Implementing Quality of Service for the Software Defined Networking Enabled Future Internet

EWSDN 2014
2nd September 2014

Sachin Sharma, Dimitri Staessens, Didier Colle, David Palma, Joao Goncalves, Ricardo Figueiredo, Donal Morris, Mario Pickavet, and Piet Demeester

Agenda of our talk

CityFlow

- Motivations
- Our QoS Framework for SDN
- Objective of our experiments
- Experimental Setup and Results
- Conclusions

epcc iMinds CONNECT.INNOVATE.CREATE ONESOURCE RedZinc SEVENTH FRAMEWORK PROGRAMME

Motivation

- Achieving high QoS is a major concern in the current Internet
  - Service Level Agreement between business customers and a Service Provider

- Current QoS mechanisms (IntServ & DiffServ) are practically not possible to implement globally over the Internet

Motivation

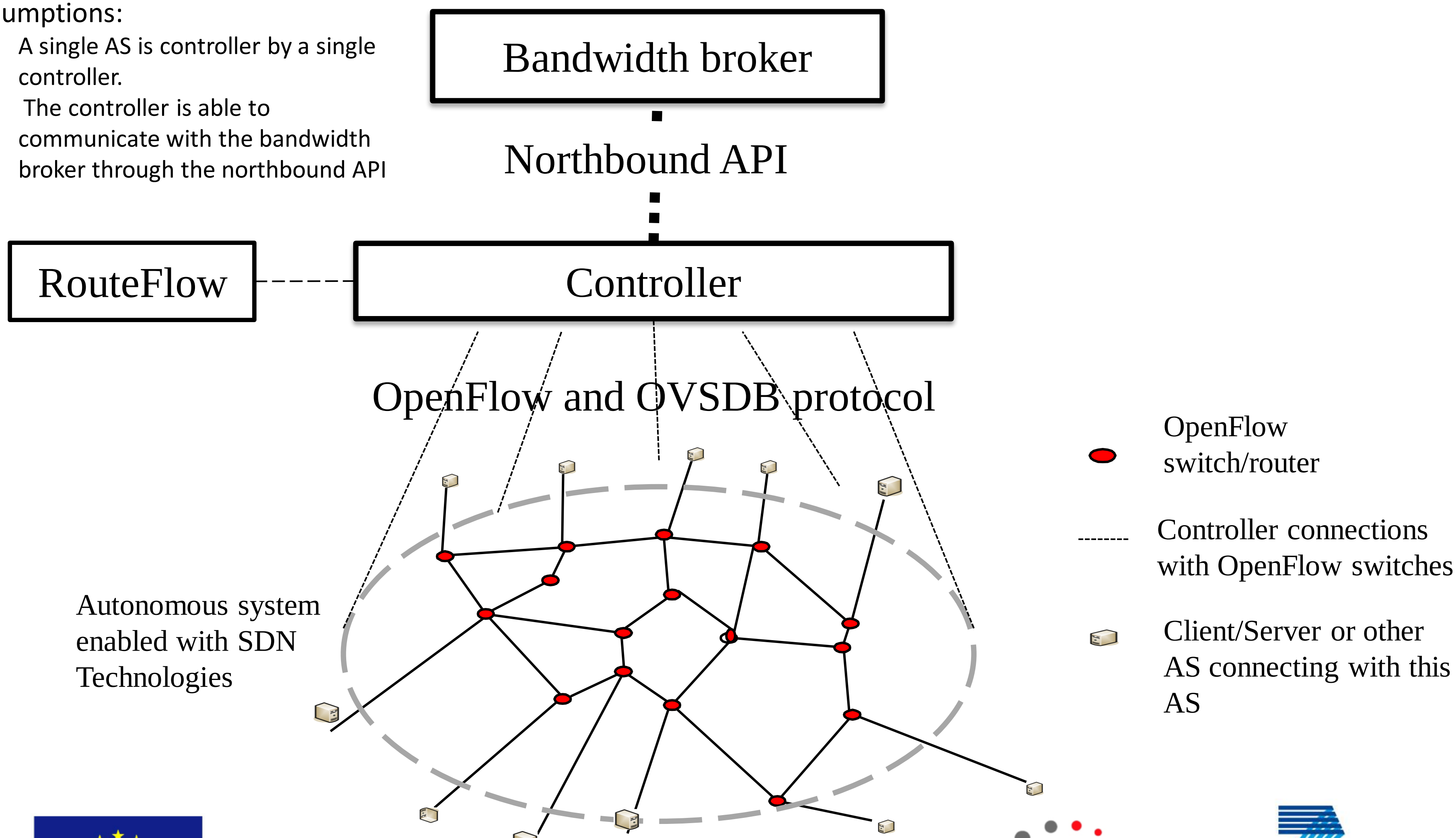
- Three tasks are needed to be performed in the current Internet for establishing QoS
  - Determining availability of network resources
    - Difficult to determine because each device runs its own control software
  - Configure network resources if they are available
    - No standard protocol available to configure resources
  - Reconfigure network resources on a failure conditions
    - Resilience protocols need to depend on vendor specific protocol for reconfiguration
- With SDN, all the above tasks are practically possible to implement
  - Centralized control
  - Standard OpenFlow, OF-CONFIG or OVSDB protocol for configuration and reconfiguration.

