Smart WWW Traffic Balancing

Erol Gelenbe
Ricardo Lent
Juan Arturo Nunez

School of Electrical Engineering & Computer Science

University of Central Florida

Introduction

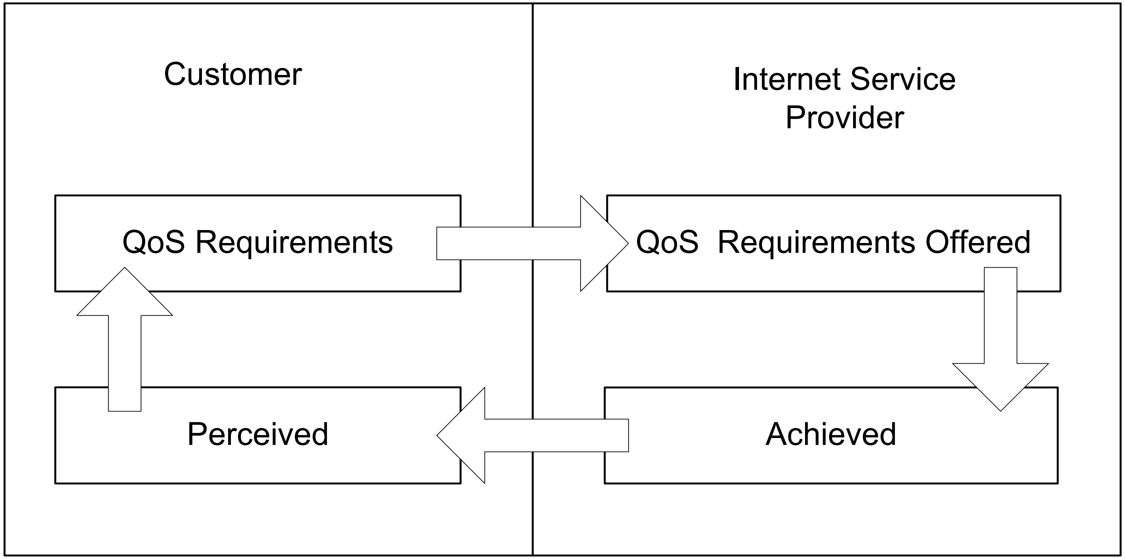
- The Internet is one of the biggest data networks in the world.
- Users rely in the infrastructure provided by the internet service providers (ISPs)
- There are new IP applications with more stringent QoS requirements.
- ISPs must fulfill those requirements.

Problem Context

– New applications have new QoS requirements.
  - Voice over IP (VoIP).
  - Video conferencing.
  - Network games ,etc.

– IETF has proposed several approches to serve QoS requirements, i.e. IntServ, DiffServ, IPv6

Problem Context (Continued)



Relationship between the ISP and the customer in terms of QoS

QoS in the Internet

- QoS in the Internet can be provided with different approaches
- The basic purpose is to improve application performance requirements and optimize network resources.

QoS in the Internet (continued)

- QoS may be provided via different network layers
  - Traffic Directing and Load Balancing (Application)
  - Traffic Engineering and Fast Reroute (Network)
  - Differentiated Services (Transport)

Traffic Directing and Load Balancing (Application Layer)

- Traffic re-directing distributes the traffic widely [akamai].
- Load balancing in a local scope tries to spread the traffic evenly

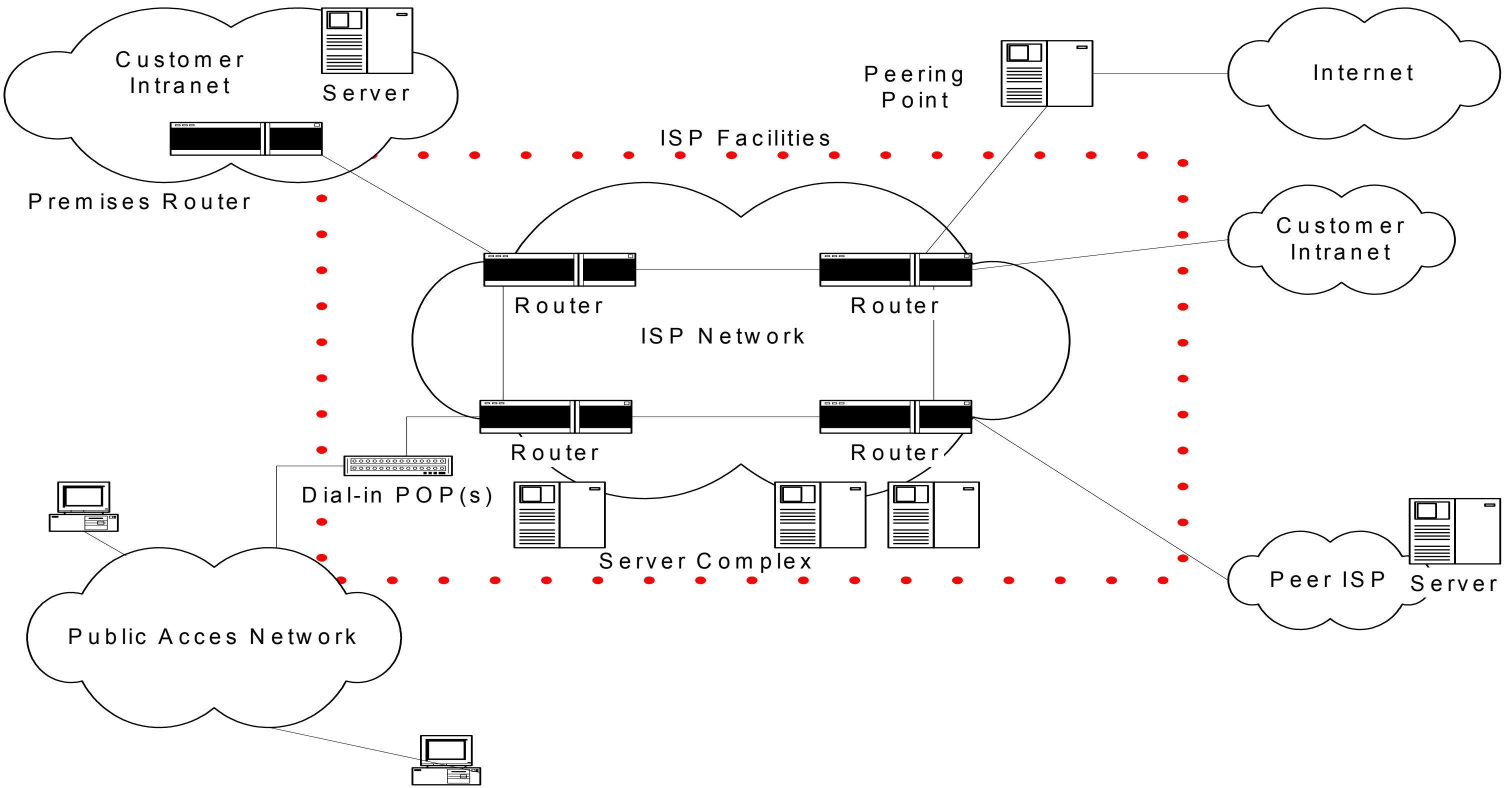
Traffic Engineering and Fast Reroute (Network Layer)

- Traffic Engineering optimizes resources efficiency and network performance, i.e. (MPLS) and enhanced IGPs
- Fast Reroute temporally repairs a failure link so it can continue to carry traffic before a more optimal path is computed.

