



the globus project  
www.globus.org

# Requirements for a Bandwidth Broker Supporting High-Performance TCP Flows

Volker Sander

sander@mcs.anl.gov

Argonne National Laboratory

2/9/2000



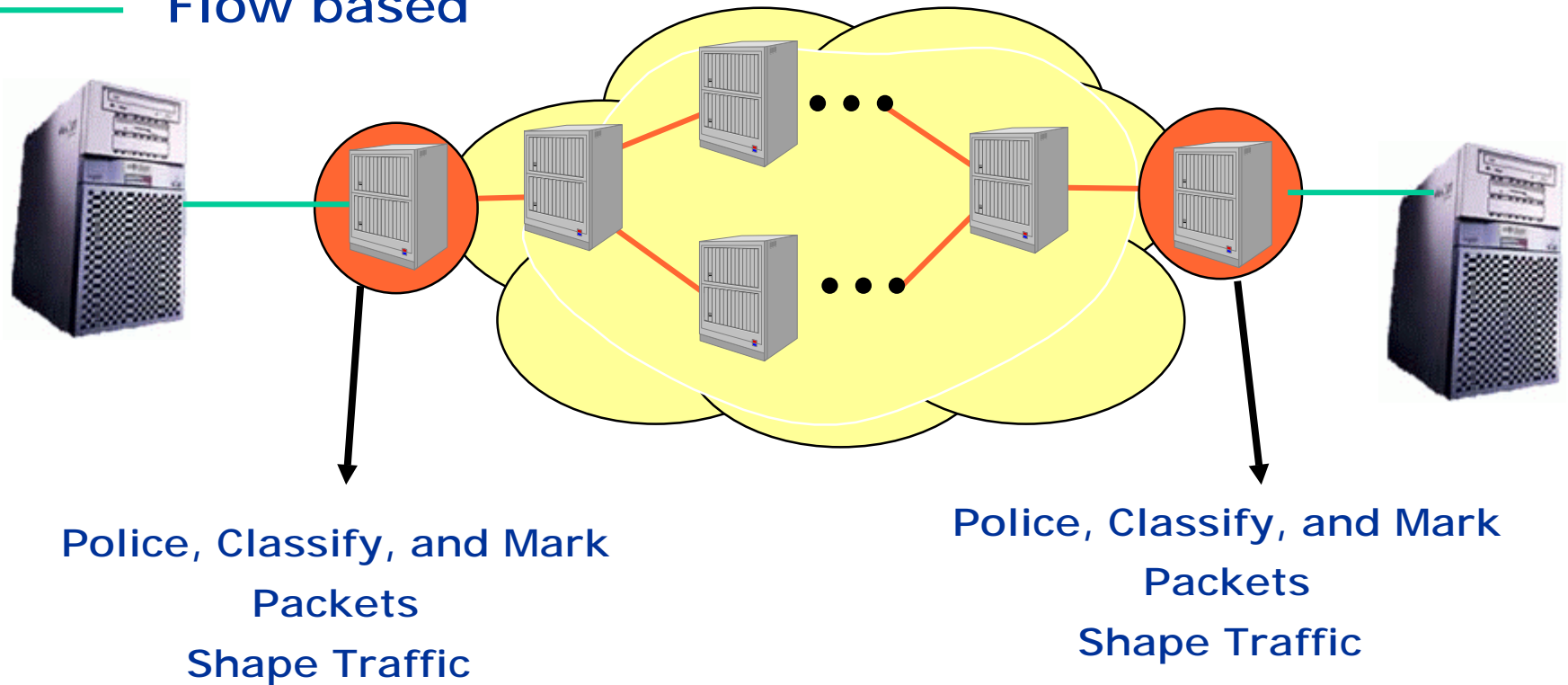
# Outline

- Implementing EF PHB
- The Classical Evaluation
- The TCP Challenge
- Results
- Configuration
- Advanced Issues
- Conclusions / Future Plans