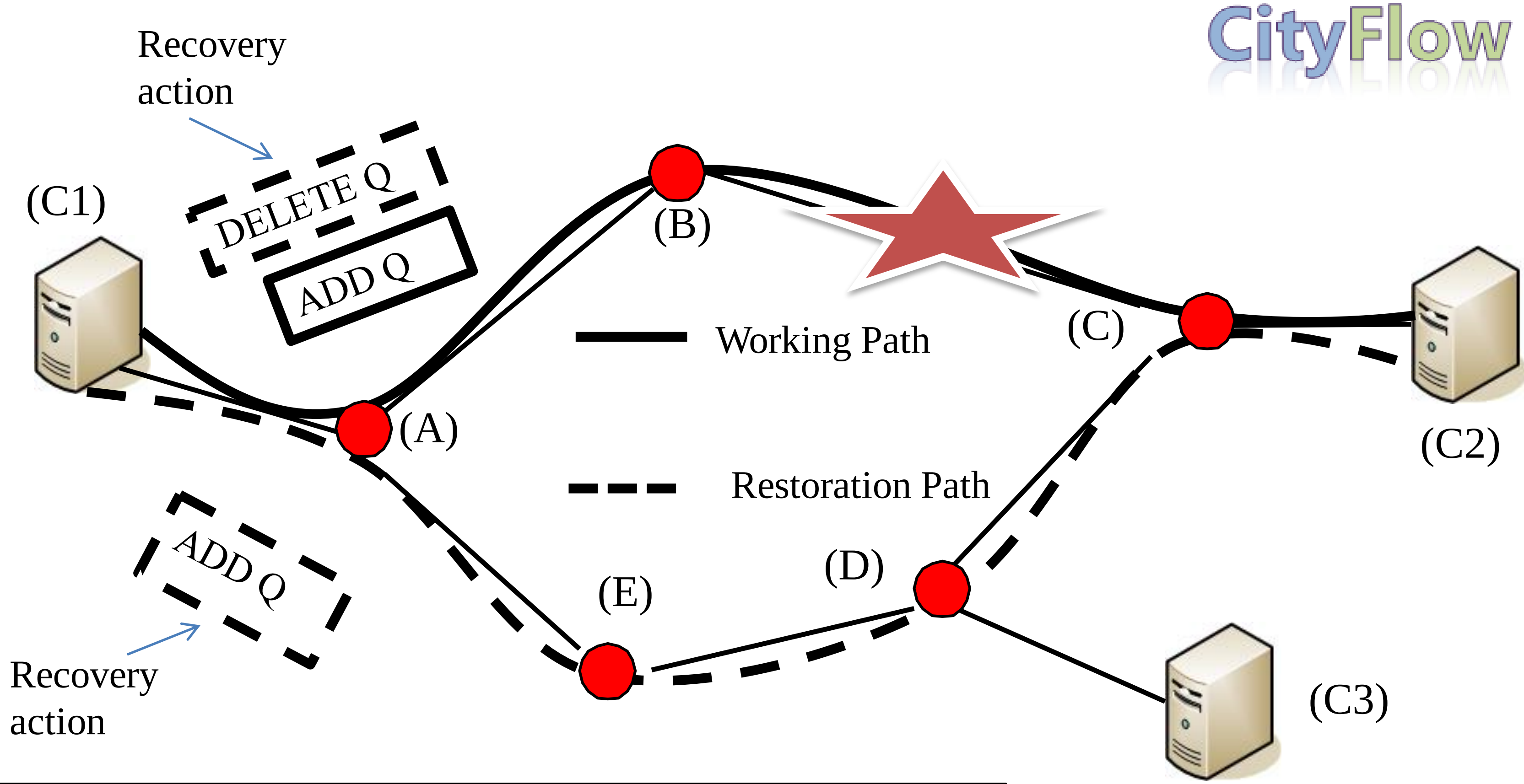
Our proposed Framework

CityFlow

Assumptions:

1. A single AS is controller by a single controller.
2. The controller is able to communicate with the bandwidth broker through the northbound API





By default, three queues are implemented for each port :

•The first queue is for **control traffic**

•The second queue is for **high priority traffic**

•The third queue is for **best effort traffic**

•Q is a **rate limiter queue for high priority traffic at the edge** router (Router A)

