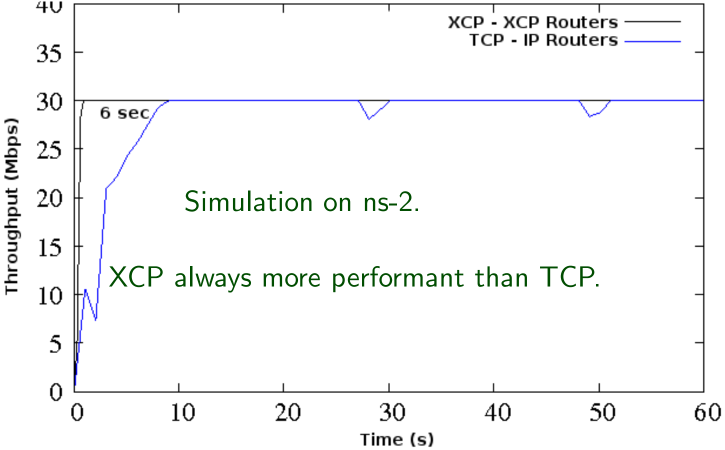
eXplicit Control Protocol [Katabi]

- Protocol based on the use of assisted routers (generalizes ECN).
- XCP Header: TCP Header + 3 new fields.
- The XCP routers update the *H_feedback*.
- The feedback is sent back to the sender in the ACK.
- The sender updates the $cwnd = cwnd + feedback$.



XCP vs TCP





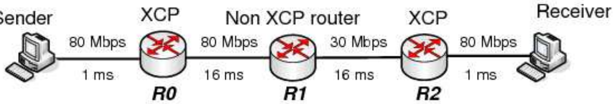
Simulation on ns-2.

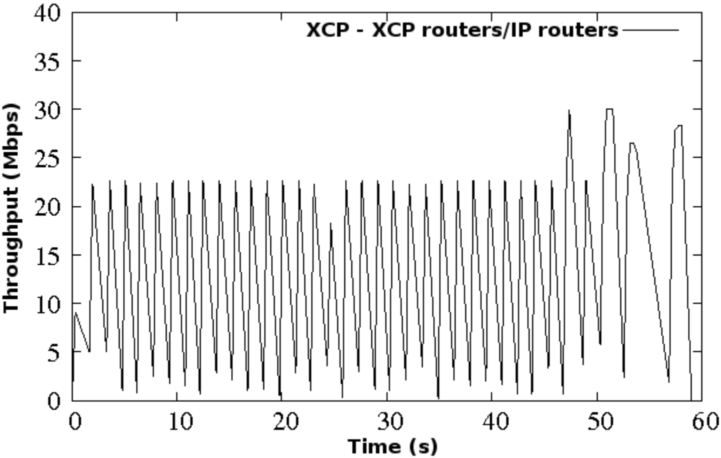
XCP always more performant than TCP.

But...

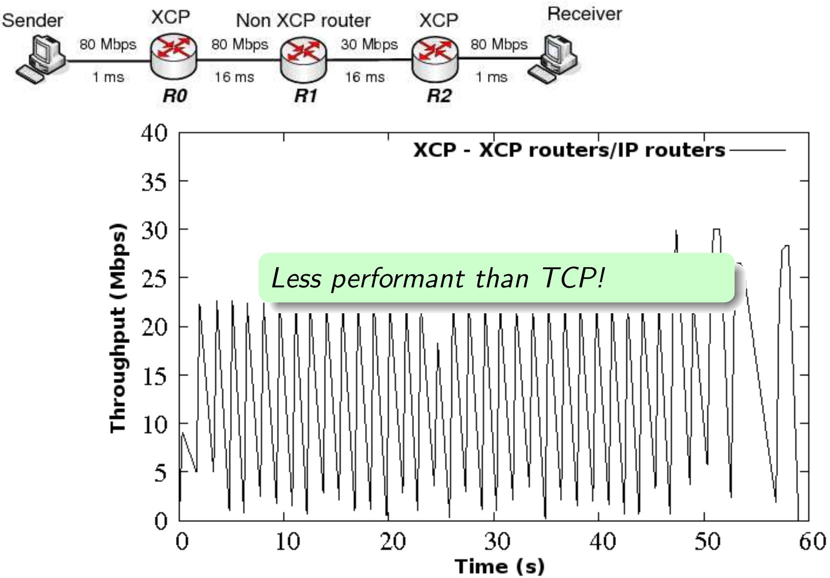
1. It works well only in 100% XCP networks.
2. No interoperabilty between equipments
   - Bad performance if classical IP routers are placed in the bottleneck.
3. No interoperability between protocols
   - Throughput really small when it shares the bottleneck with end-to-end protocols.
4. Imposible to think in a decremental deployment.