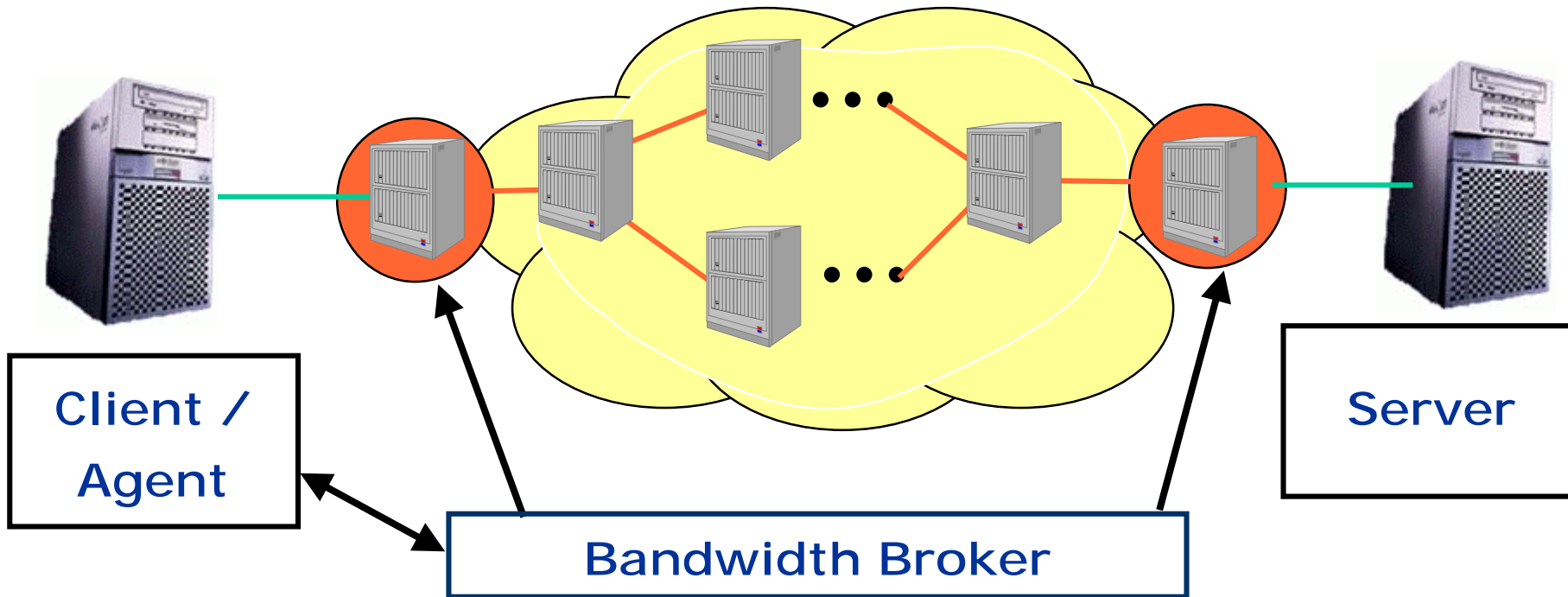
# The Basic Diffserv Concept

- Aggregate based scheduling implements PHB
- Flow based



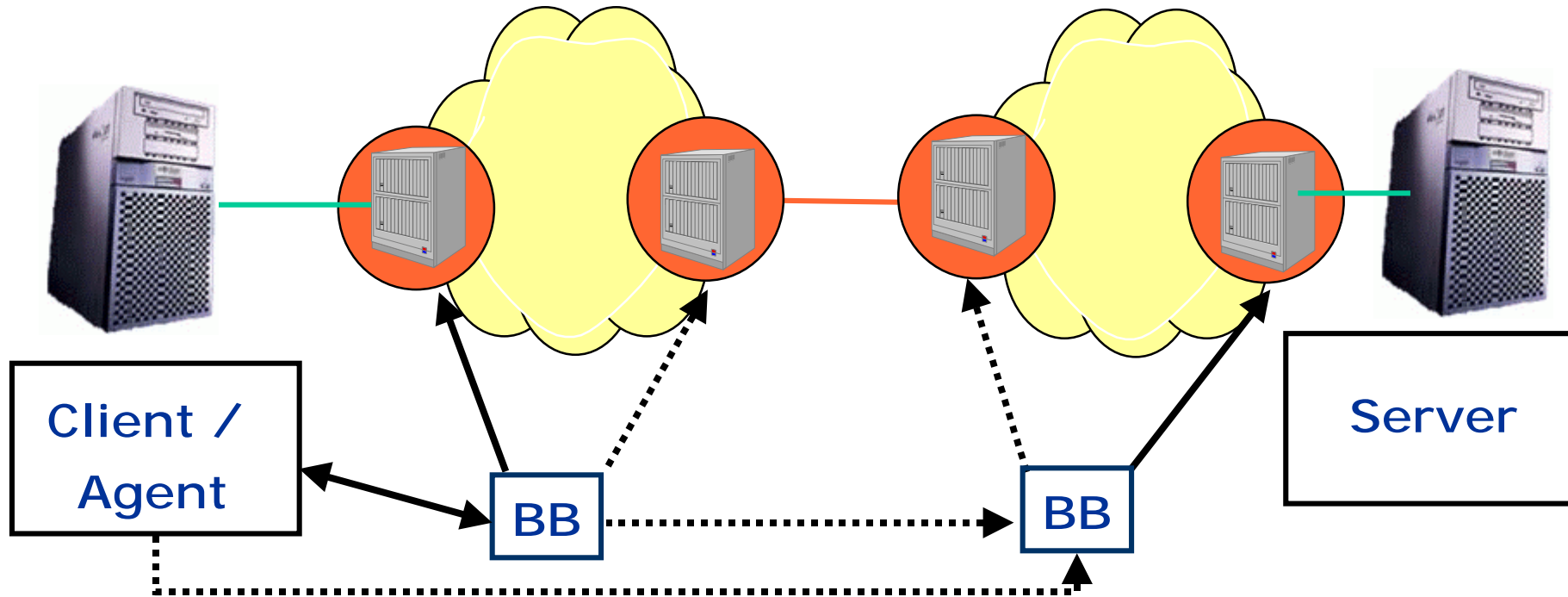


# The Bandwidth Broker





# The Multidomain Extension





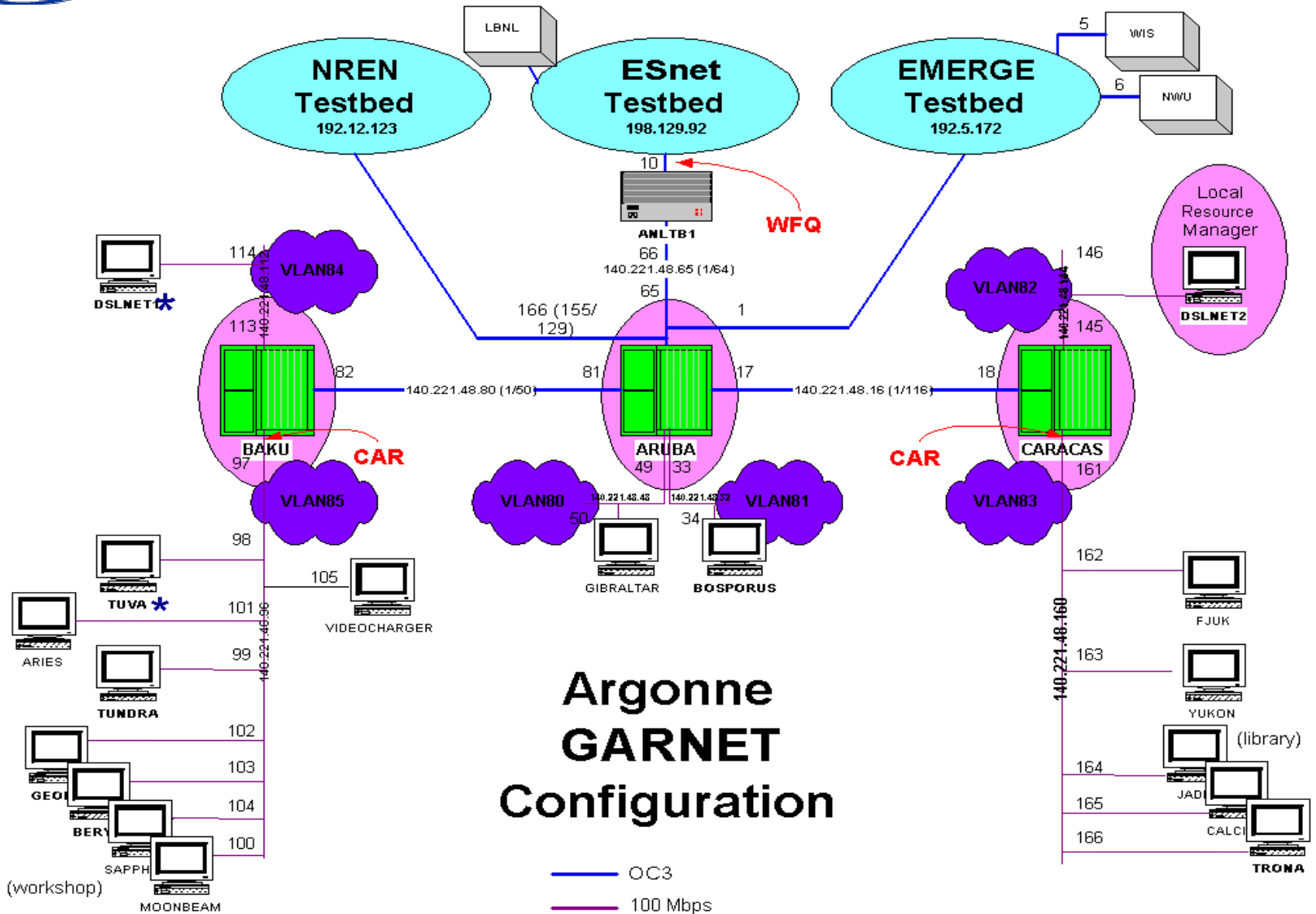
# Evaluation Tools

- UDP Traffic Generator
    - Modified Version of Andy Adamson's gen\_send and gen\_rcv
      - > Evaluate admission control
      - > Creating competing traffic
    - MGEN/Drec
      - > Evaluate delay and Jitter for Premium UDP Flow
    - IPERF
  - Modified Version of ttcp
    - GARA-enabled (wait for reservation)
    - Support for a desired application rate
    - Consecutive bandwidth reporting
    - Bulk transfer ttcp
-



# Basic Experiments

- Evaluate Admission Control
- Evaluate UDP Flows under Congestion
  - Delay
  - Jitter
  - Influence of traffic shaping
- Evaluate Premium TCP Flows under Congestion
  - Constant throughput
  - Affect of QoS on RTT
  - Influence of traffic shaping
- Evaluate Inter-Domain Issues