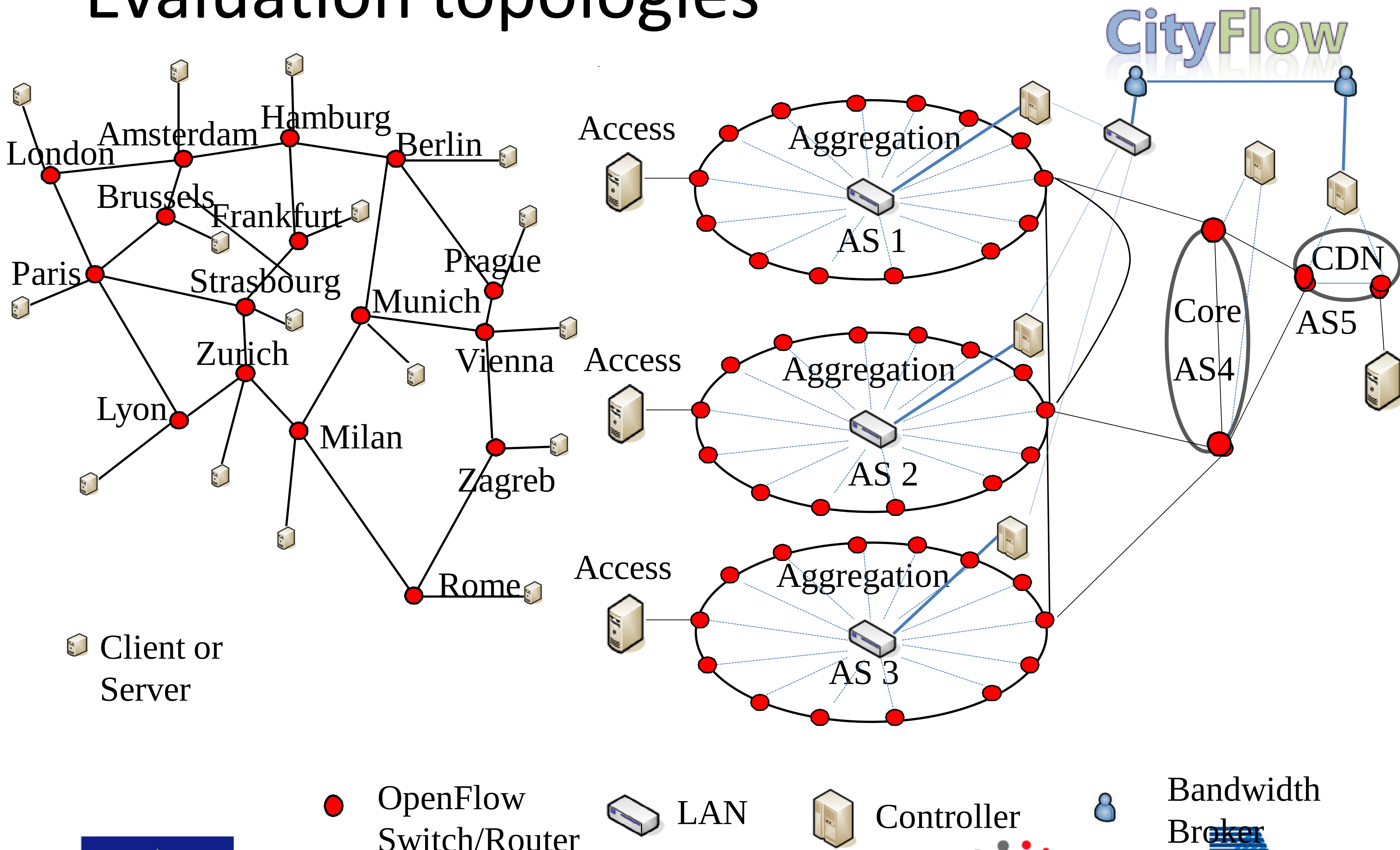
Objective of Experiments

CityFlow

- To evaluate three QoS failure recovery scenarios for SDN networks using our implemented Framework
  - When enough bandwidth in the working and the failure recovery path, neither high-priority nor best-effort traffic should get affected
  - When limited bandwidth, best-effort traffic get affected to fullfill the requirements of high-priority traffic
  - When very low bandwidth, high-priority traffic also gets affected, but there should be no interference from best-effort traffic.

epcc iMinds CONNECT.INNOVATE.CREATE ONESOURCE RedZinc SEVENTH FRAMEWORK PROGRAMME

Evaluation topologies



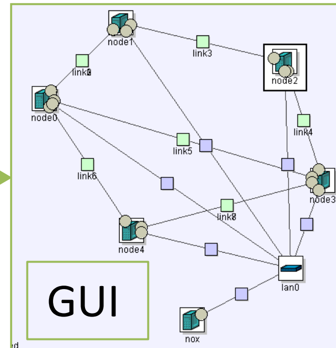
OFELIA island @ iMinds

CityFlow

Creating a CityFlow experiment

CityFlow

OpenFlow Experiment idea



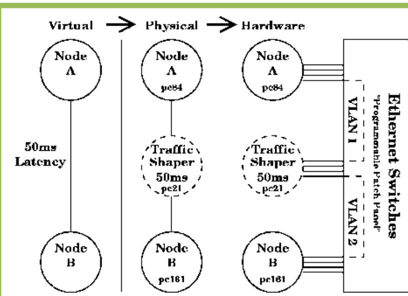
Use Cityflow OF images

```
#generated by Netbuild 1.03
set ns [new Simulator]
source tb_compat.tcl

#set ofclick1 [$ns node]
#tb-set-node-os $ofclick1 UBU1004-STD
#set ofclick2 [$ns node]
#tb-set-node-os $ofclick2 UBU1004-STD
set nox [$ns node]
tb-set-node-os $nox DEB60-STD

#set datachannel1 [$ns duplex-link $ofclick1 $ofclick2 1000Mb 0ms DropTail]
#set controlchannel1 [$ns duplex-link $ofclick1 $nox 1000Mb 0ms DropTail]
#set controlchannel2 [$ns duplex-link $nox $ofclick2 1000Mb 0ms DropTail]

#tb-set-ip-link $ofclick1 $datachannel1 172.16.0.1
#tb-set-ip-link $ofclick2 $datachannel1 172.16.0.2
#tb-set-ip-link $nox $controlchannel1 172.16.1.100
#tb-set-ip-link $ofclick1 $controlchannel1 172.16.1.1
#tb-set-ip-link $nox $controlchannel1 172.16.1.100
#tb-set-ip-link $ofclick2 $controlchannel2 172.16.2.1
#tb-set-ip-link $nox $controlchannel2 172.16.2.100

#$ns rtproto Static
$ns run
#netbuild-generated ns file ends.
```

