# Quality of Service Aspects in an IPv6 Domain

Authors:
Ch. Bouras, A. Gkamas, D.Primpas, K. Stamos

Research Academic Computer Technology Institute, Greece
University of Patras, Computer Engineering and Informatics Department, Greece

http://ru6.cti.gr

# QoS techniques

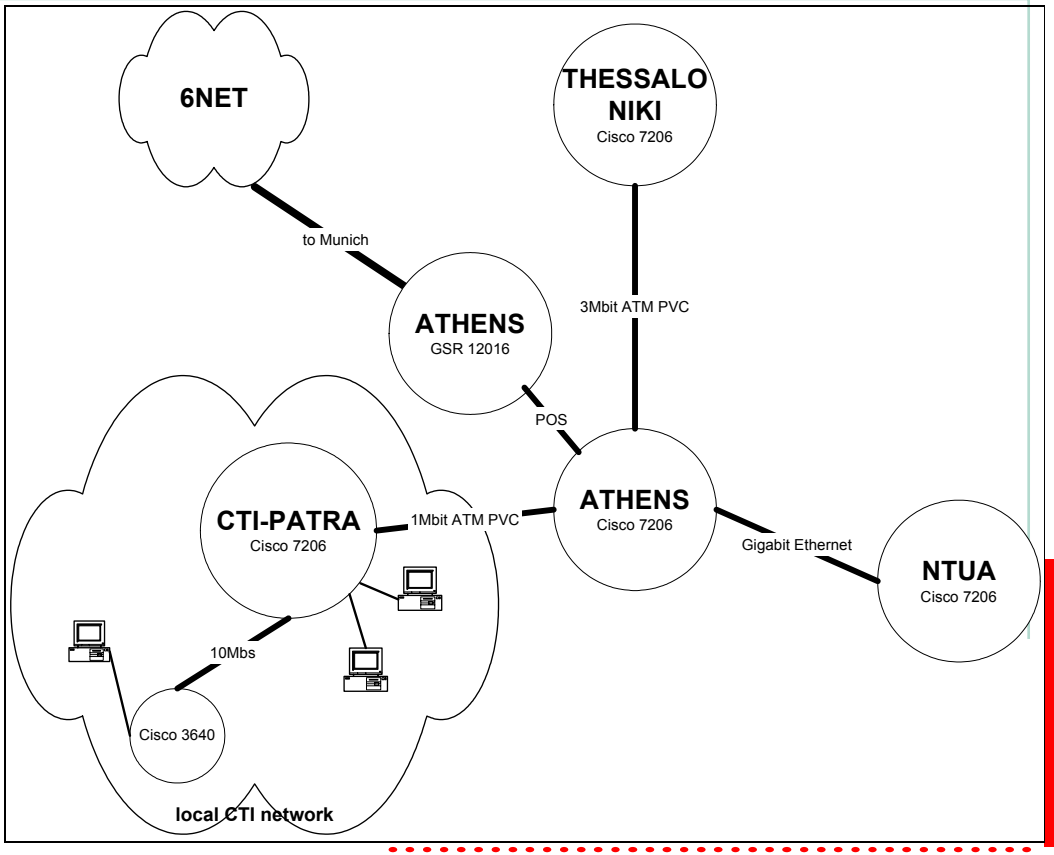
- IntServ
  - absolute guarantees via resource reservations across the paths (RSVP)
  - quite complicated operation
  - inserts significant network overhead
- DiffServ
  - classifies all the network traffic into classes
  - 2 different types (per hop behaviours):
    - expedited forwarding (EF): aims at providing QoS for the class by minimizing jitter and is generally focused on providing stricter guarantees
    - assured forwarding (AF): inserts at most 4 classes with at most 3 levels of dropping packets

# QoS and IPv6

- IPv6: New Internet Protocol
  - $2^{128}$ addresses
  - other improvements
- QoS mechanisms that are currently supported for IPv6 in most implementations are fewer (or different) compared to IPv4
- RFC 3697 (Flow Label specification)
- The whole network's behaviour is different
- QoS services should be designed and evaluated again