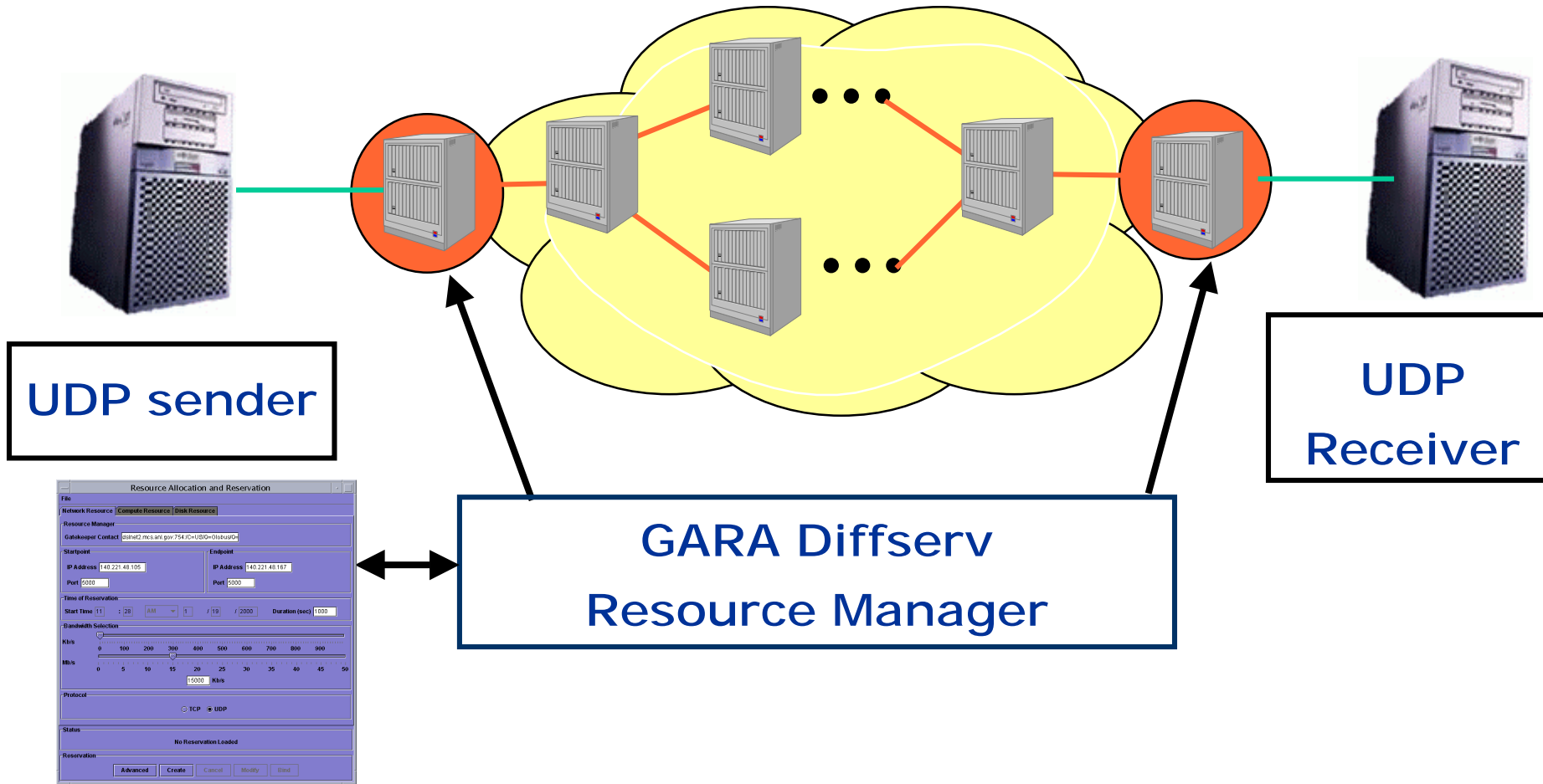


# Argonne GARNET Configuration

- OC3
- 100 Mbps



# Basic Experiment I

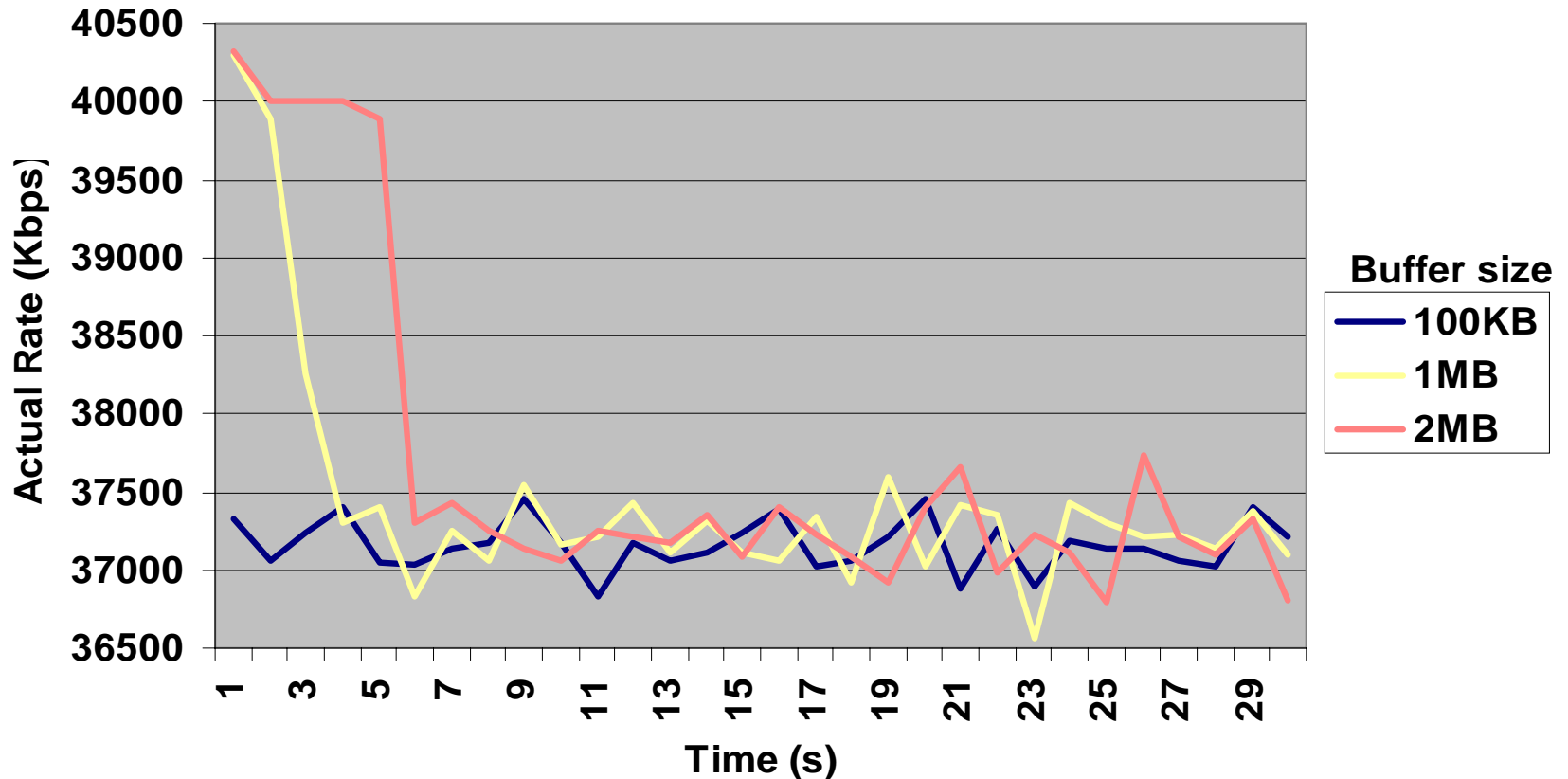




# Basic Experiment I

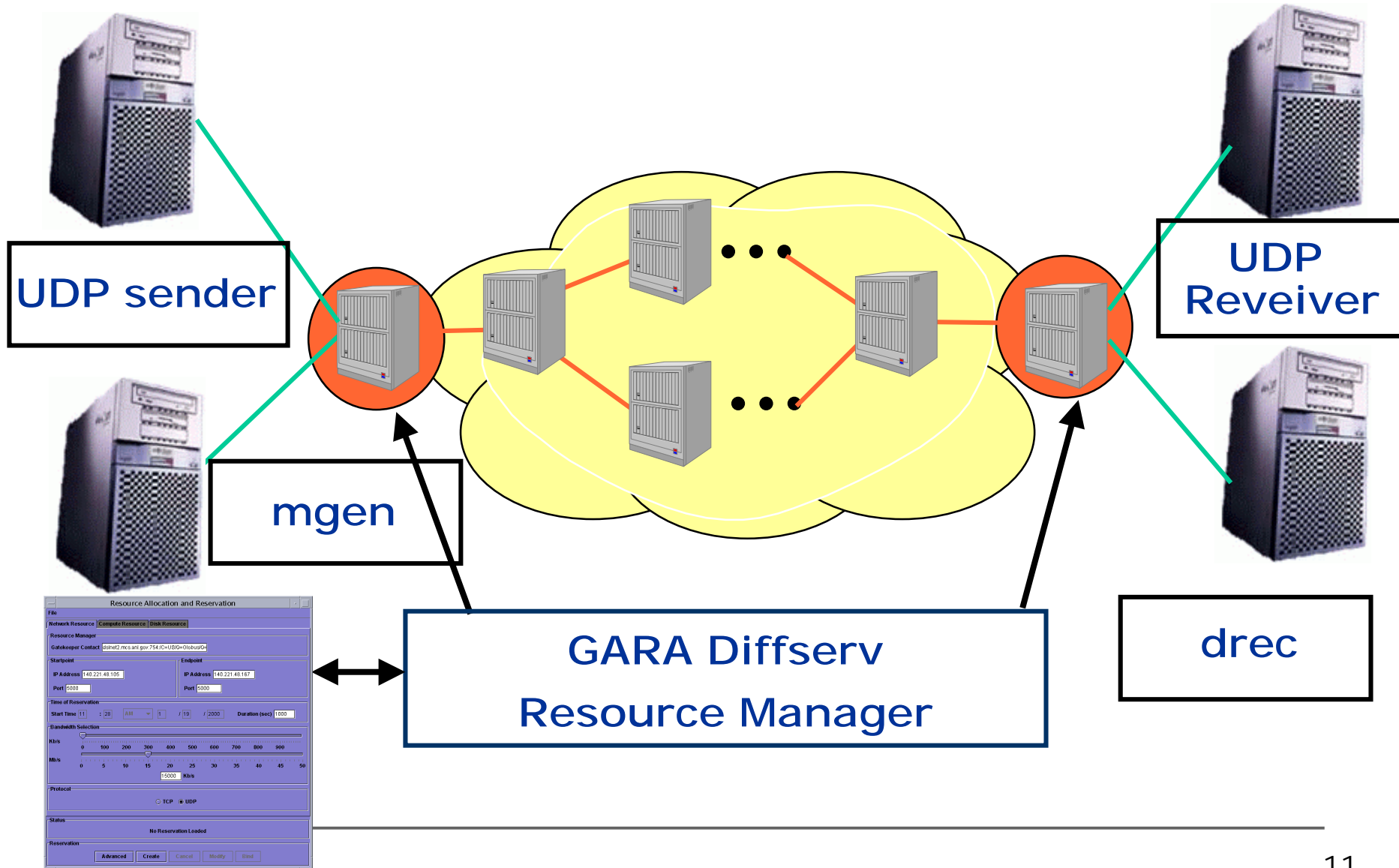
- Goal: Proof of Admission Control

UDP Rate comparison for different CAR buffer sizes  
(Desired rate: >40Mbps; Average rate limit 40 Mbps)  
Packet size 530 Bytes