Differentiated Services (Transport)

- Evolved from Integrated Services (IntServ) architecture.
- Intserv uses Resource Reservation Protocol (RSVP) to setup network resources.
- DiffsServ resolve scalability issues.
- Per flow service replaced with aggregate service.
- Complex processing is done at edge.

A New Proposal



We propose the use of the CPN as a framework at the level of the ISP to fulfill QoS requirements

Cognitive Packet Network (CPN)

- Source routing model.
- Best effort routing.
- Packets extract information from previous packets.
- Network functions performed per node.
- Traffic handled in a per-flow

CPN (continued)

- Consist of 3 different type of packets:
  - Smart Packets (SP)
  - Dumb Packets (DP)
  - Acknowledge Packets (AP)
- Node acts as a storage area for packets and mailboxes (MBs)
- Routing algorithm using Reinforcement Learning

Routing Algorithm

- In each node there is a RNN with as many "neurons" as outgoing links.
- The next hop is calculated based on the achievement or not of the QoS Goal (Delay), G = D
- The Reward function is the inverse of the Goal. (R = 1/ G)
- Decision threshold : $T_l = \beta T_{l-1} + (1 - \beta) R_l$

Routing Algorithm (continued)

- Update of network weights as follows

If T_{l-1} ☯ R_l

$w^+(i,j)$ ♌ $w^+(i,j) + R_l$

$w^-(i,j)$ ♌ $w^-(i,j) + \frac{R_l}{n-2}$, k ✎ j

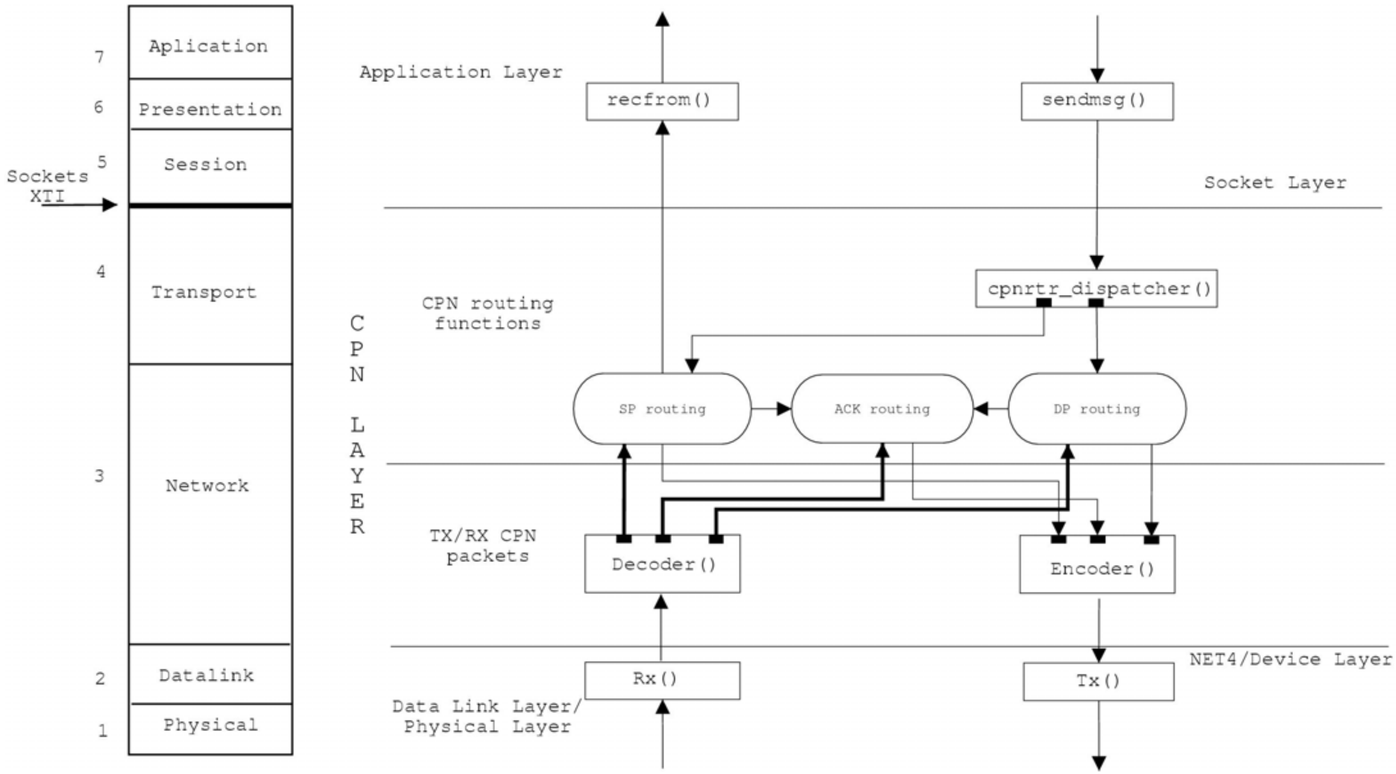
Else

$w^+(i,j)$ ♌ $w^+(i,j) + \frac{R_l}{n-2}$, k ✎ j

$w^-(i,j)$ ♌ $w^-(i,j) + R_l$

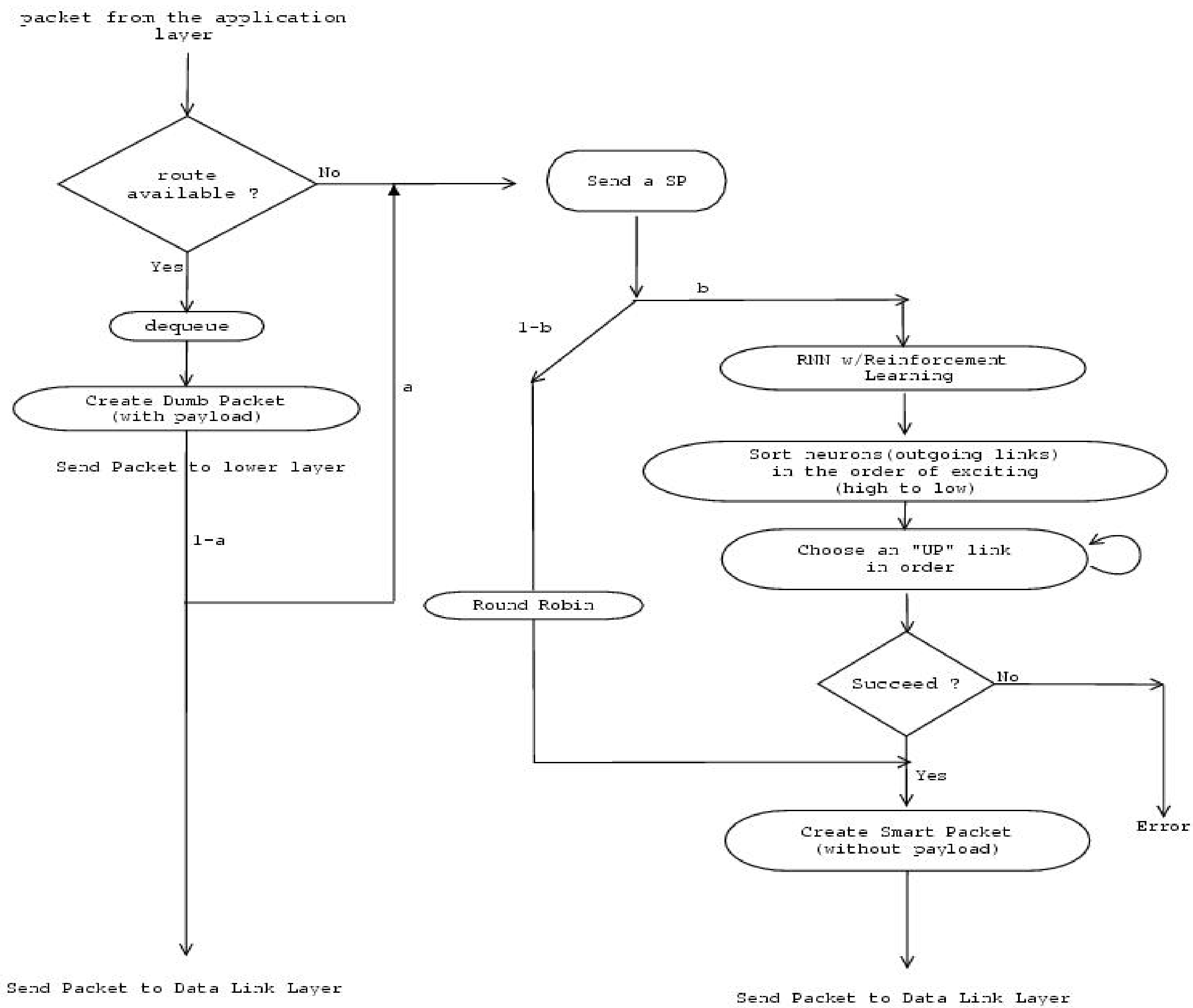
- Finally weights are re-normalized

Protocol Overview



CPN protocol suite

Generation of a SP and DP at the Source Node



Framework

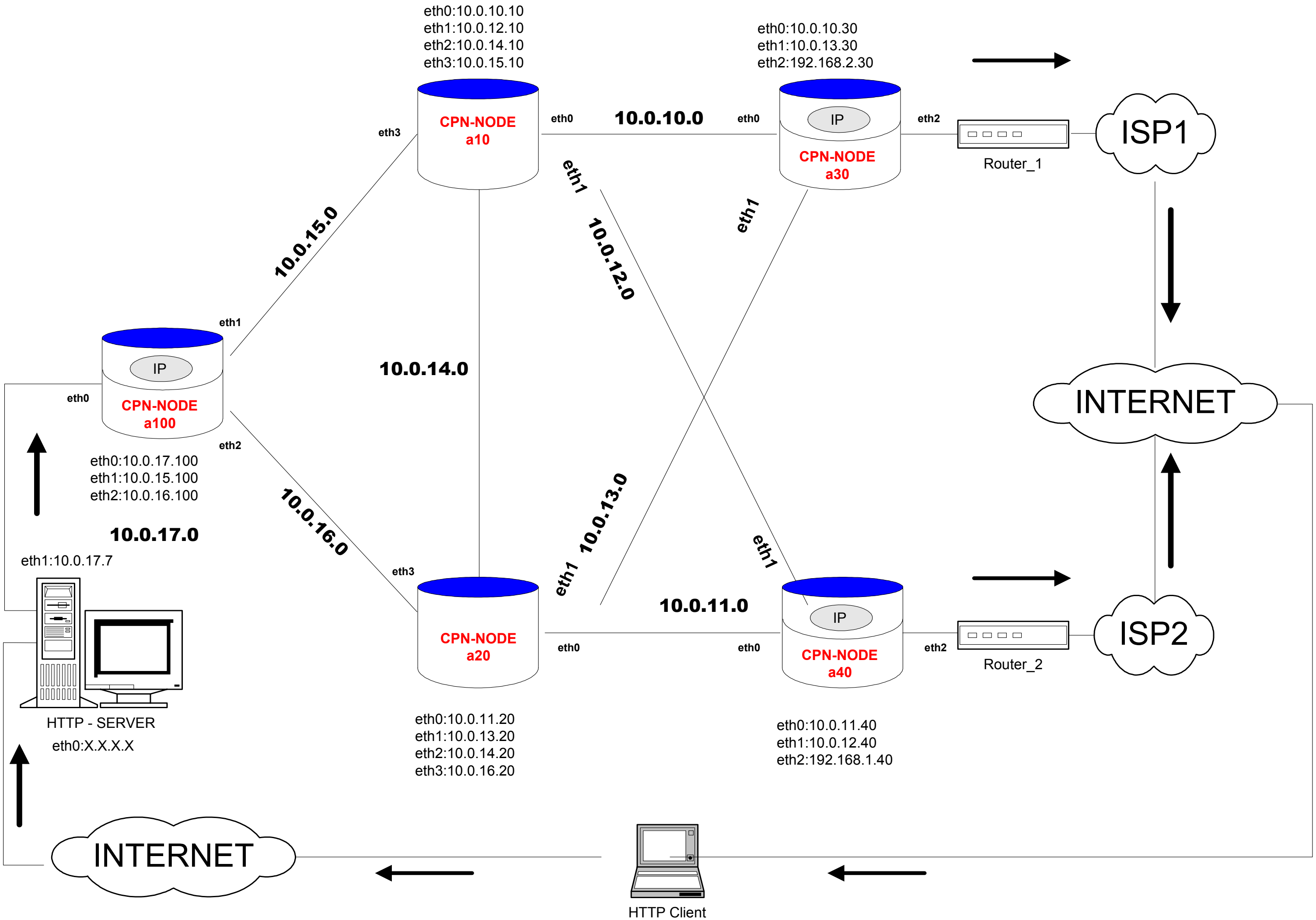
Design Goals:

- Minimize delay as the basic QoS requirement.
- Interoperativility.
- Transparency.
- Per-Flow basis treatment of traffic.