# 6NET network

- Greek part of the network
- CTI network:
  - Cisco router 7206
  - Cisco router 3640
  - 2 network switches, various pc
  - CISCO IOS 12.2(13)T



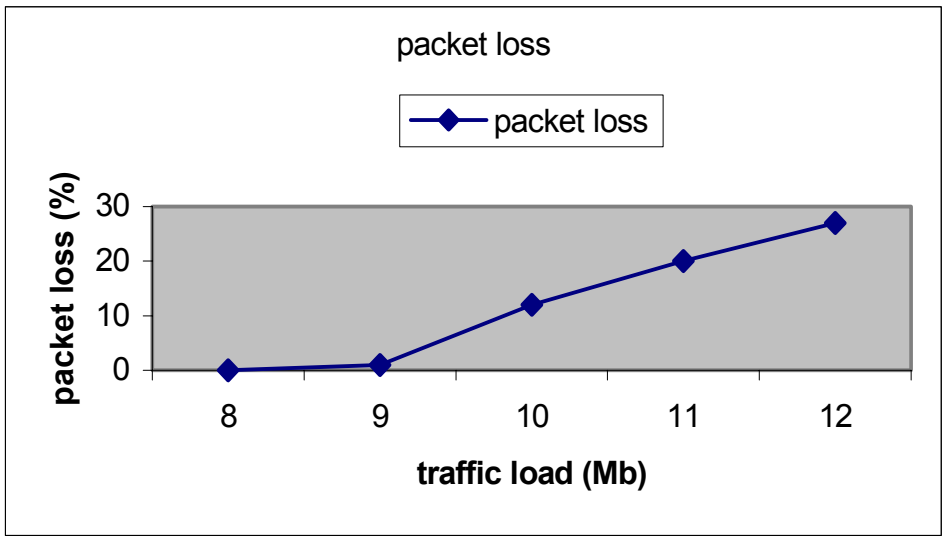
# Goals

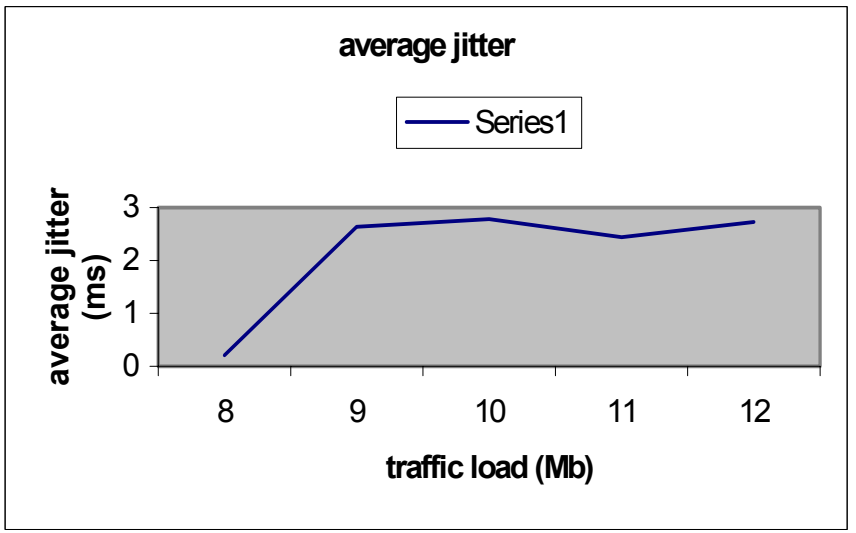
- Test an EF based service for real time applications
  - Investigate classification mechanism
  - Investigate prioritization mechanism
  - Investigate policing mechanism
  - Test all the mechanism under different traffic loads
- Test the WRED mechanism on the background traffic
  - Investigate mechanism's operation
  - Investigate its impact on QoS service