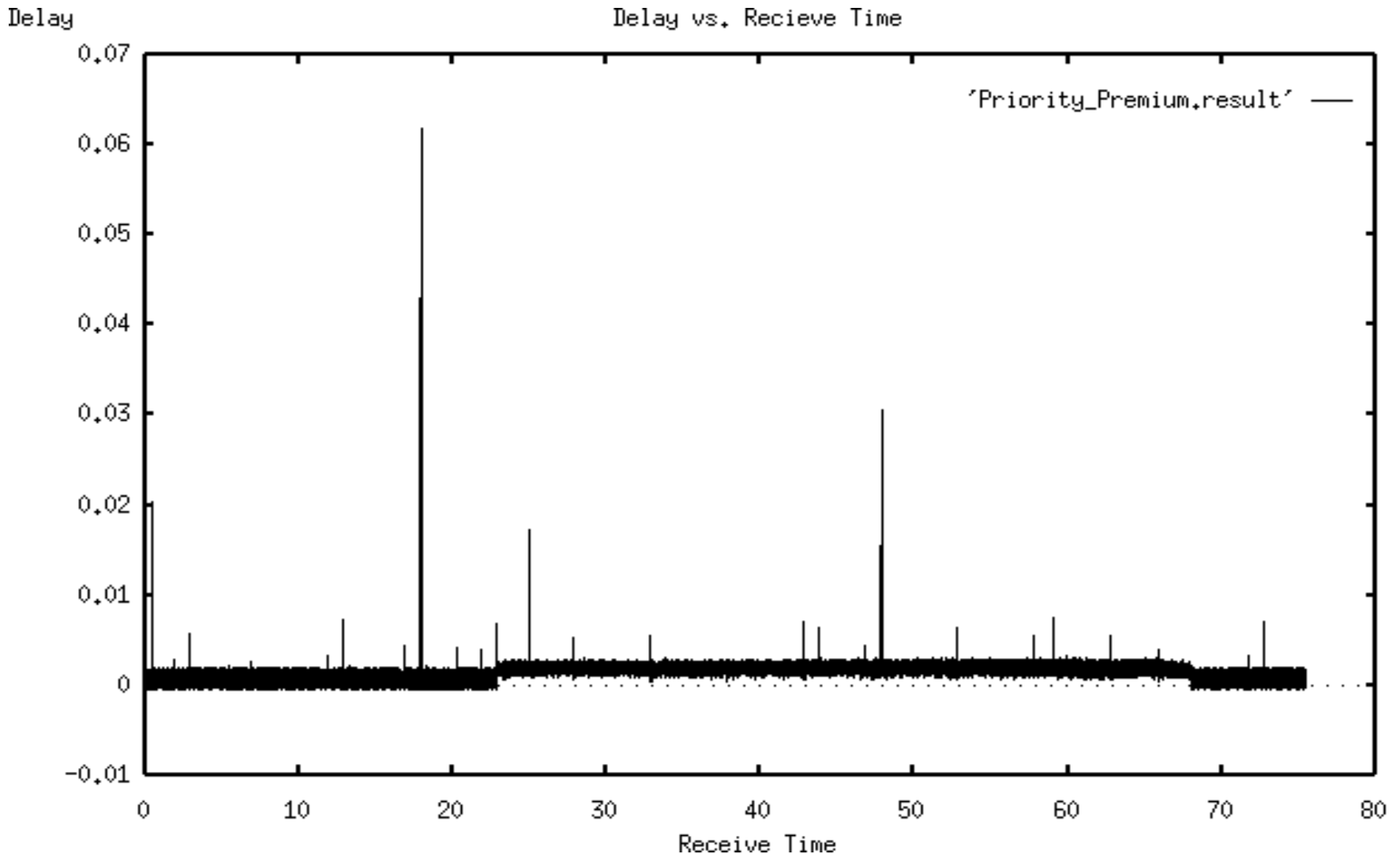
# Basic Experiment II





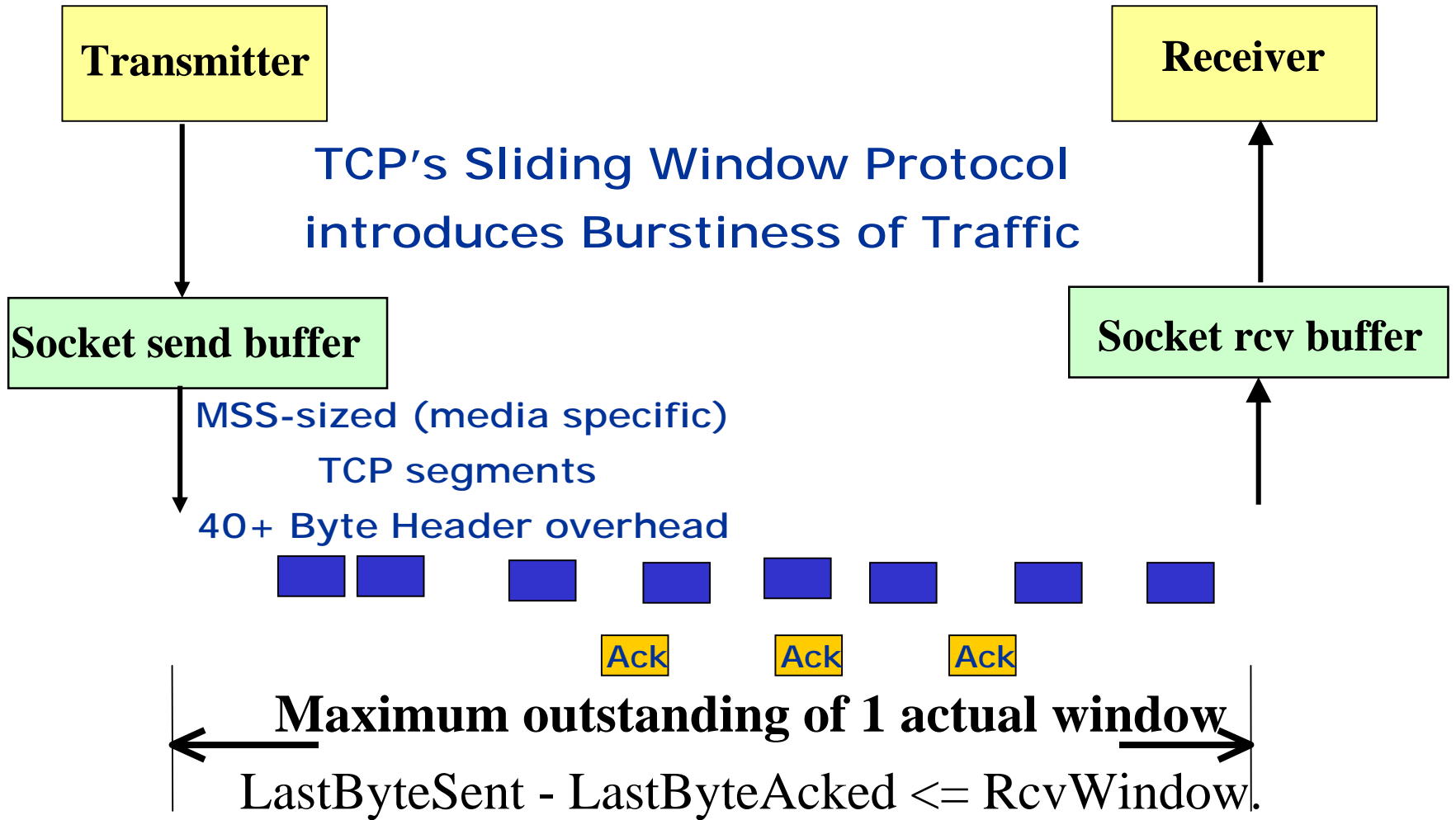
# Basic Experiment II

- Goal: Demonstrate Low-Latency for UDP flows





# TCP and EF PHB...



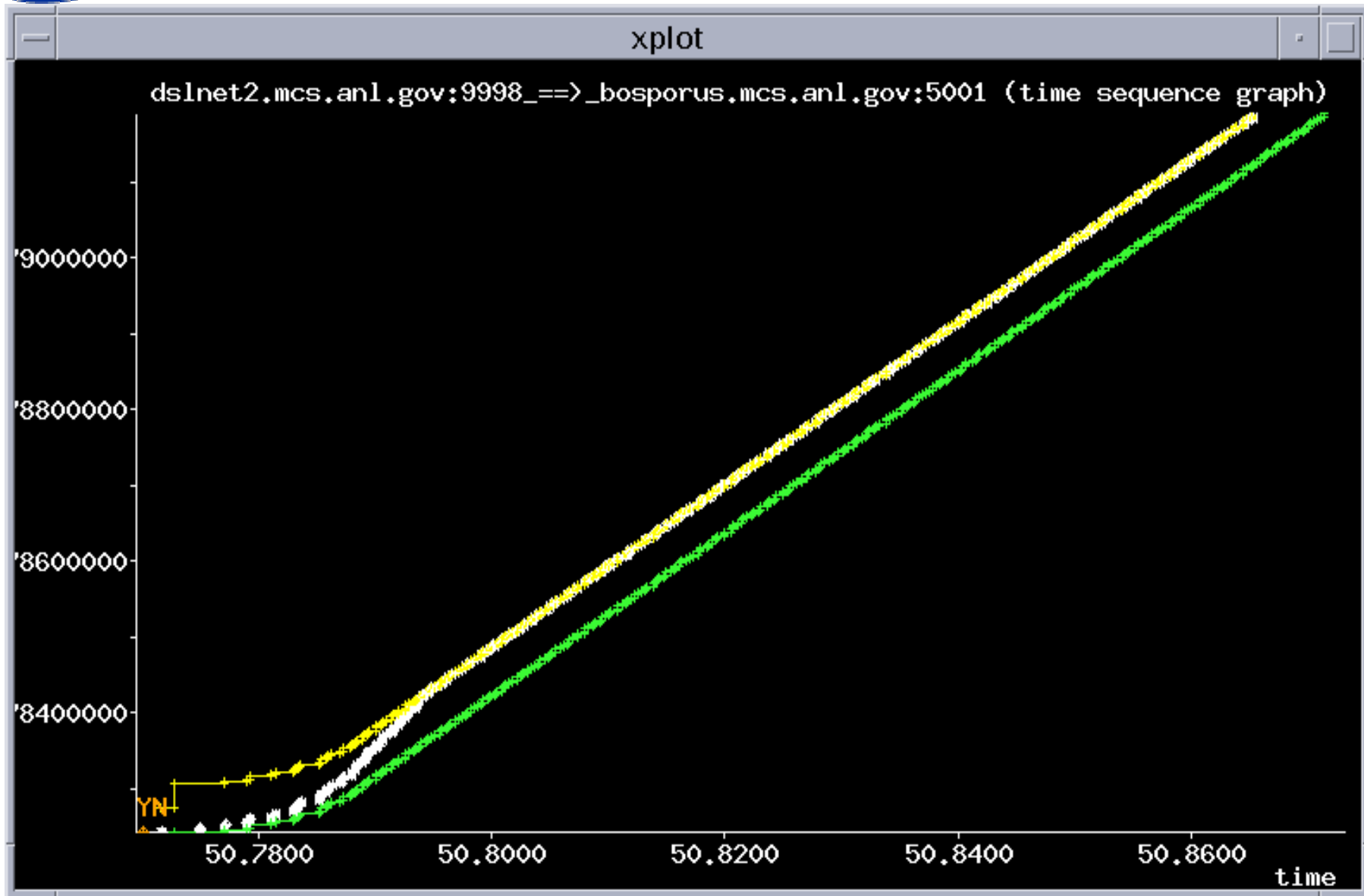


# Microscopic Behavior of TCP Flows

- Throughput is limited by
  - Socket-buffer size (Bandwidth\*Delay Product)
    - > RTT is important
    - > Impact of Queuing (correlates somehow to Low-Latency)
  - Size of Network Pipe
    - > Policing
    - > What to do with exceeding packets
  - Application
    - > Is there enough data in the socket buffer
    - > Write size
- Burstiness depends on this