Overall Architecture

- The test-bed used emulates an ISP.
- Division of the Components:
  - Hardware: 1 Web server(PC), 5 Core PCs, 2 Cisco Routers,1 Http client (Sun Ws)
  - Software: Capture, Edge, Xnetload

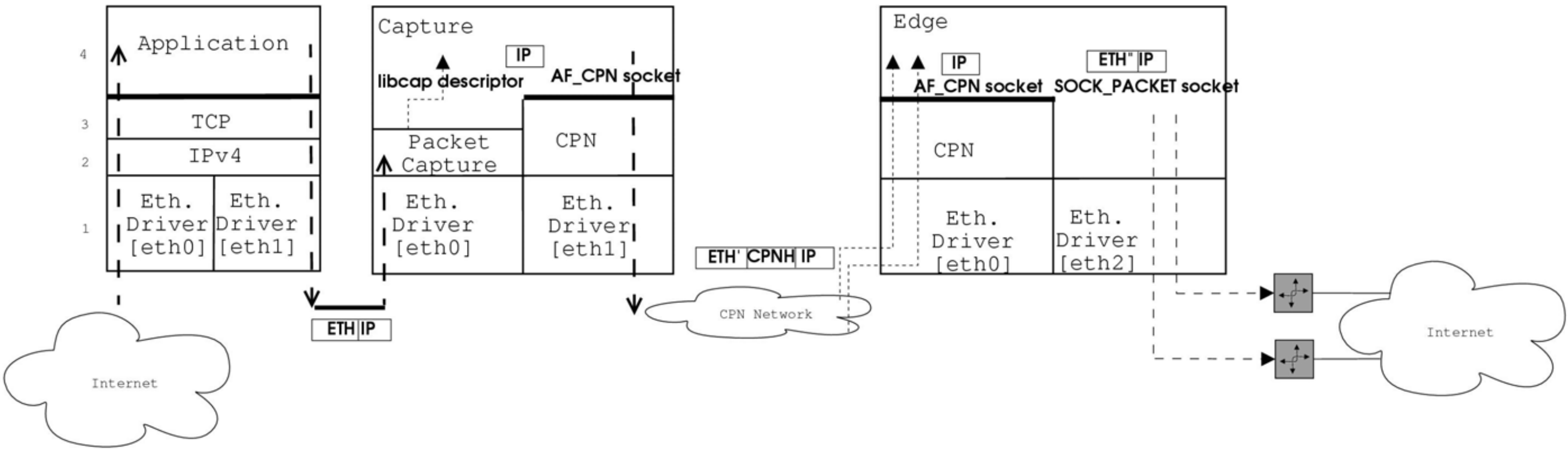
Topology



eth0:10.0.10.10
eth1:10.0.12.10
eth2:10.0.14.10
eth3:10.0.15.10
CPN-NODE a10
eth3
eth0
10.0.10.0
eth0
eth0:10.0.10.30
eth1:10.0.13.30
eth2:192.168.2.30
IP
CPN-NODE a30
eth2
Router_1
ISP1
eth1
eth1
10.0.12.0
10.0.15.0
eth1
IP
CPN-NODE a100
eth0
eth2
10.0.14.0
INTERNET
eth0:10.0.17.100
eth1:10.0.15.100
eth2:10.0.16.100
10.0.17.0
eth1:10.0.17.7
10.0.16.0
10.0.13.0
eth1
eth3
eth1
10.0.11.0
CPN-NODE a20
eth0
eth0
IP
CPN-NODE a40
eth2
Router_2
ISP2
HTTP - SERVER
eth0:X.X.X.X
eth0:10.0.11.20
eth1:10.0.13.20
eth2:10.0.14.20
eth3:10.0.16.20
eth0:10.0.11.40
eth1:10.0.12.40
eth2:192.168.1.40
INTERNET
HTTP Client

Operation

- Capture, encapsulates IP packets over CPN packets.
- Edge, deencapsulates CPN packets and send them to the neighbor router.

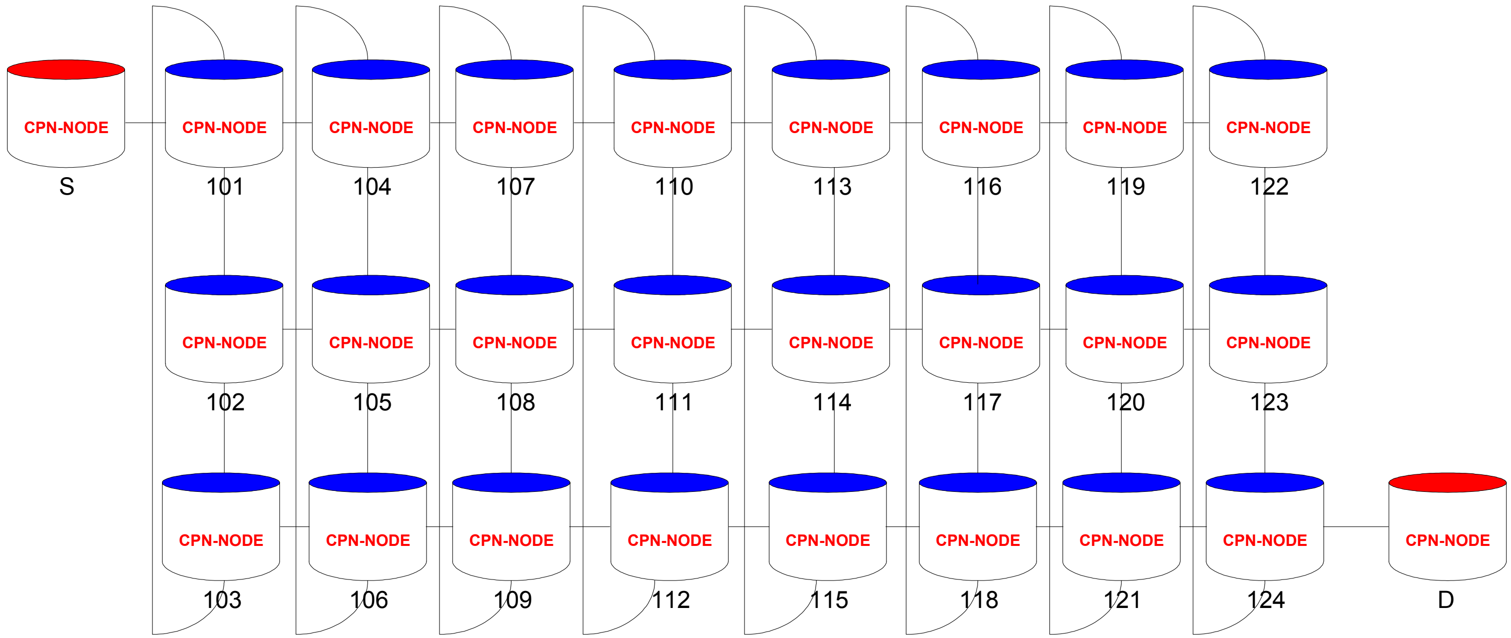


Set of Experiments

- Cold start connection set-up time measurements.
- Dynamic QoS control over the Internet.