# Experimental Procedure

- Traffic generated with Iperf traffic generator
  - IPv6 UDP traffic
    - Periodic UDP traffic with specific bandwidth
  - IPv6 TCP traffic
    - Try to sent with the fastest rate possible
- Real time traffic
  - IPv6 traffic created by OpenPhone (videoconference traffic using OpenH323 library)
- Investigation of network's performance
  - Congested when traffic load is up to 8Mb (10Mb link)

# Experiments (network's behavior)

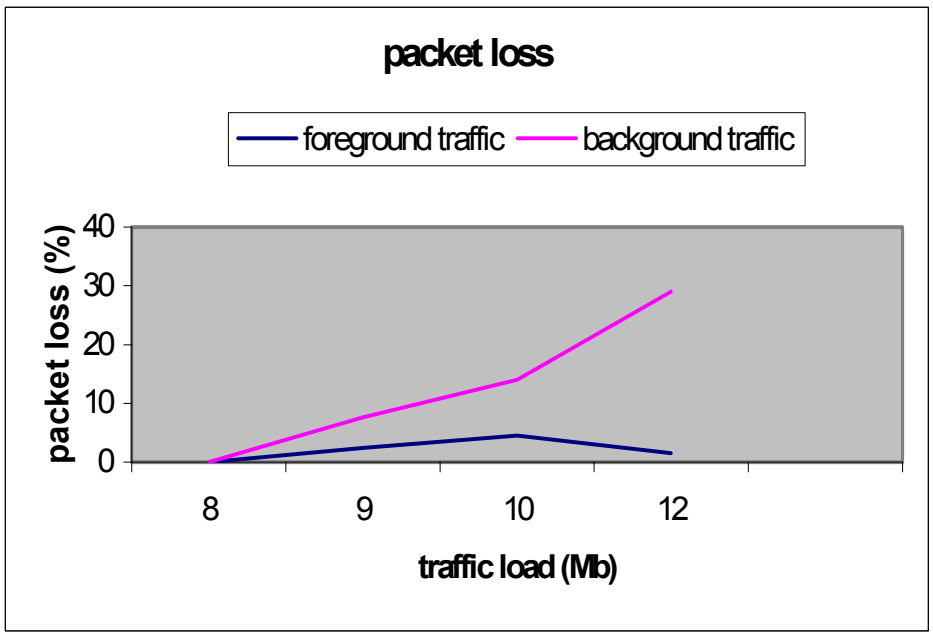


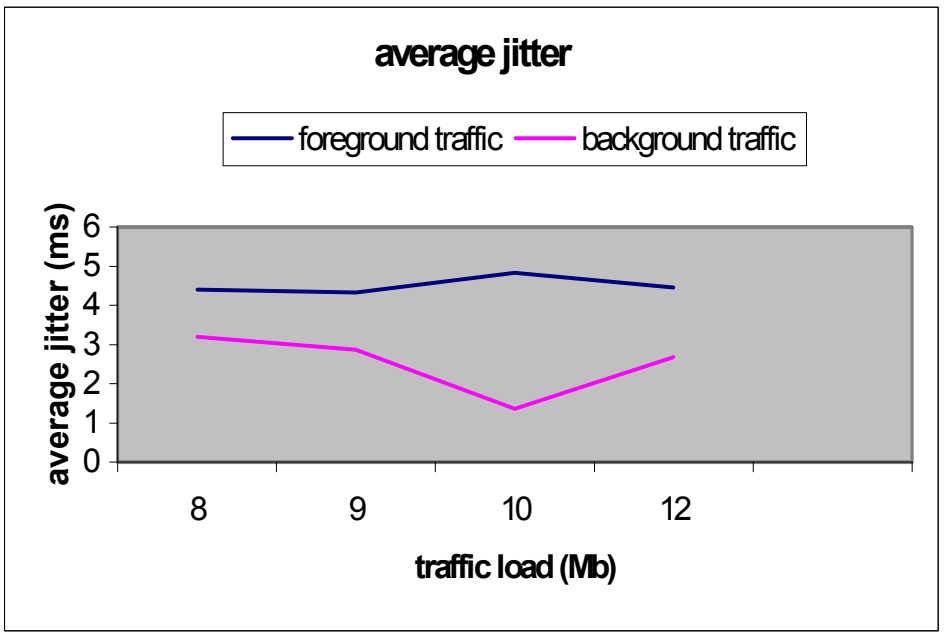



# Experiments on EF based service

- Enabled classification mechanism (using IPv6 access lists)
- Enabled prioritization mechanism (using CBWFQ, the priority mechanism)
  - It uses Low latency Queue for the classified traffic
- Generated traffic
  - Background
    - TCP and UDP on various rates by Iperf
  - Foreground
    - UDP generated by Iperf (250Kbps or 500Kbps)