ns script

Add script lines for automated datapath configuration

- Specify links to script
- Auto -> available NICs



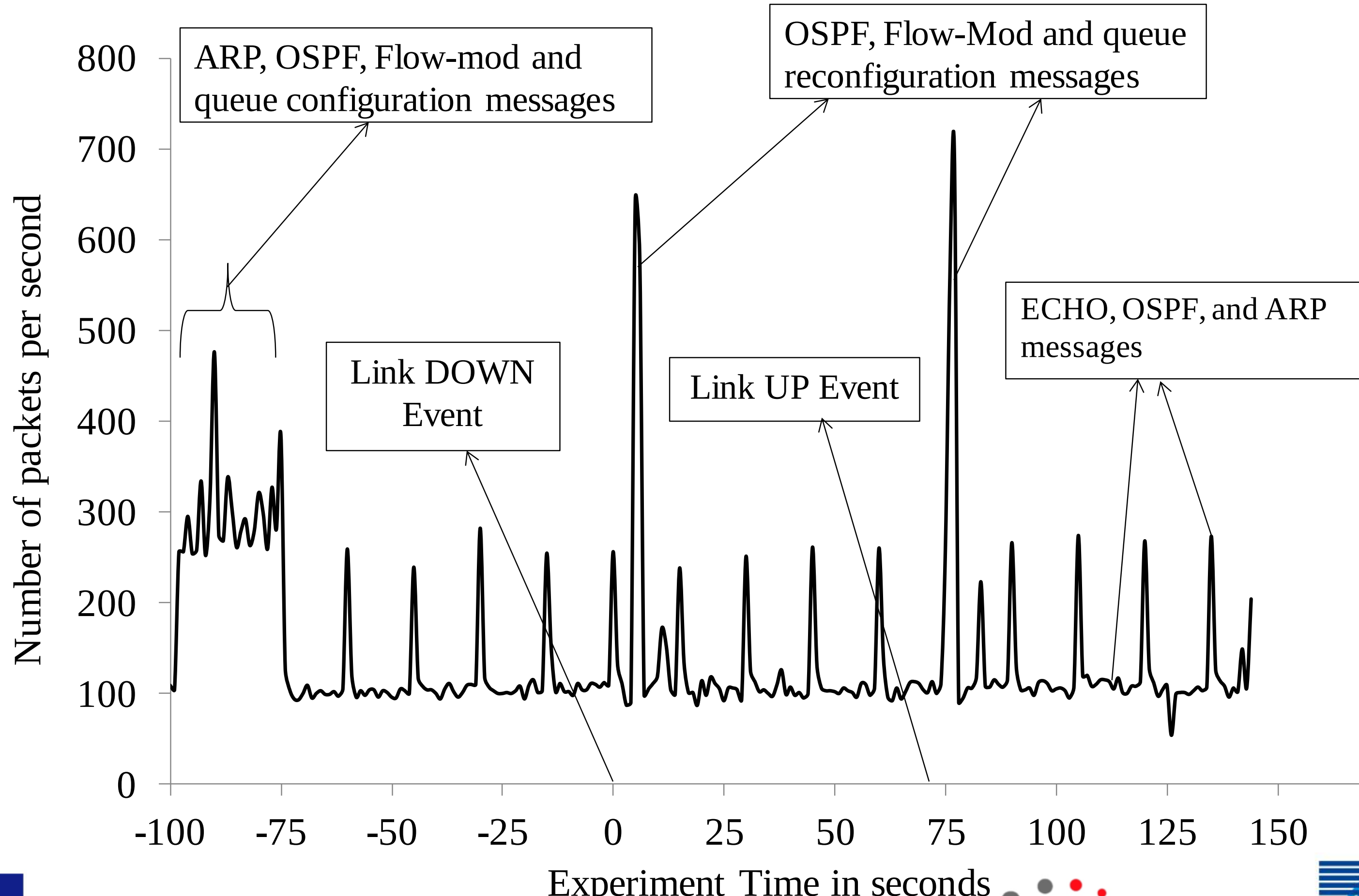
Hardware Mapping and swap in

Emulab runs the additional scripts from ns file

Emulation environment

CityFlow

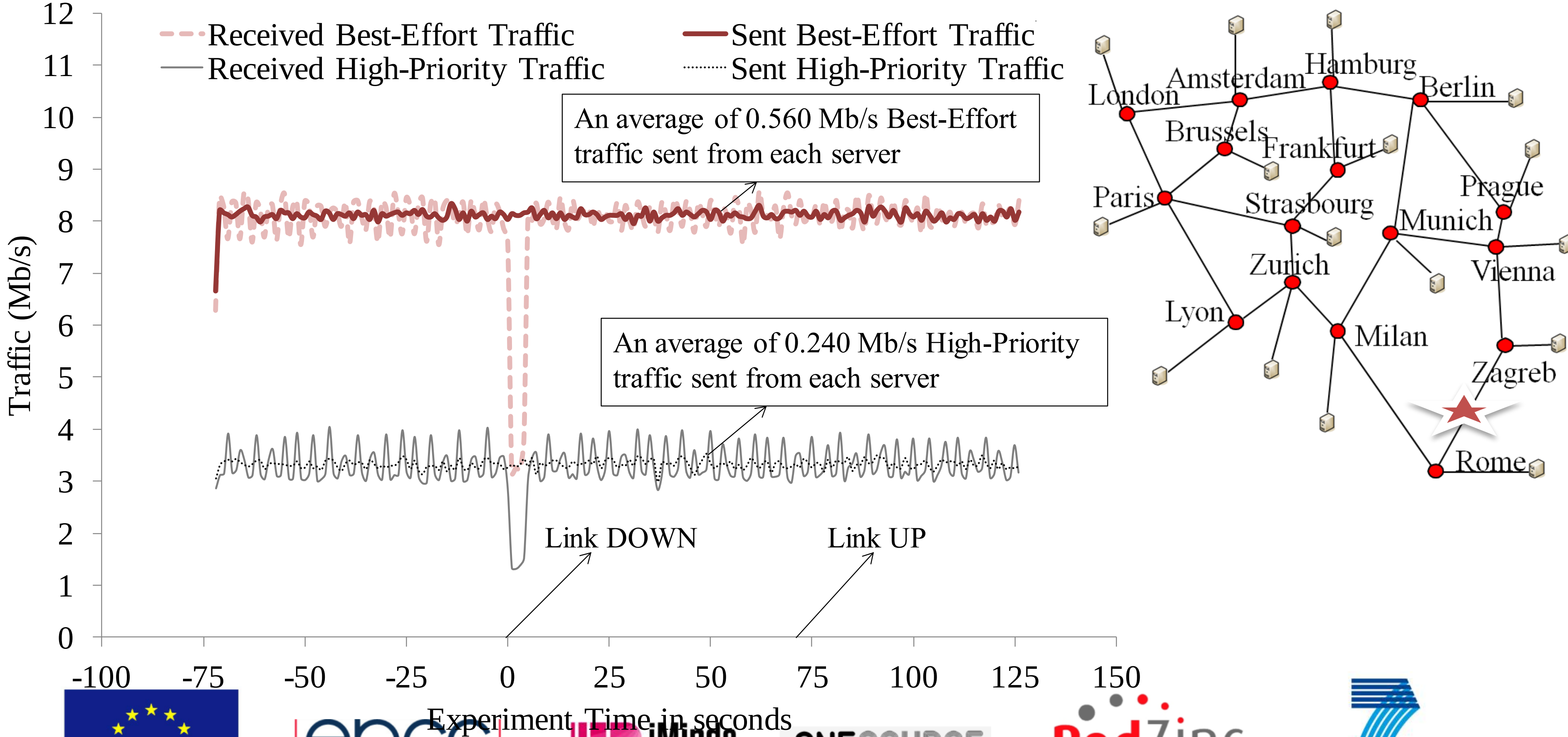
Traffic between the controller and switches