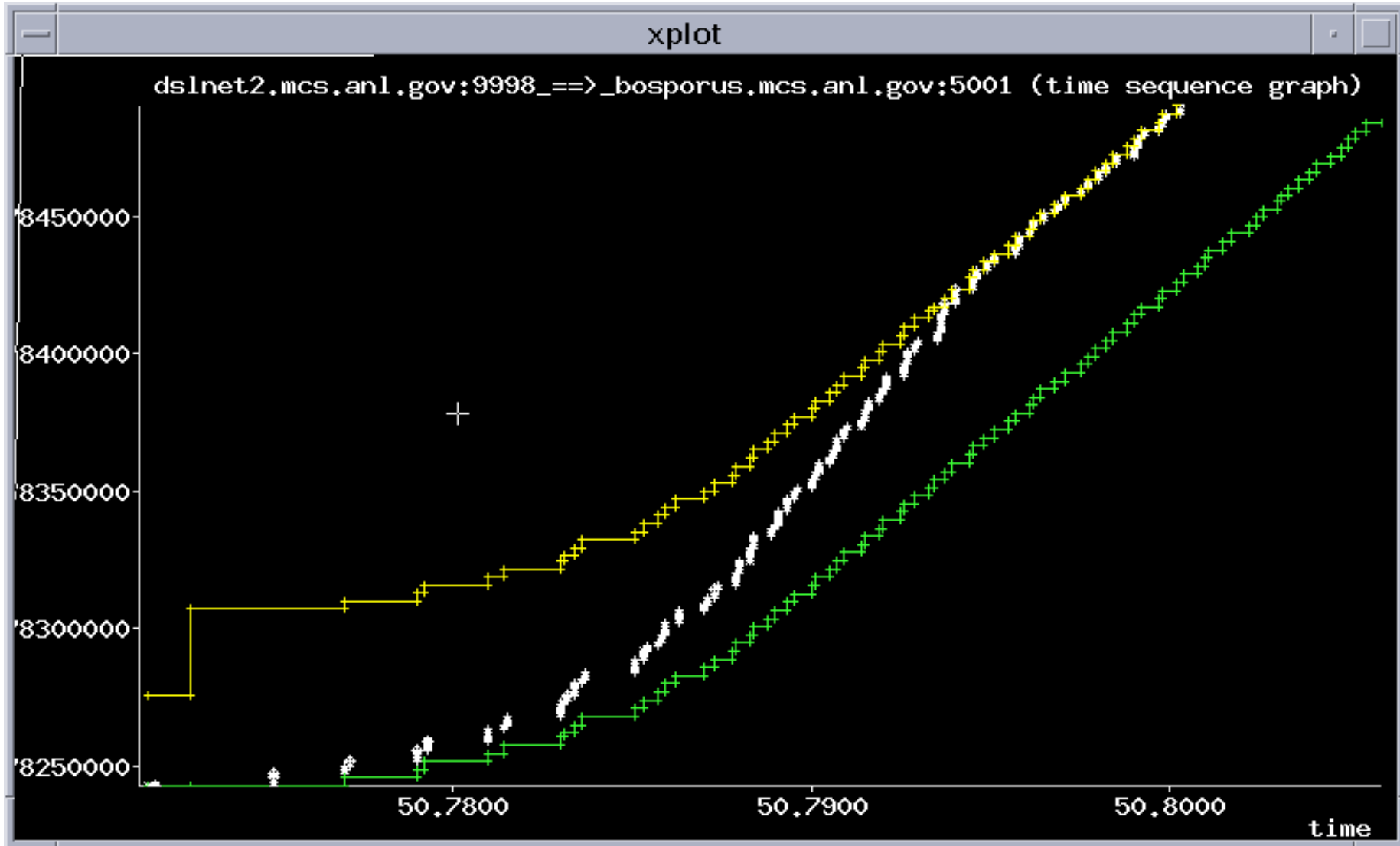


# Socket-Buffer limited Flow



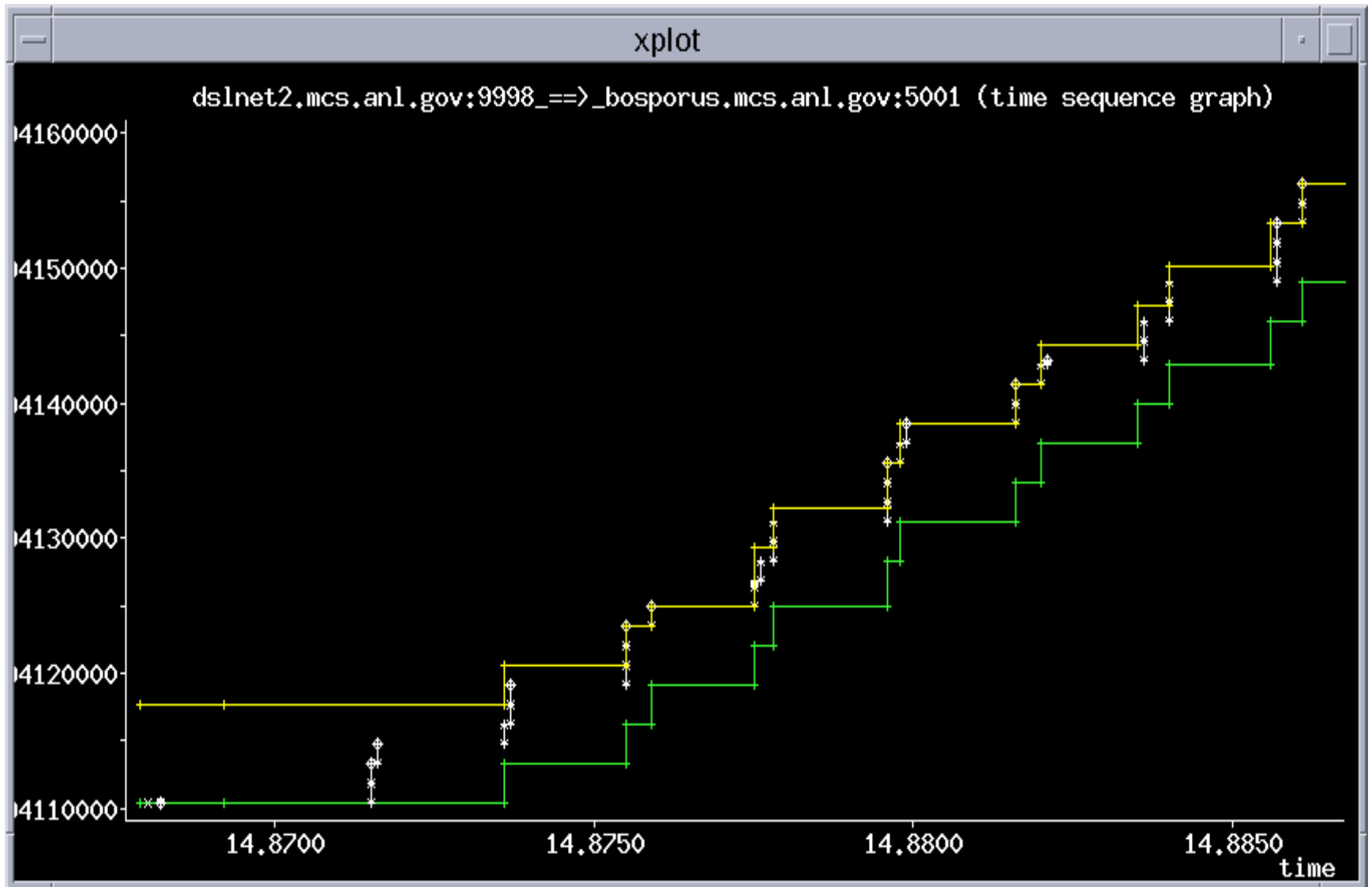


# Socket-Buffer limited Flow



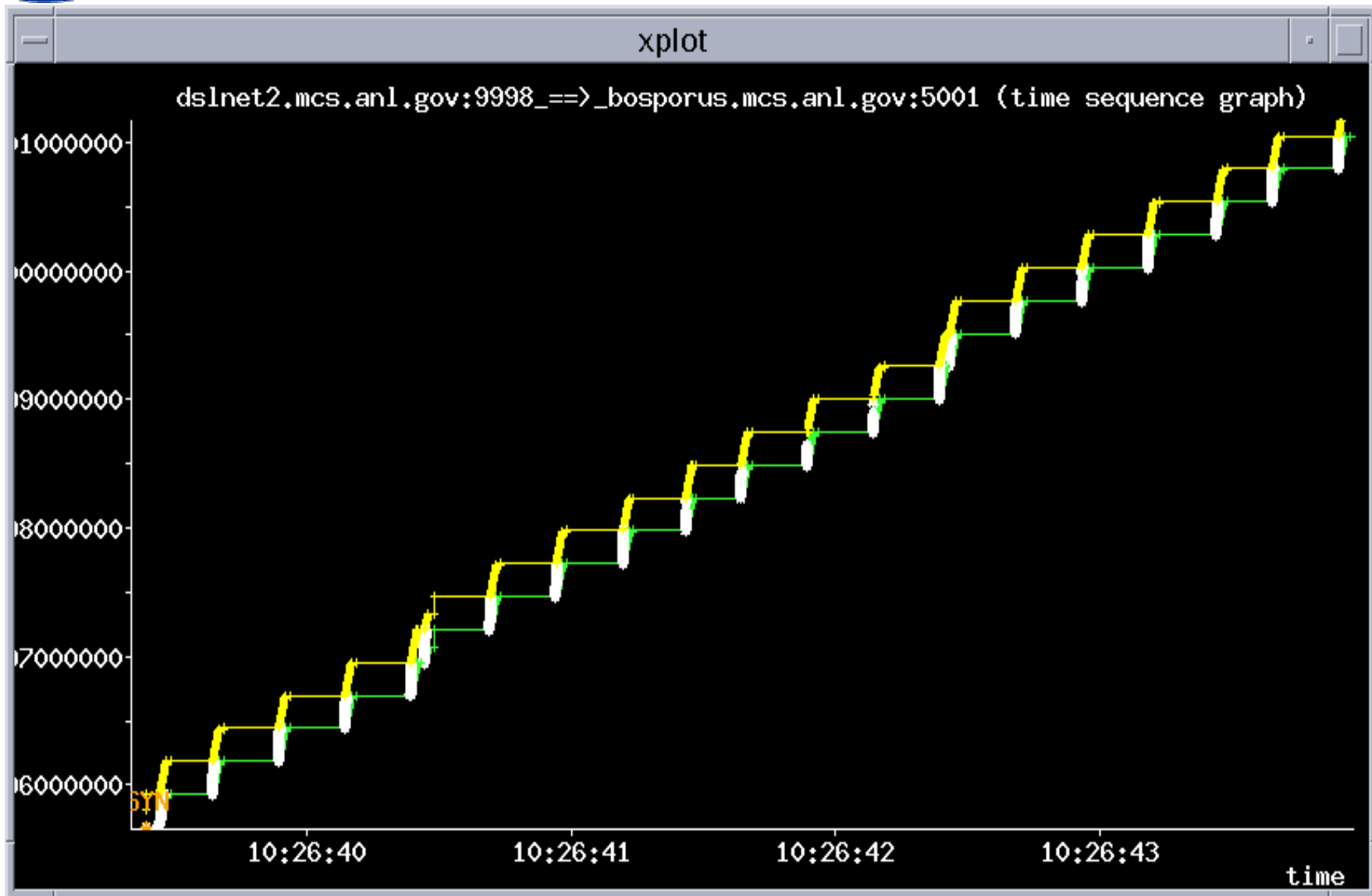


# Socket-Buffer limited Flow



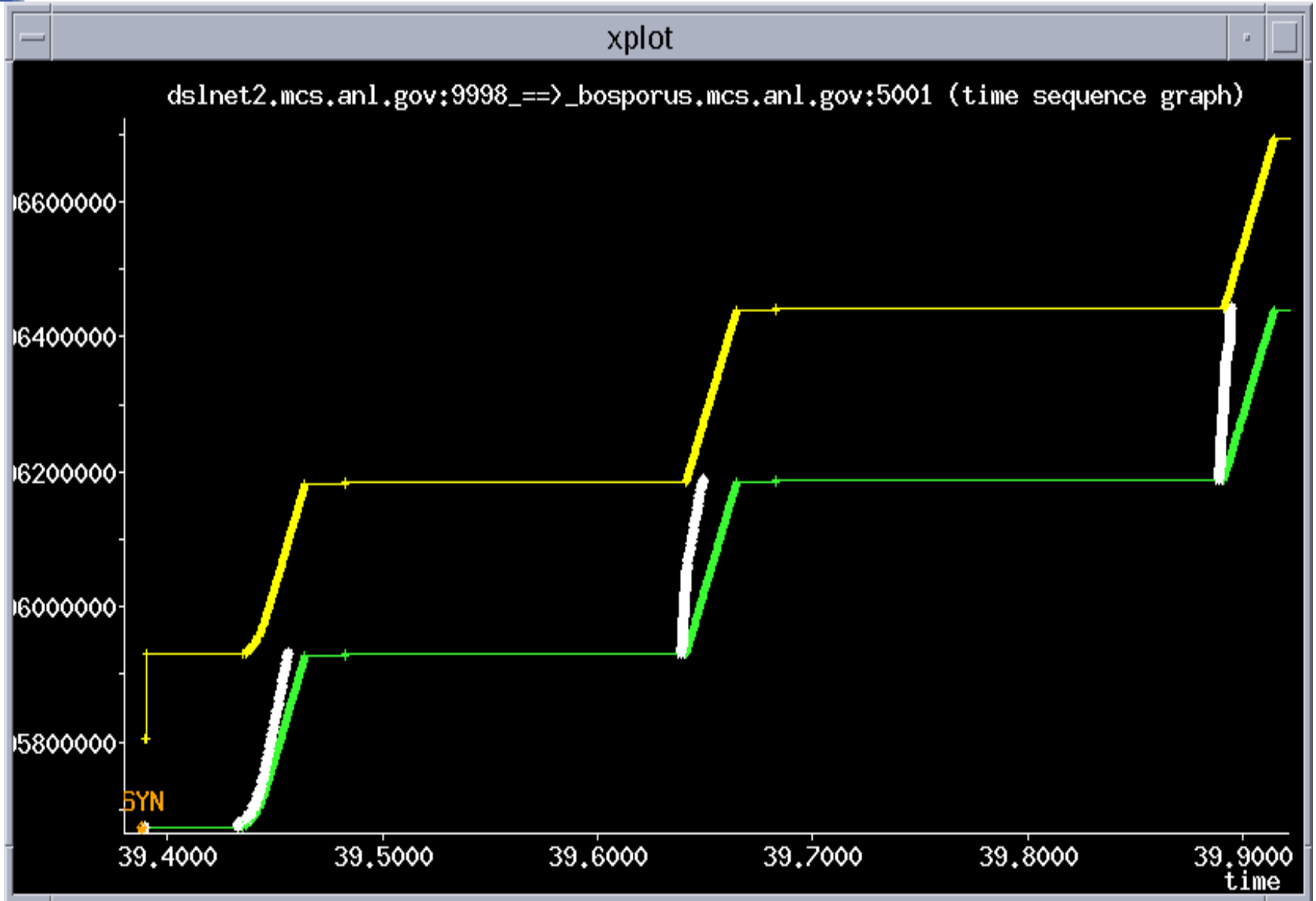