XCP Router - IP classical router





XCP Router - IP classical router



Less performant than TCP!

XCP-i: Approach for a new interoperable XCP version

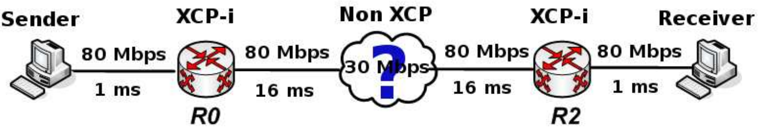
1. The absence of interoperability prevents the incremental deployment of XCP in the new networks.
2. Our approach enables to tackle the problem:
   - with no states per flow in the routers.
   - keeping the original control laws of XCP.

XCP-i: Approach for a new interoperable XCP version

1. The absence of interoperability prevents the incremental deployment of XCP in the new networks.
2. Our approach enables to tackle the problem:
   - with no states per flow in the routers.
   - keeping the original control laws of XCP.

1. Where are placed the non XCP routers?

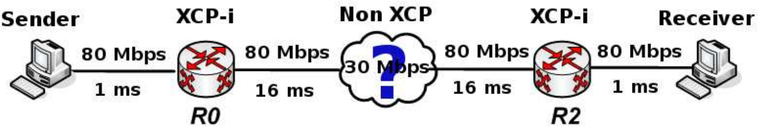
Detecting the non XCP clouds



`xcp_ttl_ = TTL`

- Discover the non XCP routers:
  - Use the *IP TTL* field.
  - New field *xcp_ttl_* in the XCP header.
  - Initialize o the same *TTL* value.
  - Compare fields *xcp_ttl_* and *TTL*.
  - Decrease *xcp_ttl_* in every XCP-i router.

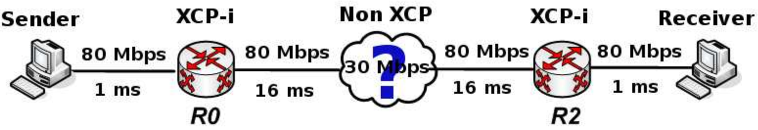
Detecting the non XCP clouds




```
xcp_ttl_ = TTL    xcp_ttl_ == TTL
                   xcp_ttl_--
                     TTL--
```

- Discover the non XCP routers:
  - Use the *IP TTL* field.
  - New field *xcp_ttl_* in the XCP header.
  - Initialize o the same *TTL* value.
  - Compare fields *xcp_ttl_* and *TTL*.
  - Decrease *xcp_ttl_* in every XCP-i router.

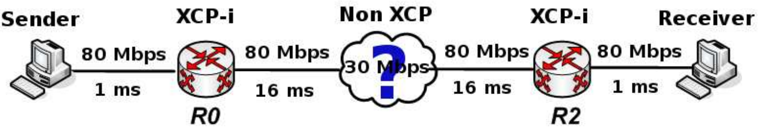
Detecting the non XCP clouds




```
xcp_ttl_ = TTL    xcp_ttl_ == TTL              xcp_ttl_ != TTL
                   xcp_ttl_--                     TTL--
                     TTL--                    xcp_ttl_ = TTL
```

- Discover the non XCP routers:
  - Use the *IP TTL* field.
  - New field *xcp_ttl_* in the XCP header.
  - Initialize o the same *TTL* value.
  - Compare fields *xcp_ttl_* and *TTL*.
  - Decrease *xcp_ttl_* in every XCP-i router.

Detecting the non XCP clouds



`xcp_ttl_ = TTL`

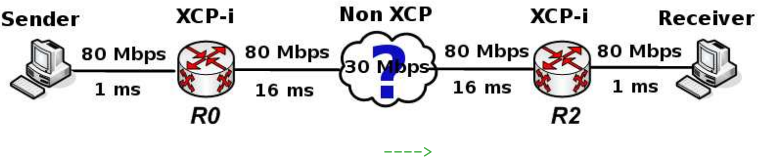
`xcp_ttl_ == TTL`
`xcp_ttl_--`
`TTL--`

`xcp_ttl_ != TTL`
`TTL--`
`xcp_ttl_ = TTL`

- Discover the non XCP routers:
  - Use the *IP TTL* field.
  - New field *xcp_ttl_* in the XCP header.
  - Initialize o the same *TTL* value.
  - Compare fields *xcp_ttl_* and *TTL*.
  - Decrease *xcp_ttl_* in every XCP-i router.