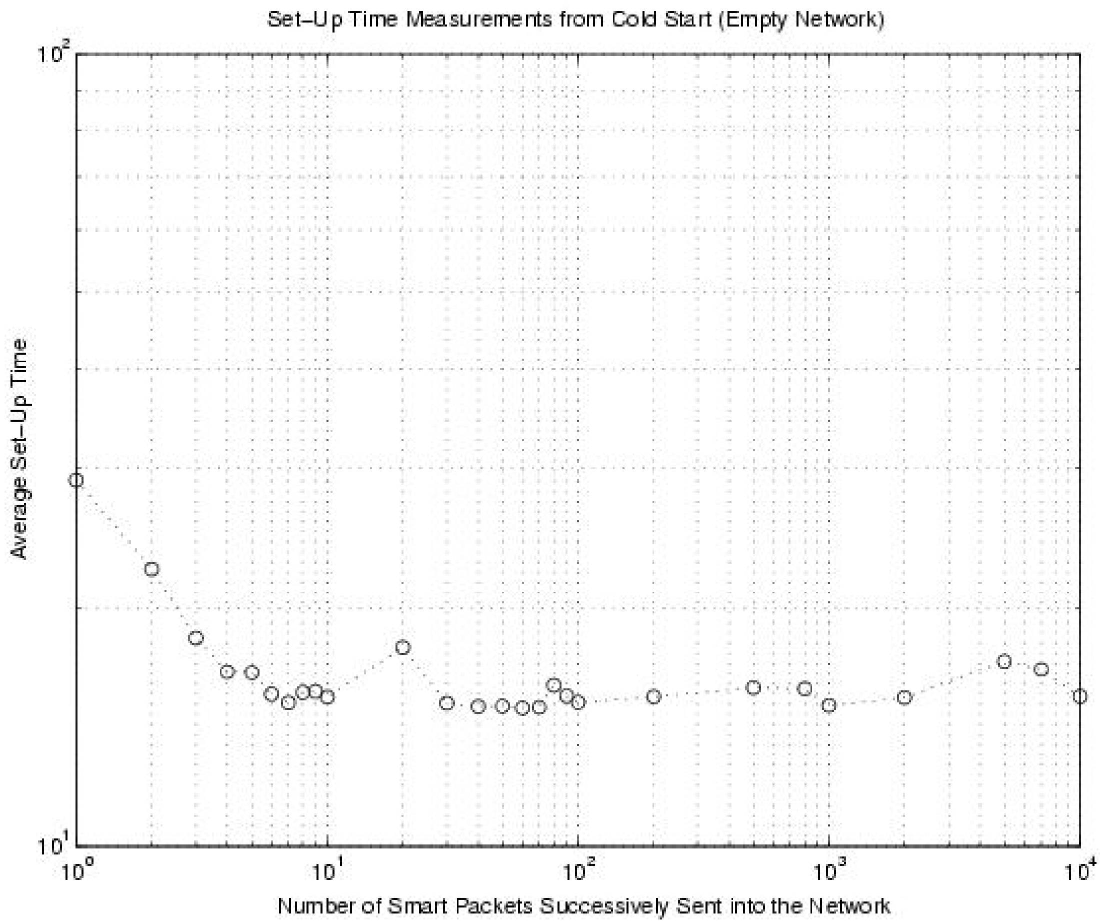
Experiments: Cold Start Connection Set-up Time Measurements

- Source routing, S-D path.
- Empty mailboxes.
- We want show the behavior of the network to set-up new S-D paths.



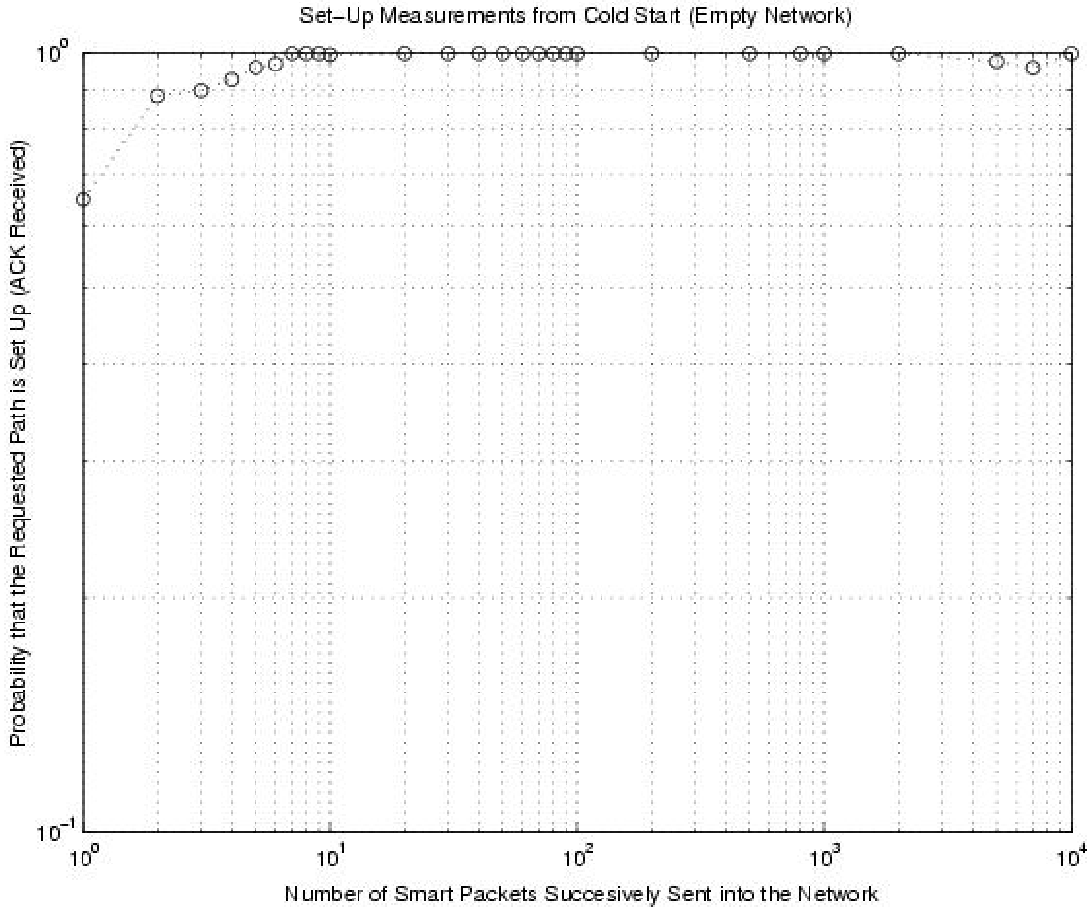
Average Set-up Time

- Avg. Delay vs number of SPs.
- Improves when sending more than 1 SP.
- More than 10-20 does not improve.
- Value ~ 10-20 ms.