# Experimental results



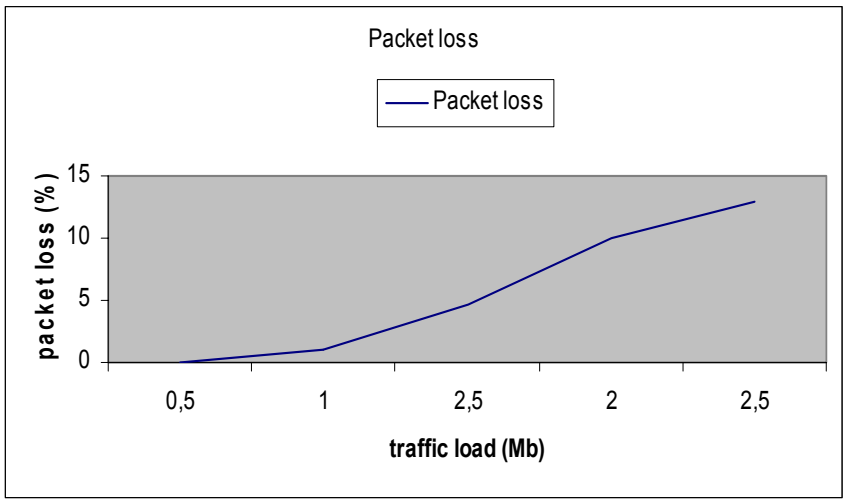



# Testing the EF based QoS service

- The packet loss graphic displays efficient operation
  - The graphic is created with results only for UDP traffic
  - For results with TCP and UDP background traffic are similar
- The jitter is significantly less on foreground traffic
  - Foreground traffic uses longer path to the destination (crosses CTI's production network and is calculated between the 2 ends)
  - If we measure it on the same path with the background's traffic it is very low

# Testing the upper bound of prioritization mechanism

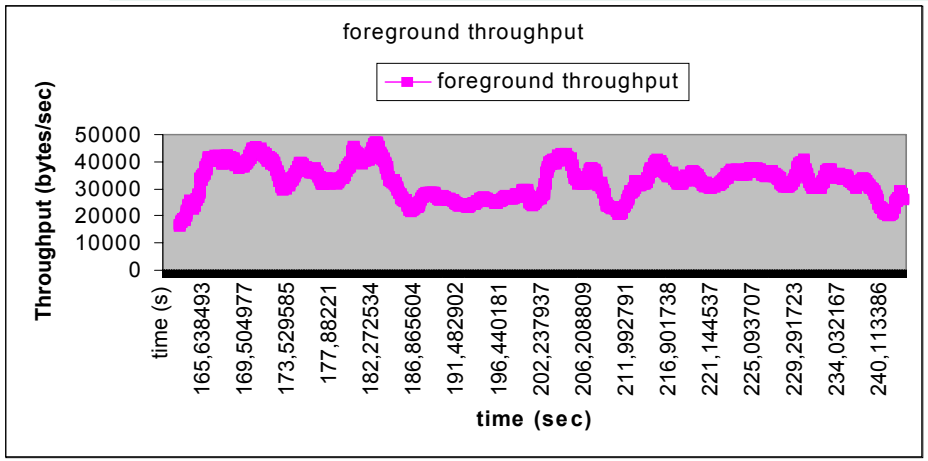
- The CBWFQ mechanism (priority command) had been configured to use only 20% of the bandwidth of the link
- We tried to overload the prioritization mechanism to investigate its performance
- We used UDP traffic (background and foreground), increasing the foreground each time
- The background traffic was 8Mb