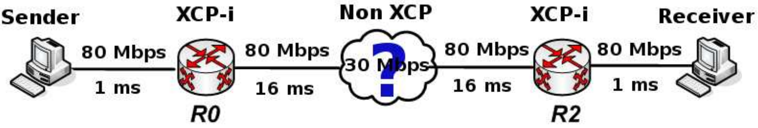
2. What is inside the non-XCP cloud?

Computing the resources in the non-XCP cloud



- Compute the state of the network in the non-XCP cloud (R0 -> R2).
  - Execute a processus to compute the available bandwidth in the non XCP cloud (Packet train, quickprobe).
- Discover the last XCP-i router (R0).
  - New field *last_xcp_router_* in the XCP header.
  - Update it with the IP address of the last sender node.

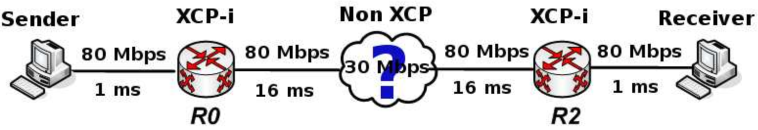
Computing the resources in the non-XCP cloud



last_xcp_router_ = S ---->

- Compute the state of the network in the non-XCP cloud (R0 -> R2).
  - Execute a processus to compute the available bandwidth in the non XCP cloud (Packet train, quickprobe).
- Discover the last XCP-i router (R0).
  - New field *last_xcp_router_* in the XCP header.
  - Update it with the IP address of the last sender node.

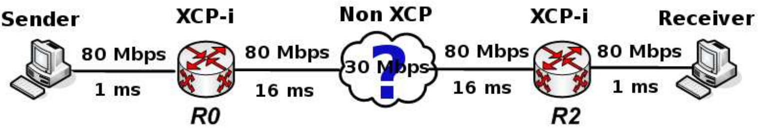
Computing the resources in the non-XCP cloud



last_xcp_router_ = S   S
                          R0
---->

- Compute the state of the network in the non-XCP cloud (R0 -> R2).
  - Execute a processus to compute the available bandwidth in the non XCP cloud (Packet train, quickprobe).
- Discover the last XCP-i router (R0).
  - New field *last_xcp_router_* in the XCP header.
  - Update it with the IP address of the last sender node.

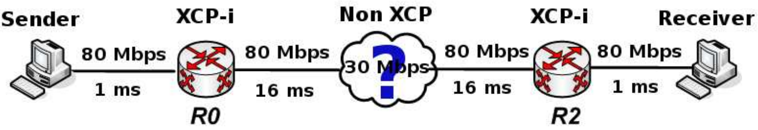
Computing the resources in the non-XCP cloud



`last_xcp_router_ = S`   `S`   `---->`   `R0`
`R0`   `R2`

- Compute the state of the network in the non-XCP cloud (R0 -> R2).
  - Execute a processus to compute the available bandwidth in the non XCP cloud (Packet train, quickprobe).
- Discover the last XCP-i router (R0).
  - New field *last_xcp_router_* in the XCP header.
  - Update it with the IP address of the last sender node.

Computing the resources in the non-XCP cloud

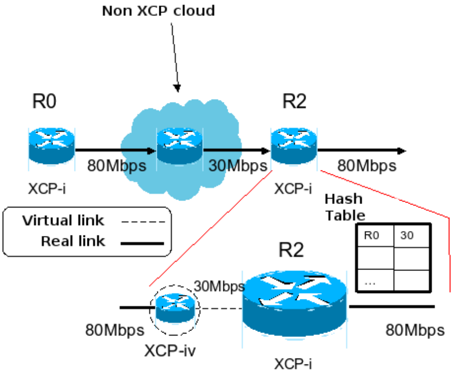


last_xcp_router_ = S   S   ---->   R0
R0   R2

- Compute the state of the network in the non-XCP cloud (R0 -> R2).
  - Execute a processus to compute the available bandwidth in the non XCP cloud (Packet train, quickprobe).
- Discover the last XCP-i router (R0).
  - New field *last_xcp_router_* in the XCP header.
  - Update it with the IP address of the last sender node.