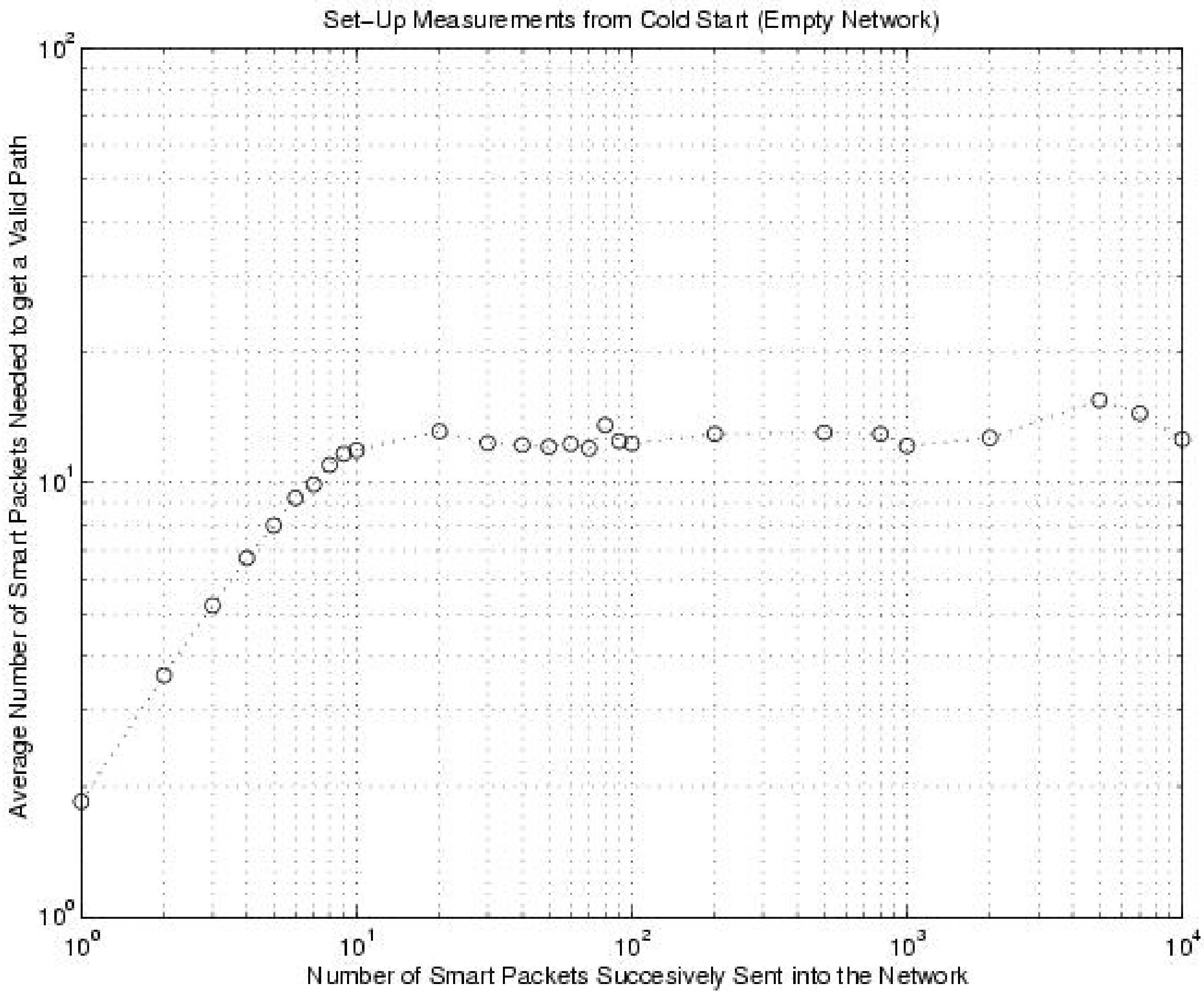
Probability of Successful Connection

- After sending more than 10 SP probability is 1.

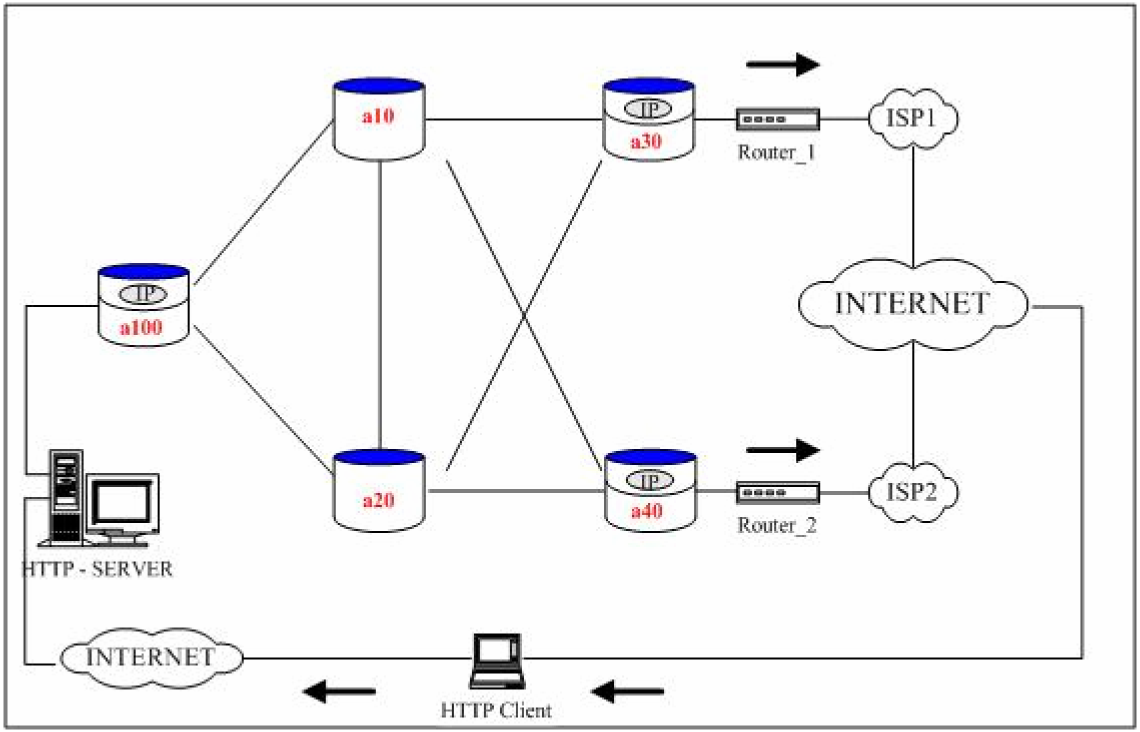


Number of Smart Packets Successively Sent

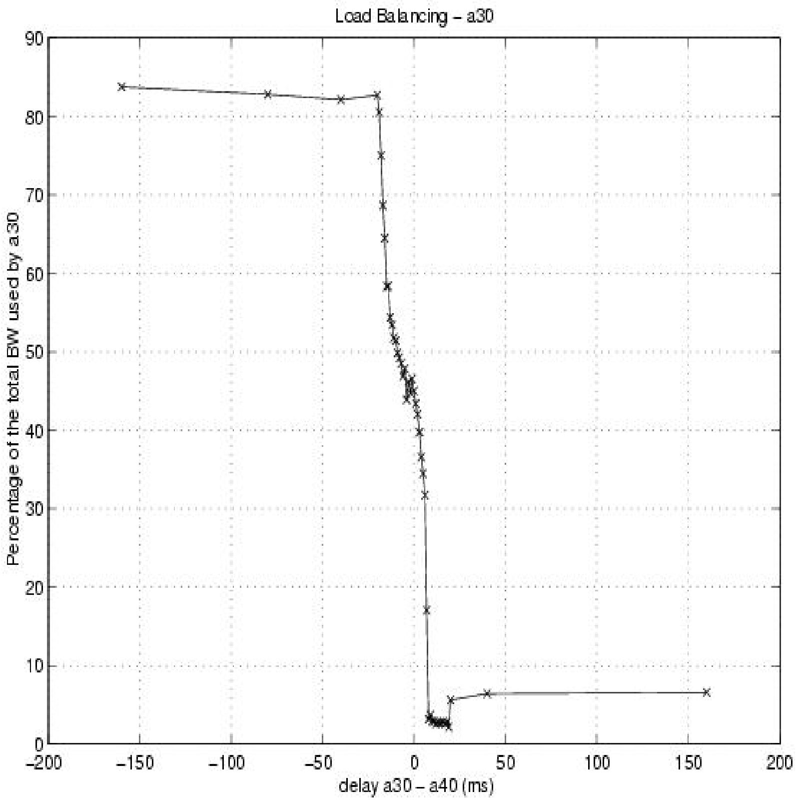
- For small number of SPs, until 10 relationship is linear.
- After 12 SPs connection is alreadystablis hed