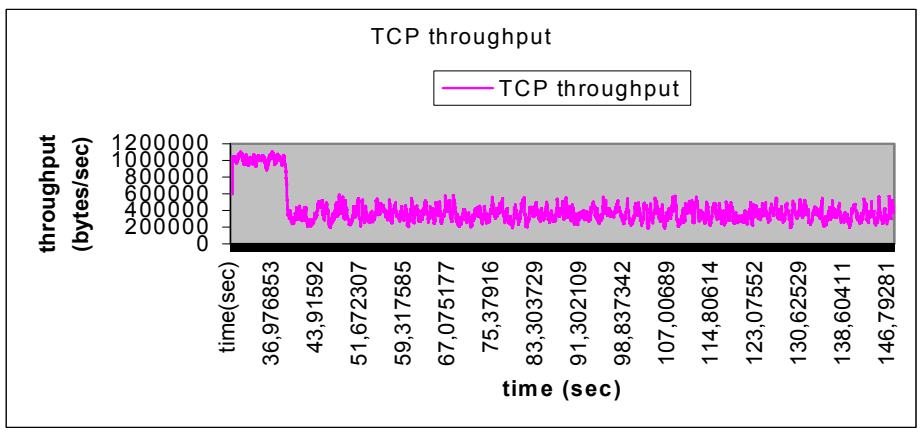


# Testing the EF based service with real time traffic

- Performed tests with real time traffic (by OpenH323)
  - Background traffic
    - Mix of TCP and UDP traffic generated by Iperf
  - Foreground traffic
    - Real time traffic generated by openphone (on testing scenario)
    - Real time traffic generated by openphone (on testing scenario) and additionally UDP traffic generated by Iperf (300Kbps)
- We want to check:
  - Throughput of foreground traffic and of TCP's background traffic
  - Quality of videoconference data

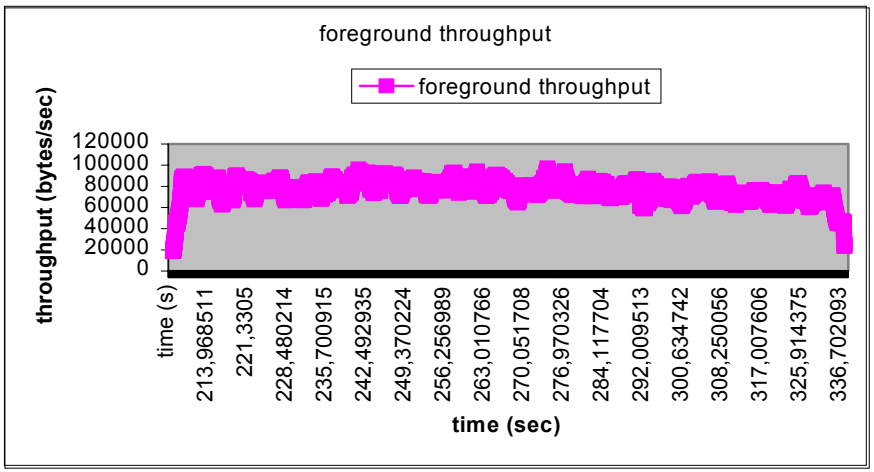
# Results with real time data (scenario 1)