epcc iMinds CONNECT.INNOVATE.CREATE ONESOURCE RedZinc SEVENTH FRAMEWORK PROGRAMME

Traffic for the client connecting Zagreb

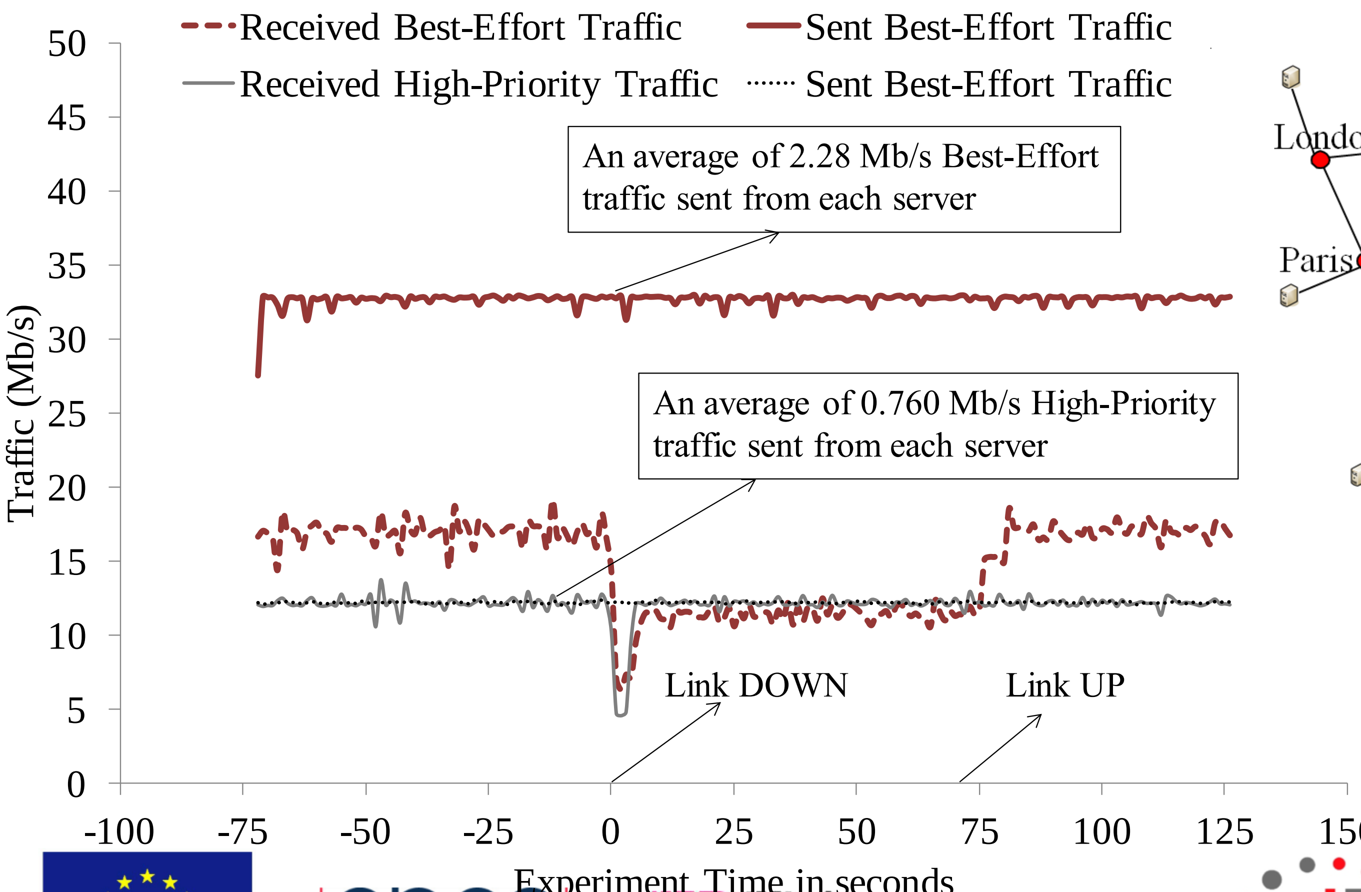
Best-Effort Traffic 0.560 Mb/s and High priority Traffic 0.240 Mb/s from each server