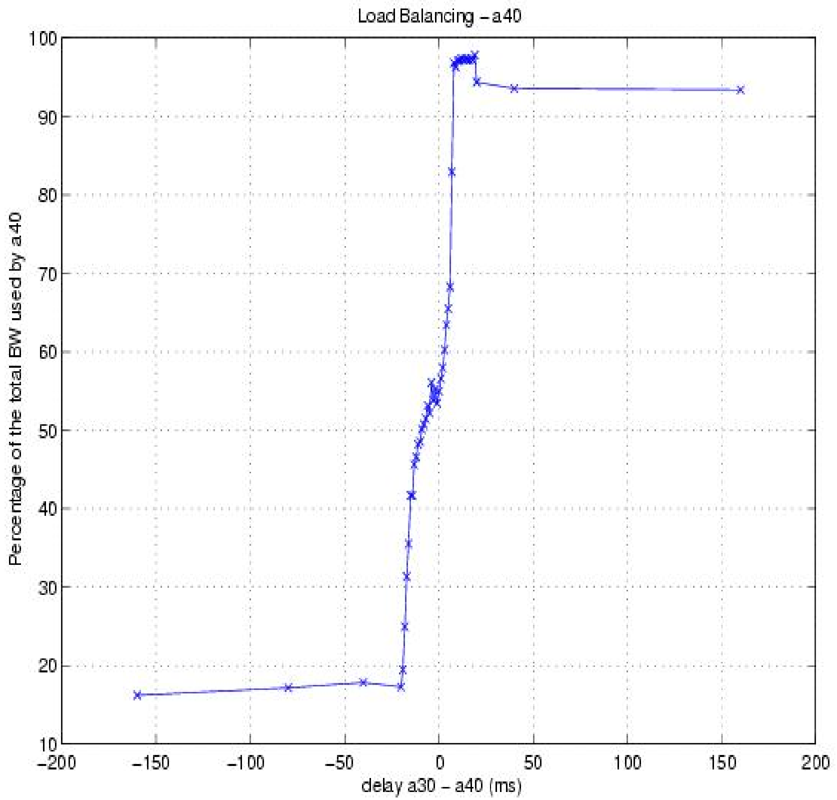
Experiments: Dynamic QoS Control (Framework)