# Application limited Flow





# Application limited Flow





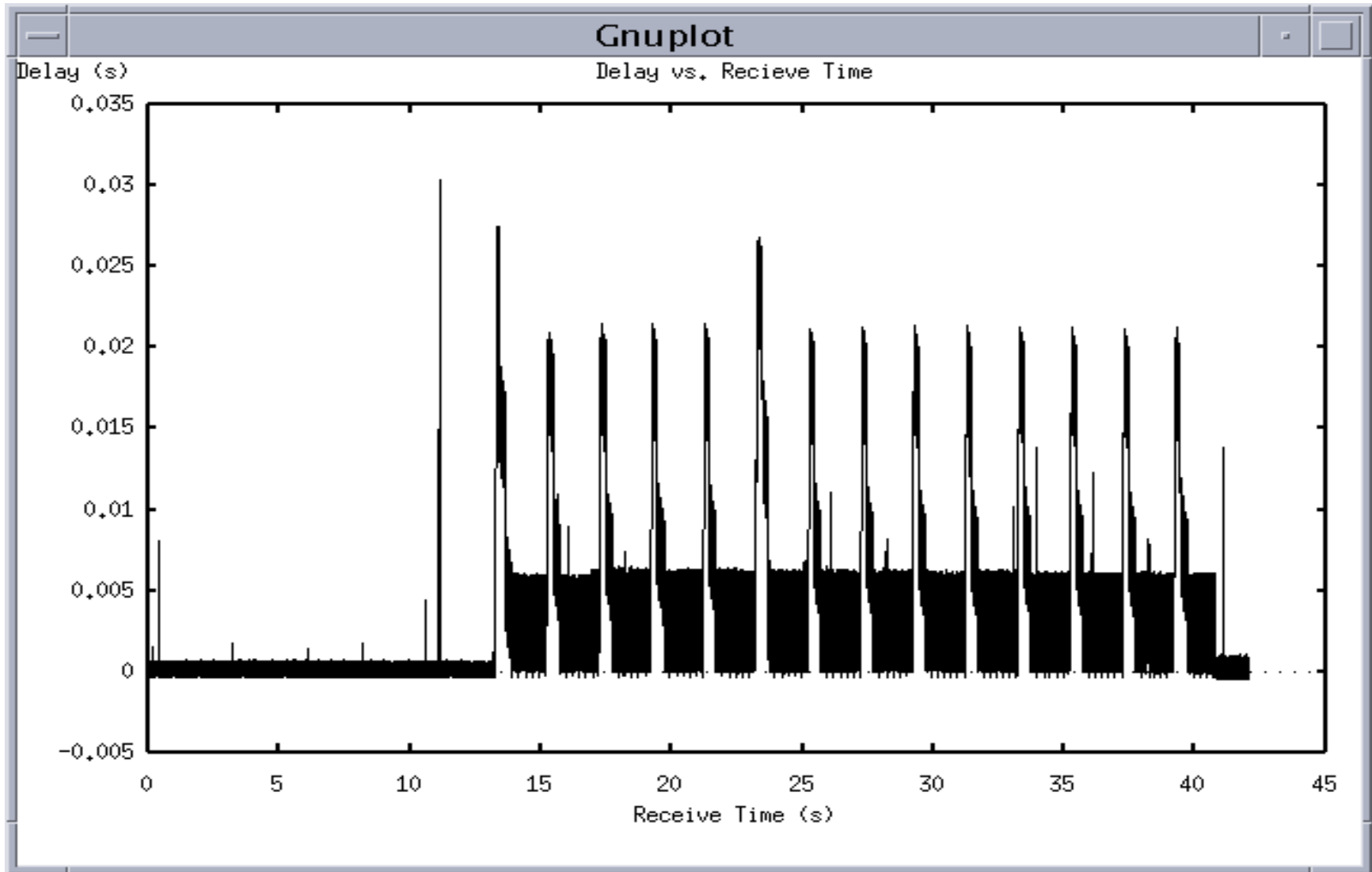
# Conclusions for Implementing a Bandwidth Broker

- Be aware of burstiness
- Token bucket depth should allow a full window burst
  - $T = \text{Reserved\_BW} * \text{Estimated\_RTT}$