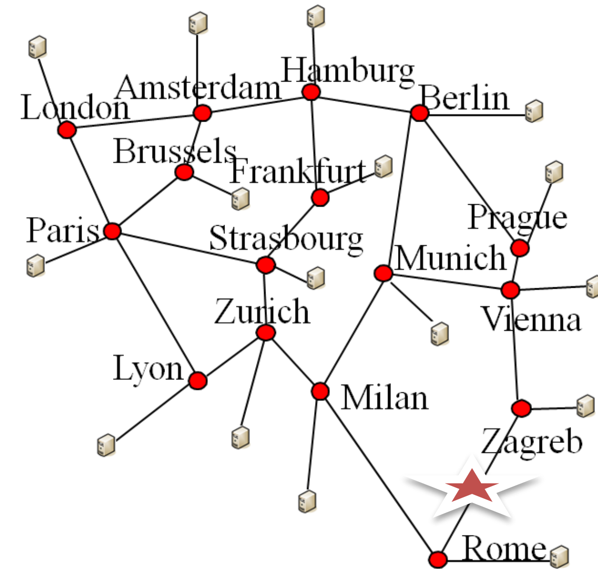


Traffic for the client connecting Zagreb

CityFlow

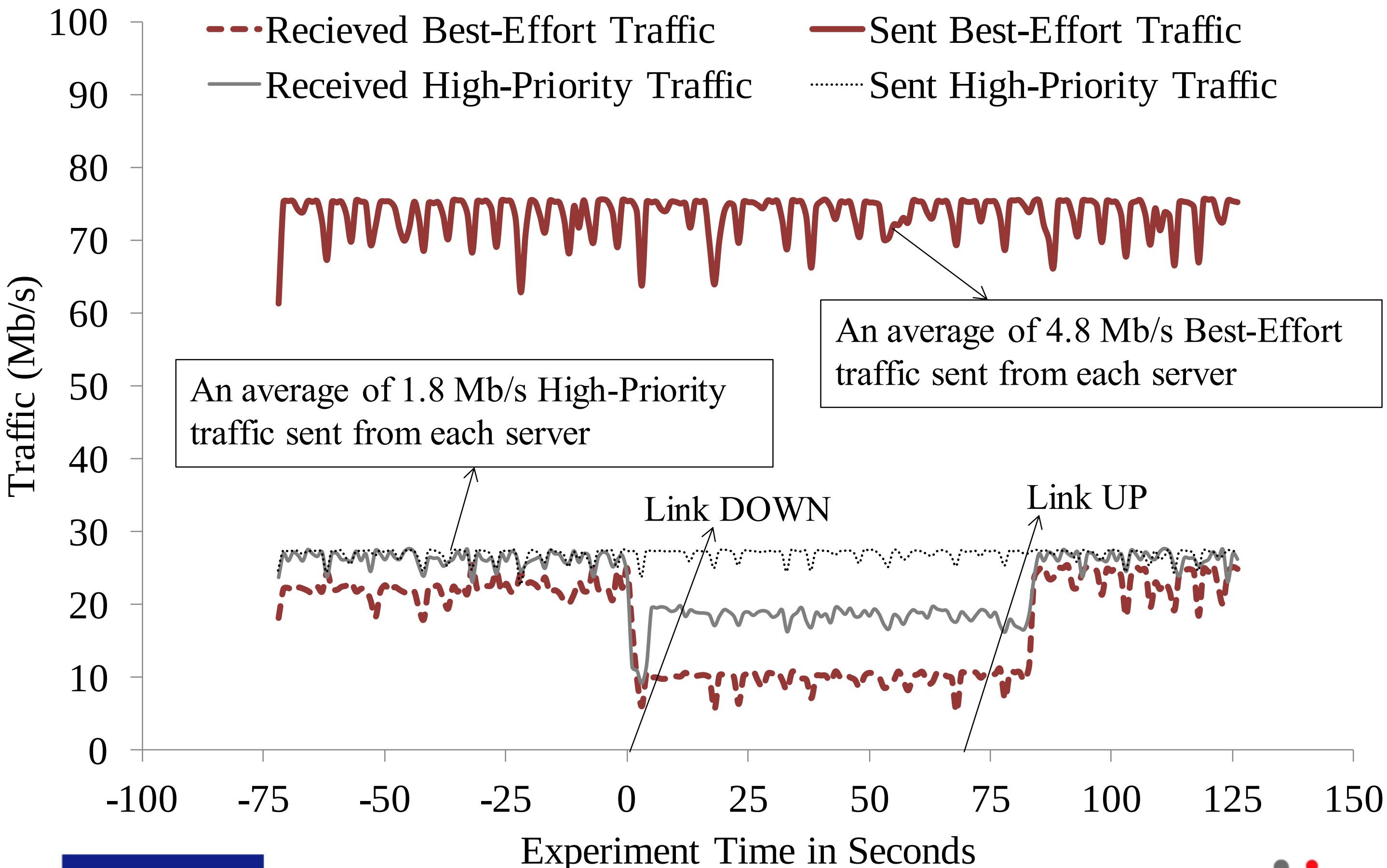
Best-Effort Traffic 2.28 Mb/s and High priority Traffic 0.760 Mb/s from each server