Web Traffic Flow to Destination via Alternative Ports

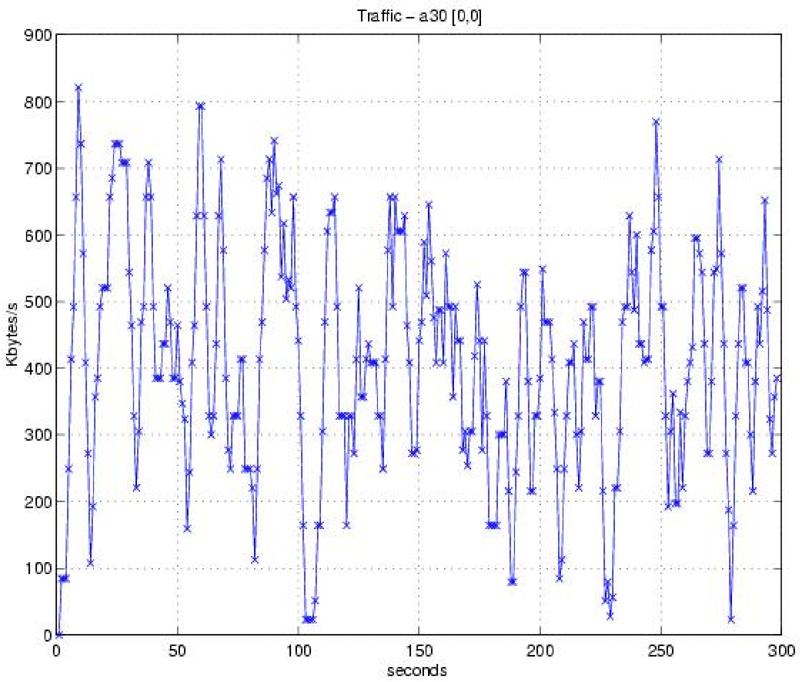


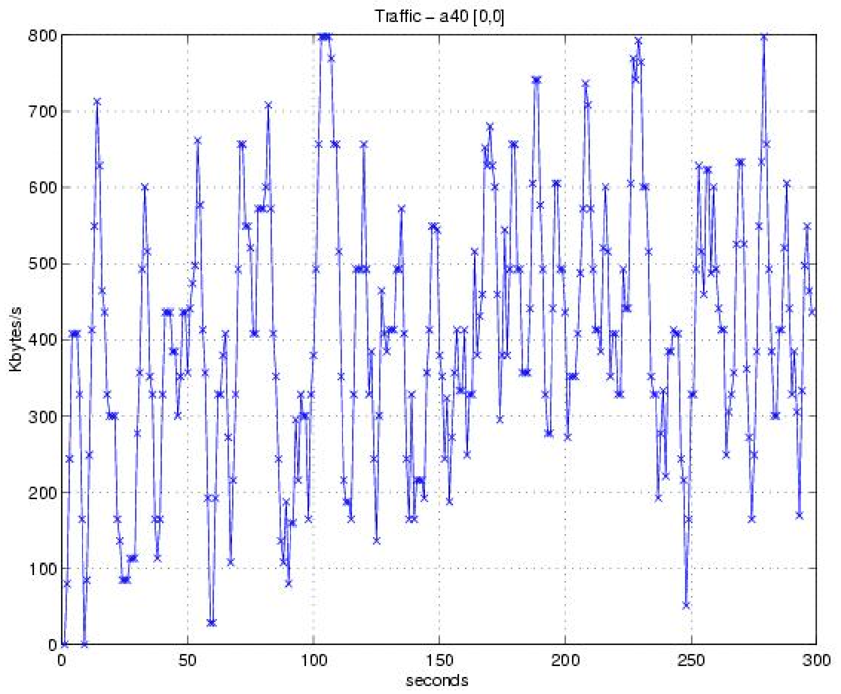
Port a30



Port a40

Delay a30 – a40 = 0

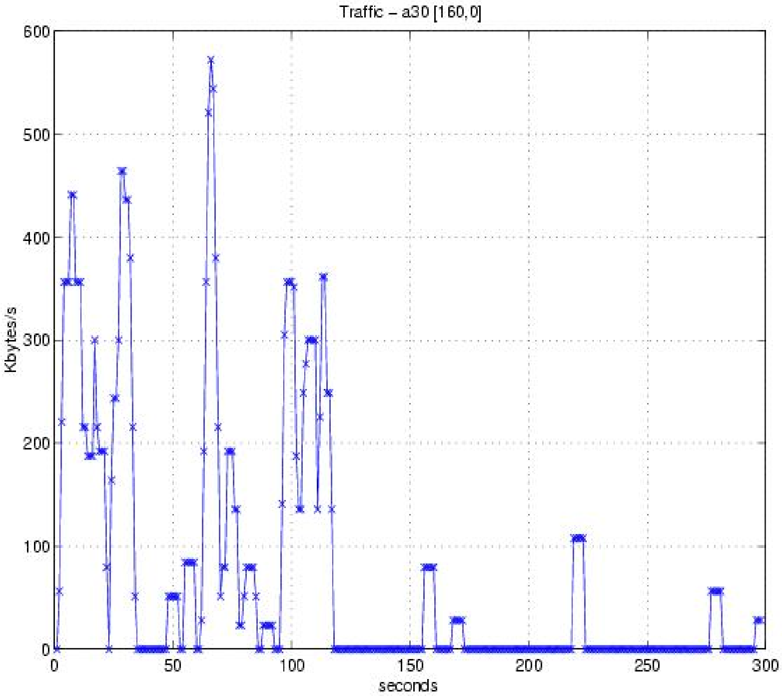


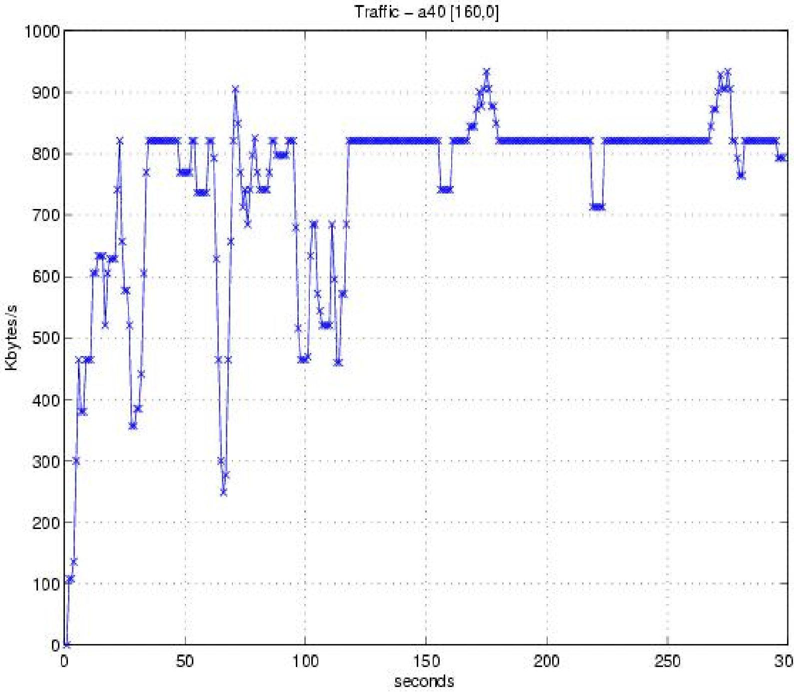


Port a30

Port a40

Delay a30 – a40 = 160

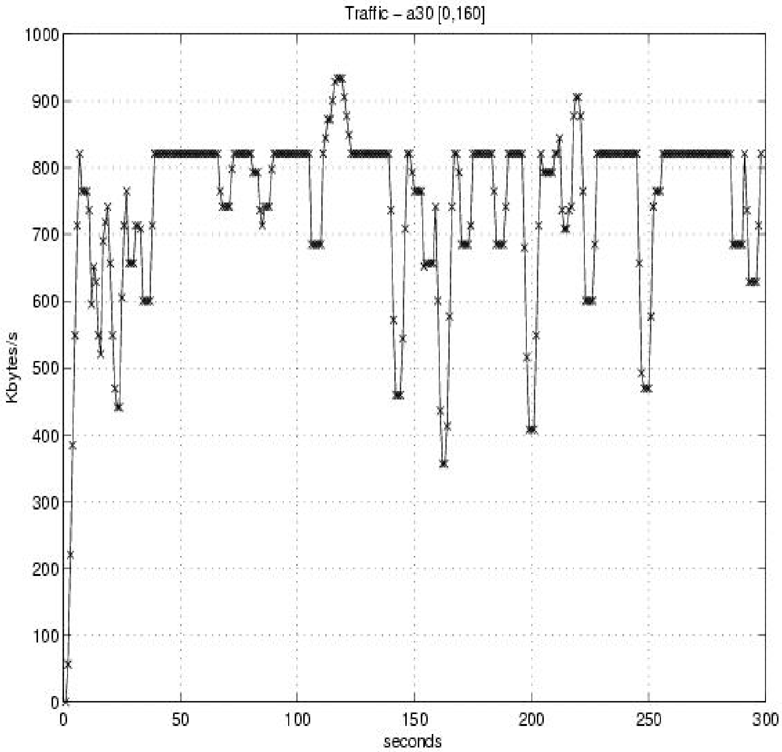




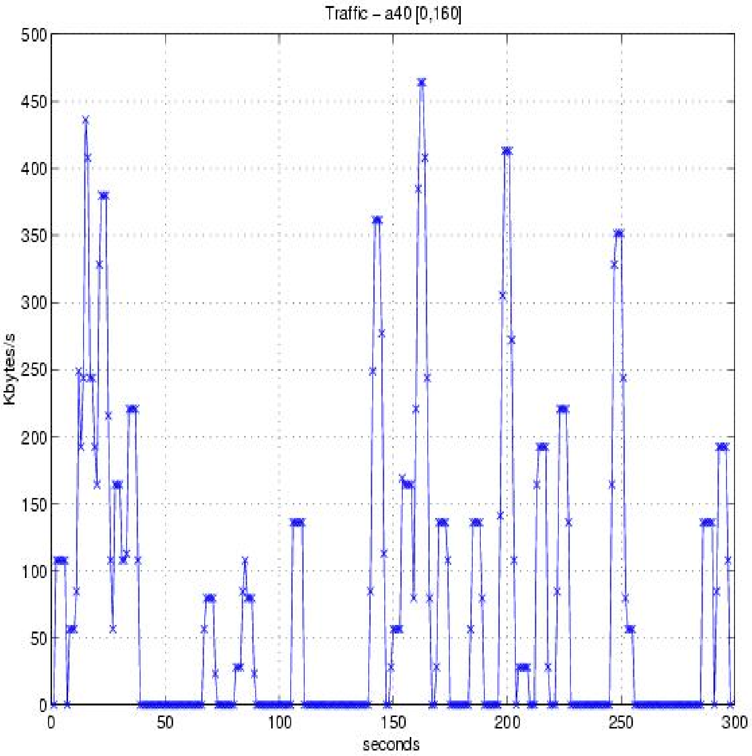
Port a30

Port a40

Delay a30 – a40 = -160



Port a30



Port a40

Conclusions

- Design goals of the framework were accomplished.
- Through experimental results in a practical scenario the framework shows its viability to fulfill QoS requirements for both the end-user and the ISP
- Framework delivers for the end-user at the application layer of QoS (delay)
- Framework allows the ISP to perform TE in a flow-basis (Similar to IntServ and RSVP)
- For the topology presented the total number of SP needed to reach a valid path is small (~10)

Future Work

- Conduct tests with more than 1 server (A farm of servers)
- Conduct tests with more than 1 source (Client).
- Use of CPN technology to improve LSP in a MPLS environment.
- IntServ relies in RSVP, CPN technology could be use to enhance RSVP.