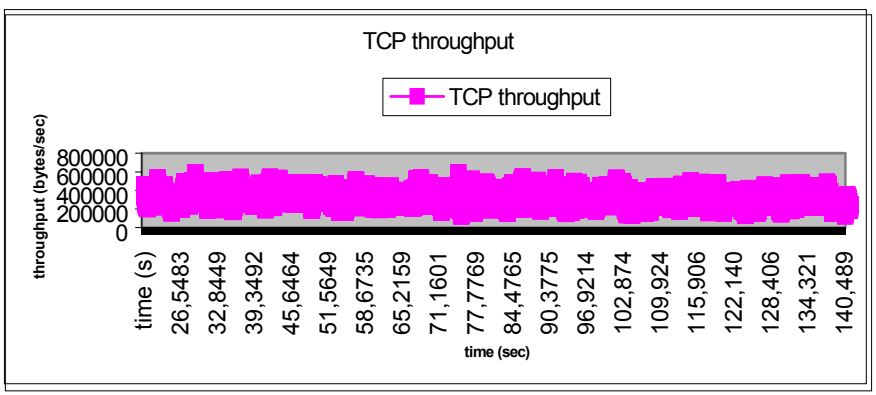



- Videoconference:
  - excellent quality
  - Few packet losses
  - Average throughput 300Kbps
- Background traffic
  - UDP: tries to earn bandwidth from the remaining
  - TCP: adjusts its rate to the remaining bandwidth

# Results with real time data (scenario 2)



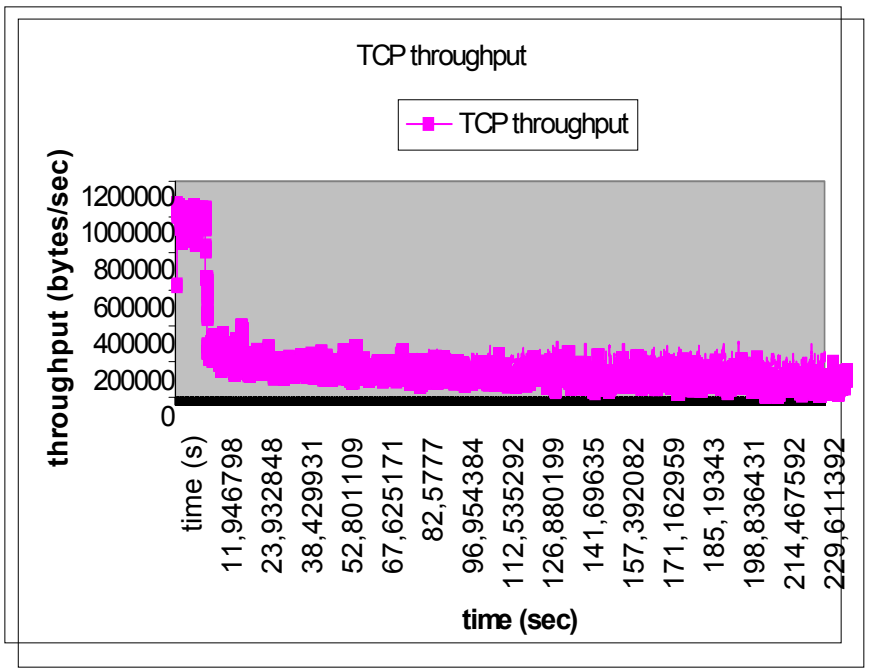

- Foreground traffic
  - Videoconference:
    - excellent quality
    - Few packet losses
    - Average throughput 300Kbps
  - Additional UDP traffic (300Kbps)
- Background traffic
  - UDP: tries to earn bandwidth from the remaining
  - TCP: adjusts its rate to the remaining bandwidth

# Investigation of WRED mechanism

- WRED mechanism
  - Min threshold, max threshold, dropping possibility
  - Investigate its impact on foreground traffic
  - Investigate its impact on background traffic
- Performed 2 testing scenarios
  - 1st scenario:
    - Minthreshold = 30, maxthreshold = 50, dropping possibility = 10%, max queue size = 75 packets
  - 2nd scenario:
    - Minthreshold = 55, maxthreshold = 75, dropping possibility = 10%, max queue size = 75 packets