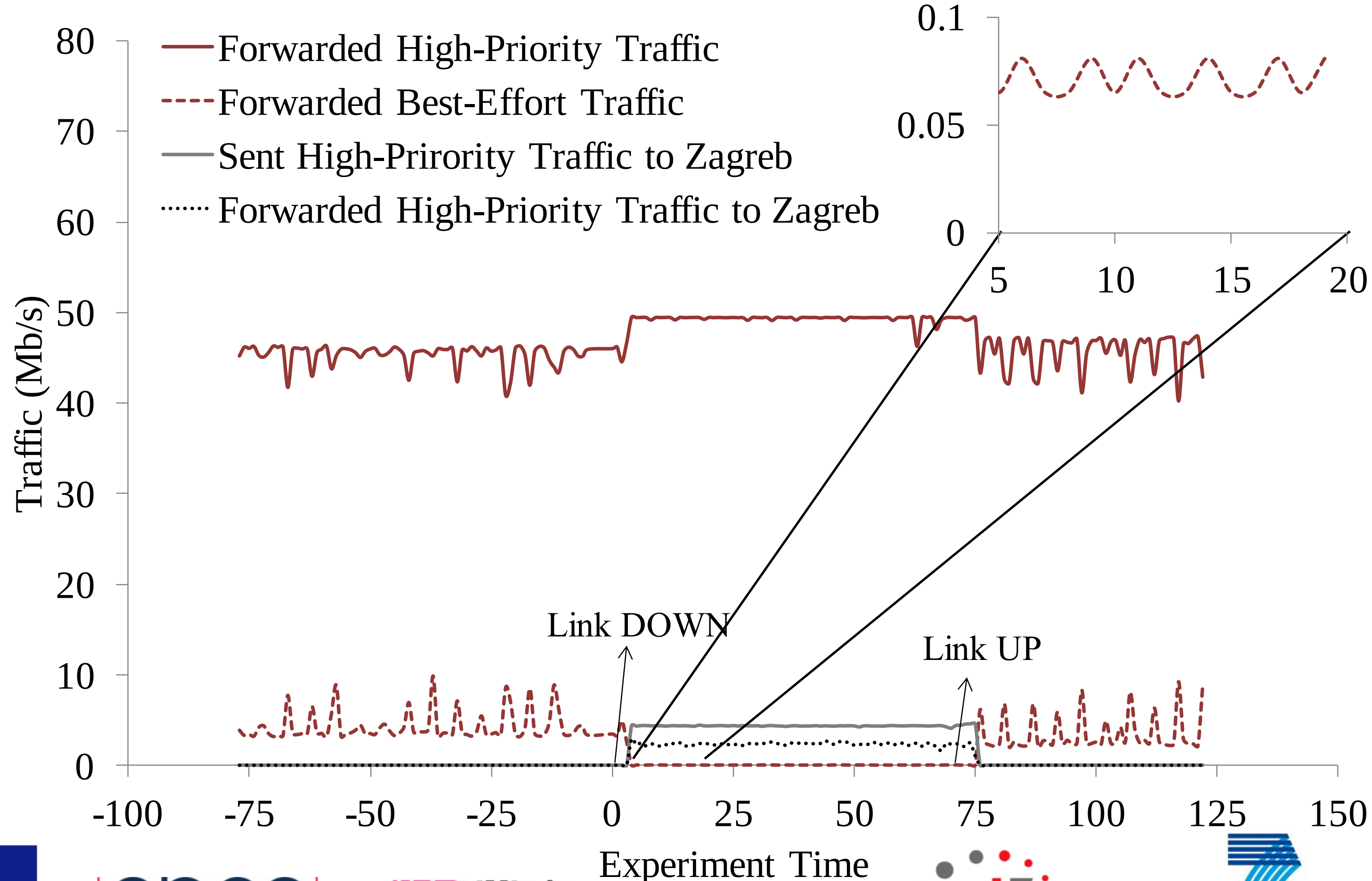


Traffic for the client connecting Zagreb

Best-Effort Traffic 4.8 Mb/s and High priority Traffic 1.8 Mb/s from each server

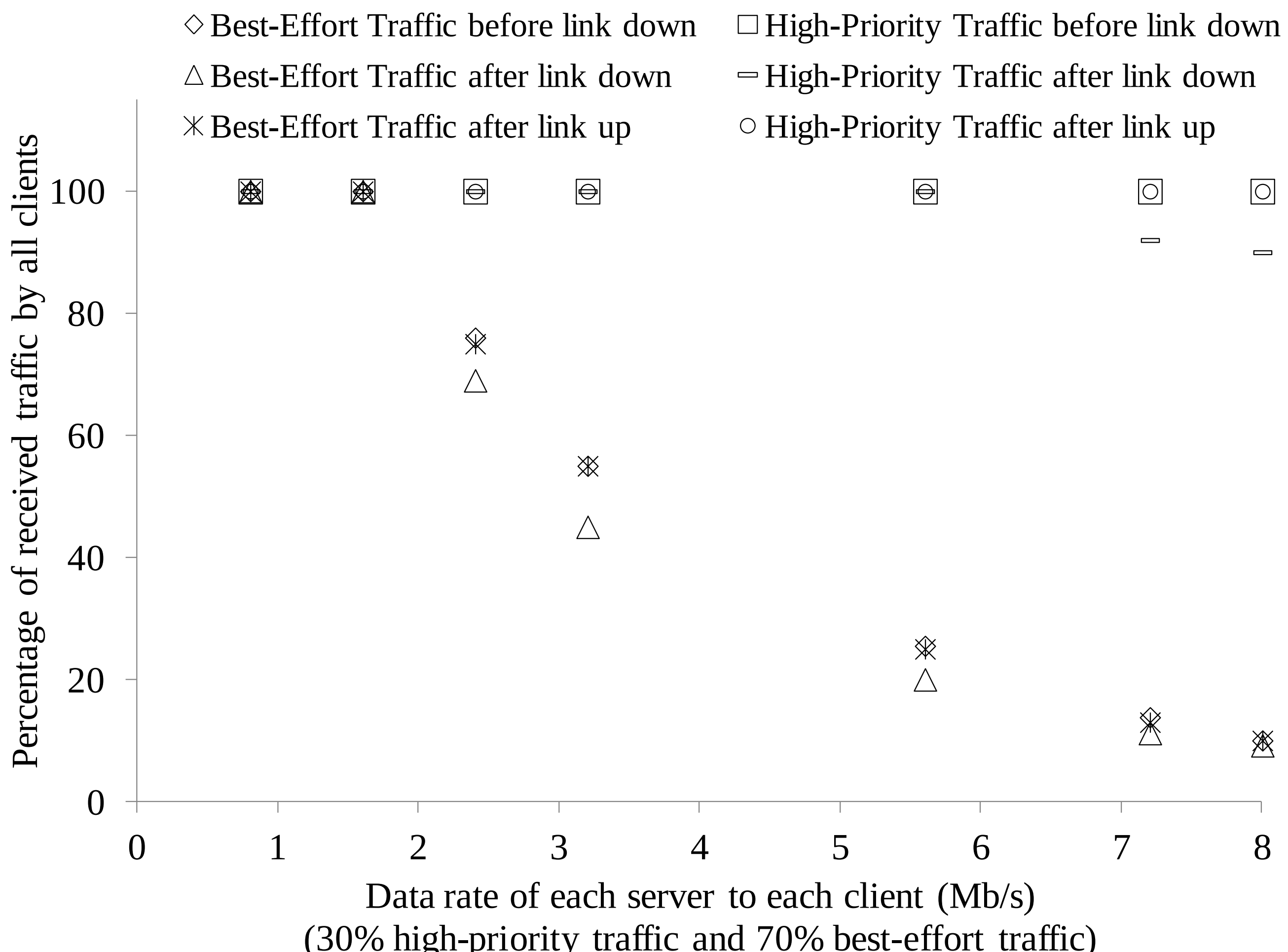


Traffic on link Milan-Munich (no interference link)



Total traffic received by all clients

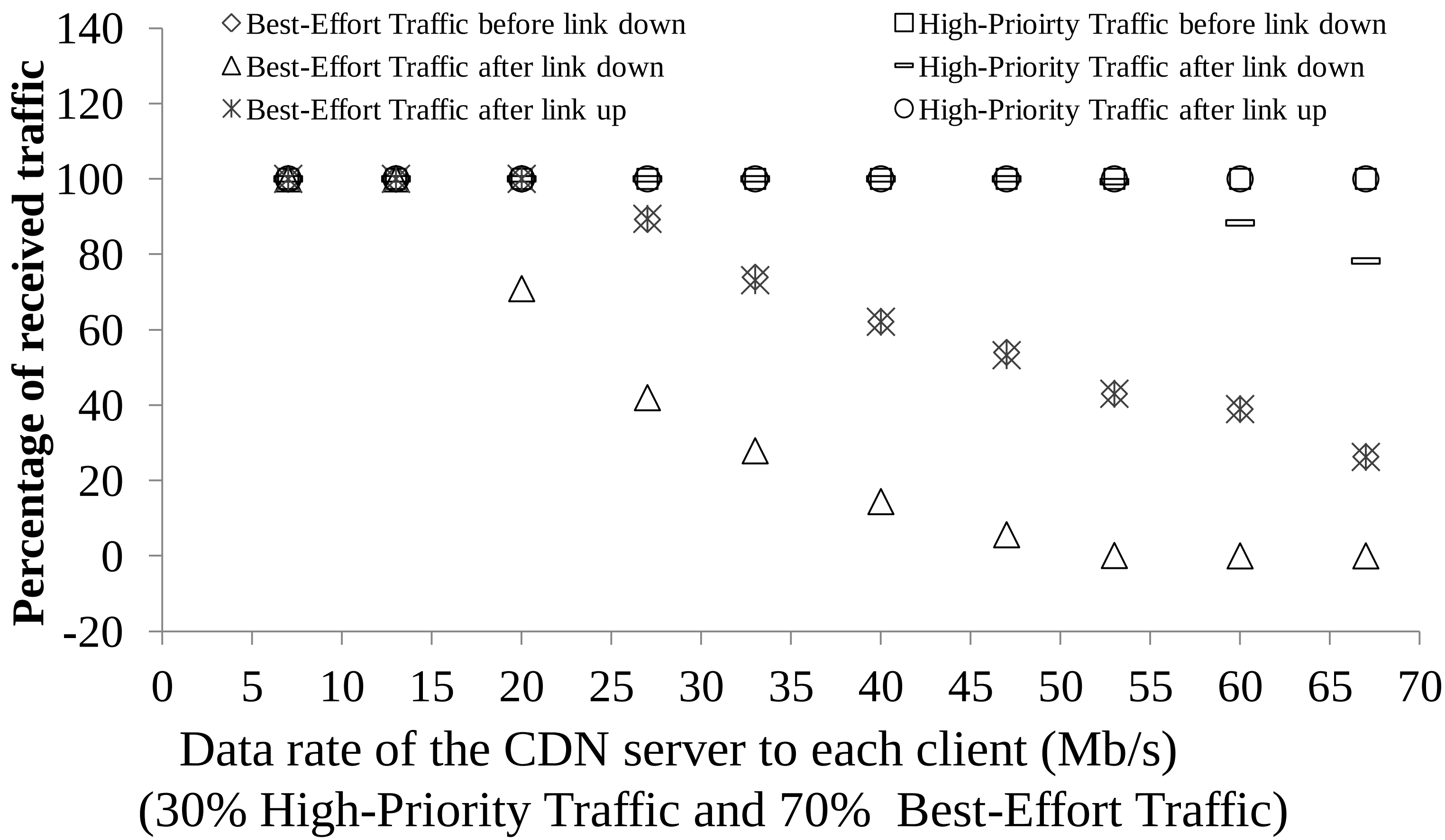
CityFlow



Multiple AS experiment

CityFlow

Total traffic received by all clients in access networks



Conclusions

- Designed a framework for Diffserv-like operation in OpenFlow
- Implemented QoS-aware failure recovery using VPS technology
- Experimentally validated the approach in single and multi-AS scenario’s

Questions?

CityFlow

- Sachin.Sharma@intec.ugent.be
- www.cityflow.eu (Project website for more detail)

epcc iMinds CONNECT.INNOVATE.CREATE ONESOURCE RedZinc SEVENTH FRAMEWORK PROGRAMME