- How does this interfere with UDP low-latency flows in one aggregate behavior?



# Sharing of Aggregates

- Shared EF Aggregate with TCP and UDP





- TCP's Flow Control

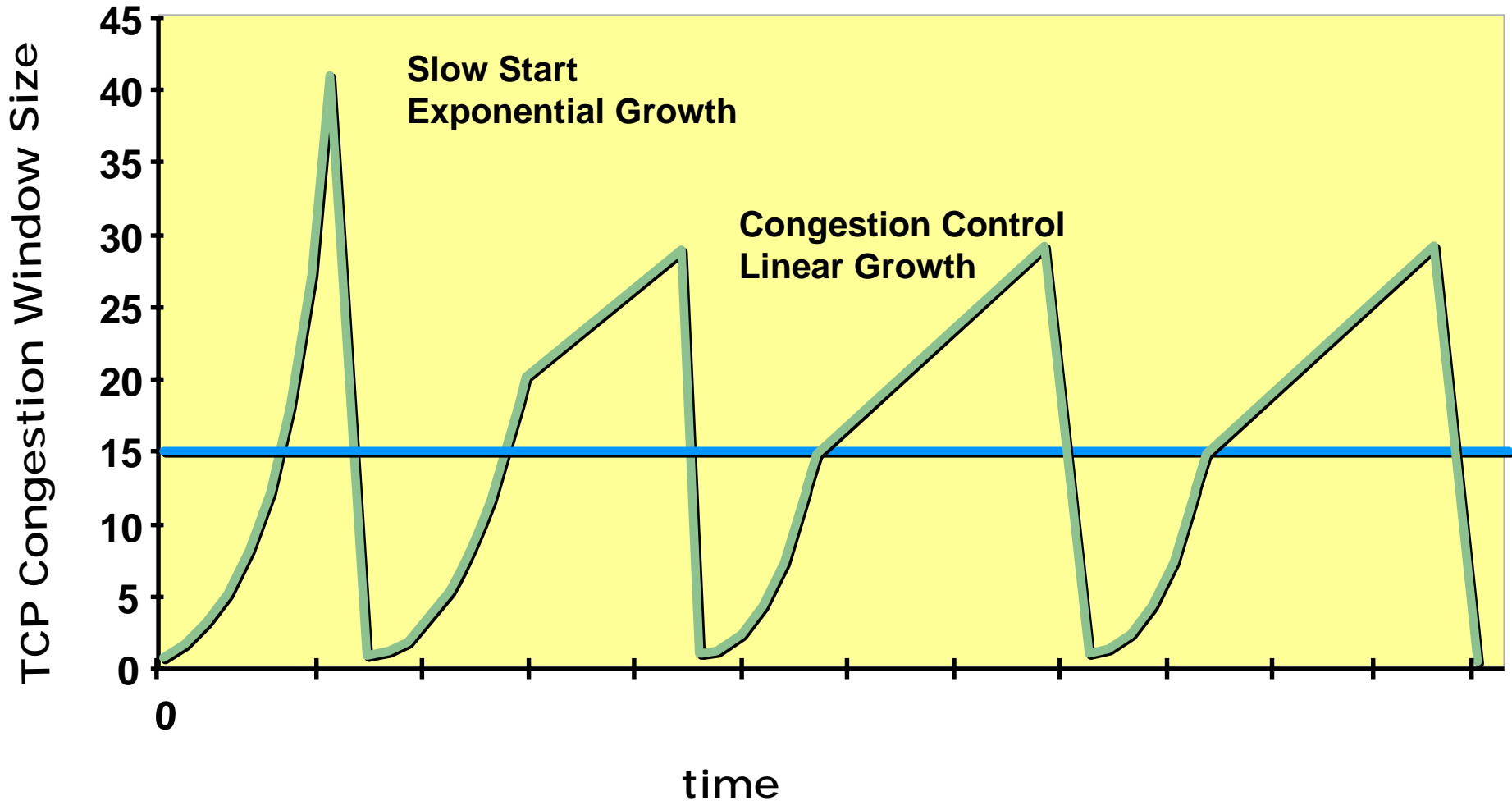
- Traffic might become bursty if the actual window size is large
- Bandwidth\*Latency product as minimum window size