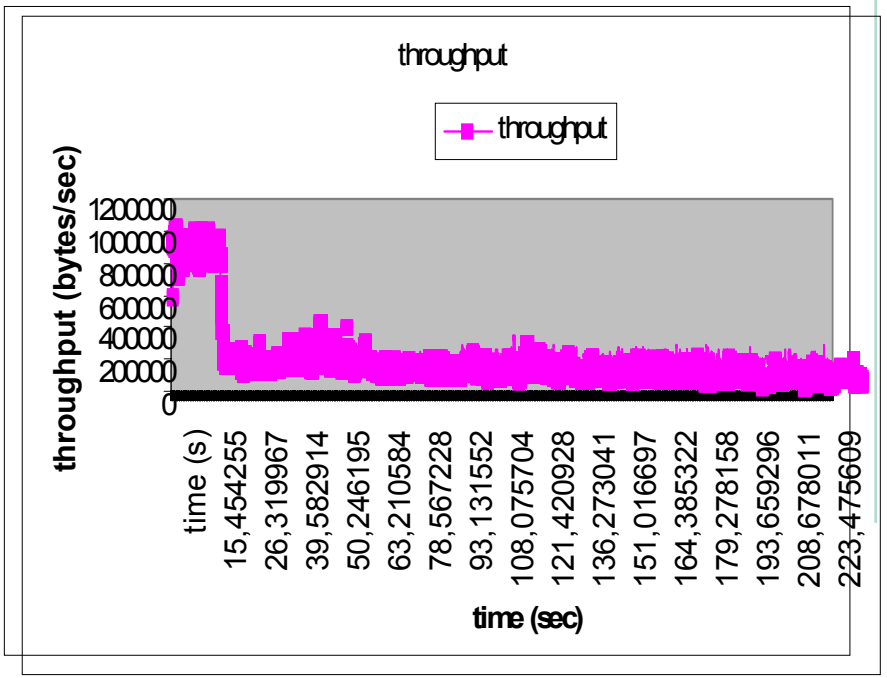
# Results for WRED (scenario 1)

- Foreground traffic
  - Real time data (OpenH323) & additional UDP traffic (700Kbps)
  - Excellent quality of videoconference
- Background traffic
  - UDP traffic had many packet losses (2%)
  - TCP throughput was also reduced compared to previous experiments



# Results for WRED (scenario 2)

- Foreground traffic
  - Real time data (OpenH323) & additional UDP traffic (700Kbps)
  - Excellent quality of videoconference
- Background traffic
  - UDP traffic had less packet losses (0.90%)
  - TCP straggled less
- No significant impact on the foreground traffic



# Overall - Conclusions

- Our contribution:
  - Configured and tested a QoS service for real time applications in IPv6 networks
    - Classification
    - Queue management: CBWFQ, the priority method (low latency queue)
    - Measured: average throughput, packet loss, jitter
  - The service was tested with real time traffic and produced very good quality on videoconference (almost zero packet loss, low jitter)
  - Investigation of WRED mechanism
    - Better performance if thresholds approach the max queue size

# Future Work

- Also test policing mechanism on the above QoS service
  - Investigate the proper configuration of policing profile for real time data
- Test WRED mechanism when foreground traffic approaches the upper bound of bandwidth of the priority command
- Investigate the bandwidth command of CBWFQ and compare it with priority
- Extend the tests using bigger network topologies (more hops)
- Test additional QoS features that will perhaps be supported for IPv6 on future network devices and software

# Thank you

Email Info:

Christos Bouras (bouras@cti.gr)

Apostolos Gkamas (gkamas@cti.gr)

Dimitris Primpas (primpas@cti.gr)

Kostas Stamos (stamos@cti.gr)