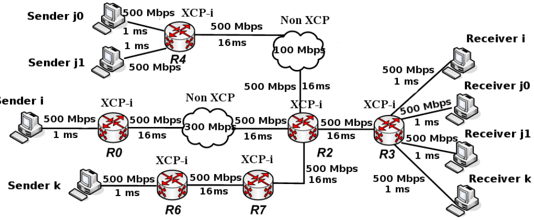
3. How to calculate a new feedback to take the ABW into account?

The Virtual XCP-i router

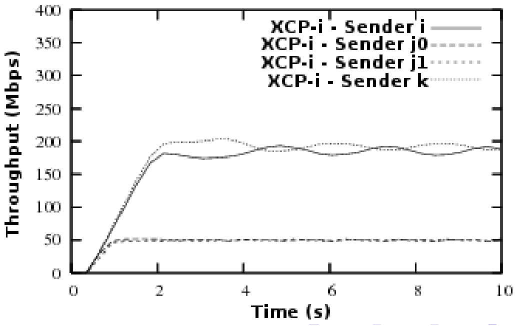


- Send the Available BW to the router which required it.
  - Hash table.
- Create a virtual router XCP-iv.
  - $f = \alpha.rtt.(O - I) - \beta.Q$
  - $f_v = \alpha.rtt.ABW - \beta.Q$
- Subtitute every non-XCP cloud by a virtual router.

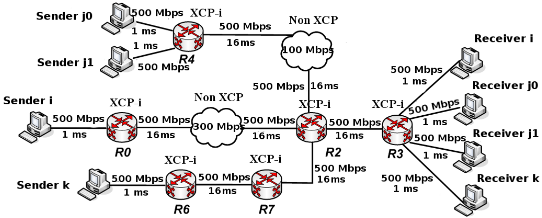
XCP-i in an heterogeneous network



- Simulation in ns-2.
- Adapted performance in a not fully XCP network.

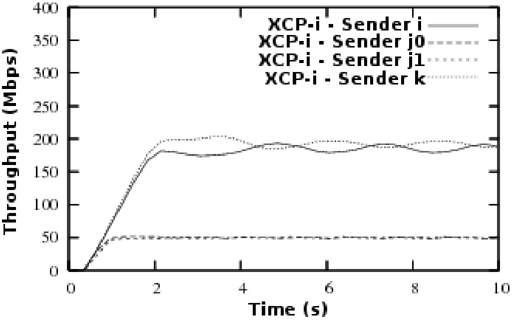


XCP-i in an heterogeneous network

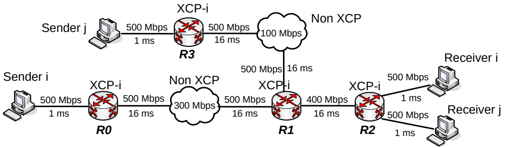


- Good fairness between flows.
- Flows stability.

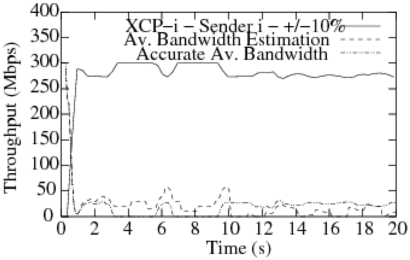
- Simulation in ns-2.
- Adapted performance in a not fully XCP network.

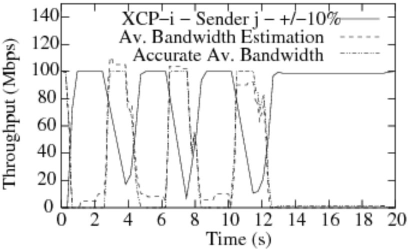


Sensibility to the bandwidth estimation accuracy



- Under estimation.
  - Under utilisation.
  - No timeouts.
- Over estimation.
  - Timeouts depend on the routers capacity.





Conclusion

- Approach for the interoperability between equipements in an heterogeneous network.
- XCP-i is the first step to an interoperable congestion control protocol based on assisted routers.
- XCP-i keeps the XCP controls laws as in the original model.
- XCP-i works in a large range of network topologies.

Future Works

- Implementation of XCP-i in a Linux kernel.
- Deployment and testing of XCP-i on a large scale (grid5000).
- Fairness with end-to-end protocols.

Discussion Time...!