- TCP's Congestion Control

- "Slow Start Phase"
  - > shrink window to 1
  - > increase size with every Ack
- Congestion Avoidance
  - > Half Window size
  - > Increase size by one for every full window roundtrip

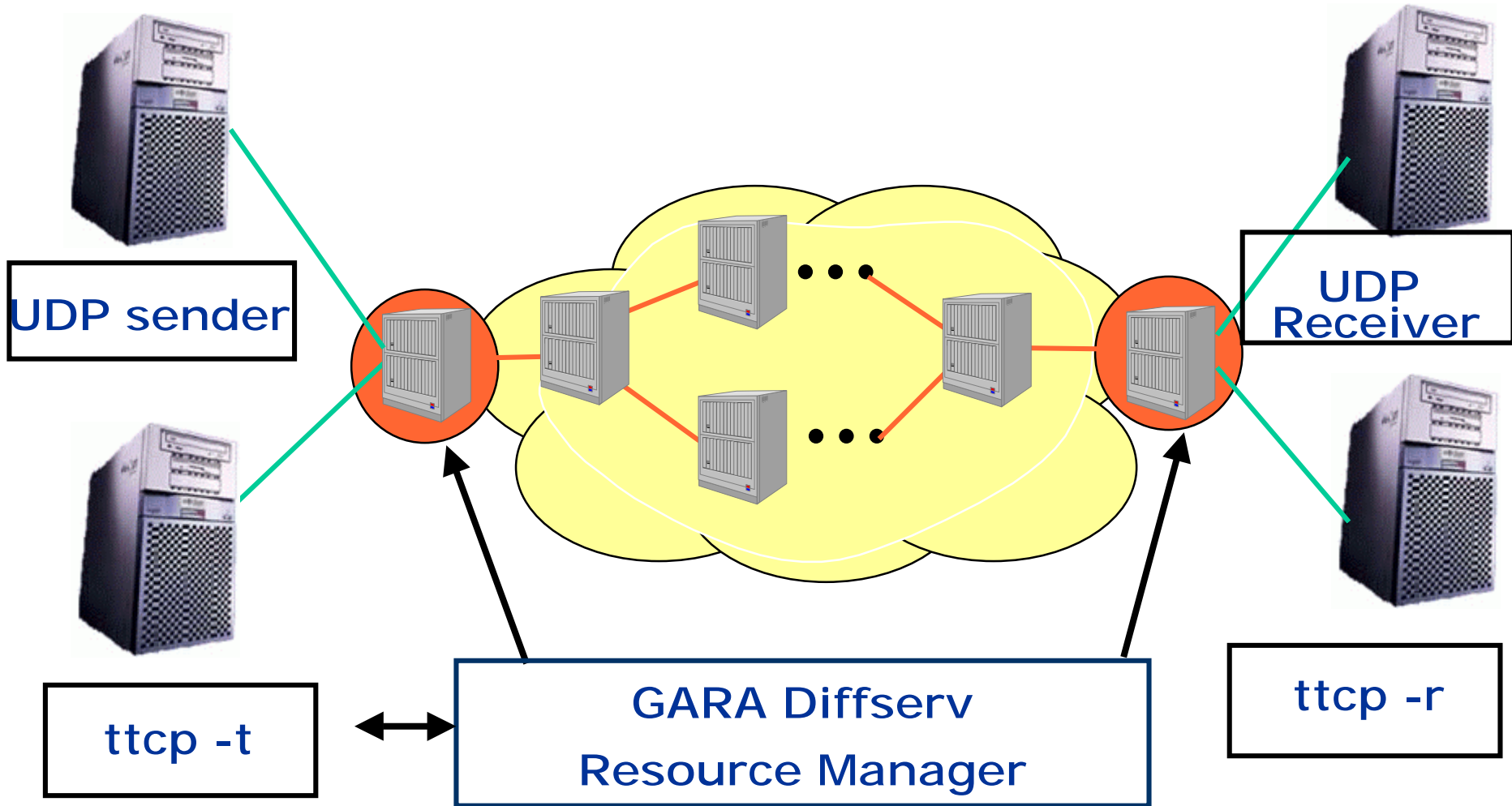


# Exceeding the Reservation



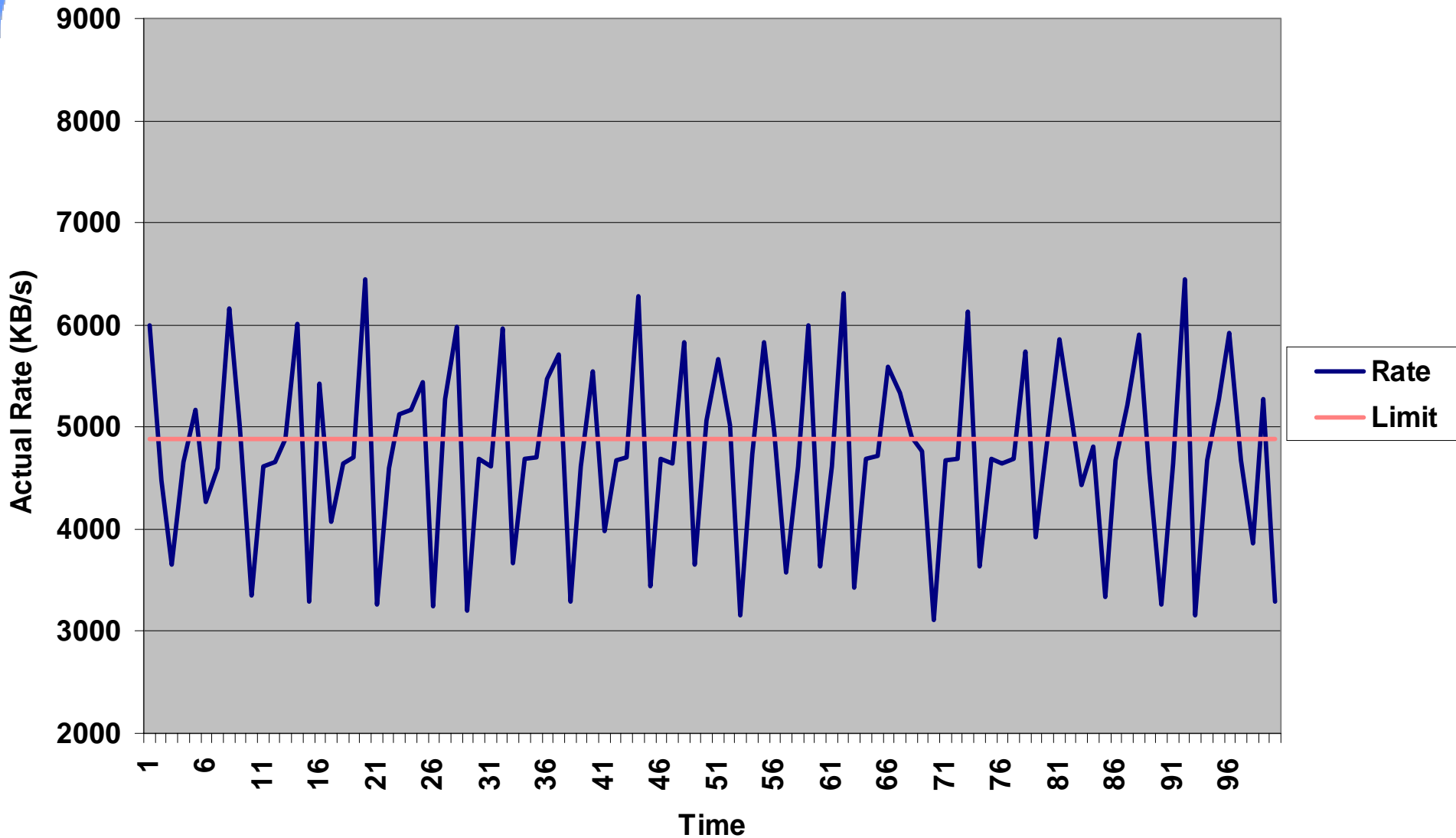


# Basic Experiment III





## Example of Oversubscribing with TCP (Attempted Rate: 6000KB/s, Buffer size: 200000 bytes)





# Conclusions for Implementing a Bandwidth Broker

- Avoid any drops if you care about short-term impact
- Instead use feedback mechanisms to inform the application / the agent to adapt
  - its transmission rate
  - its reservation



# Configuration

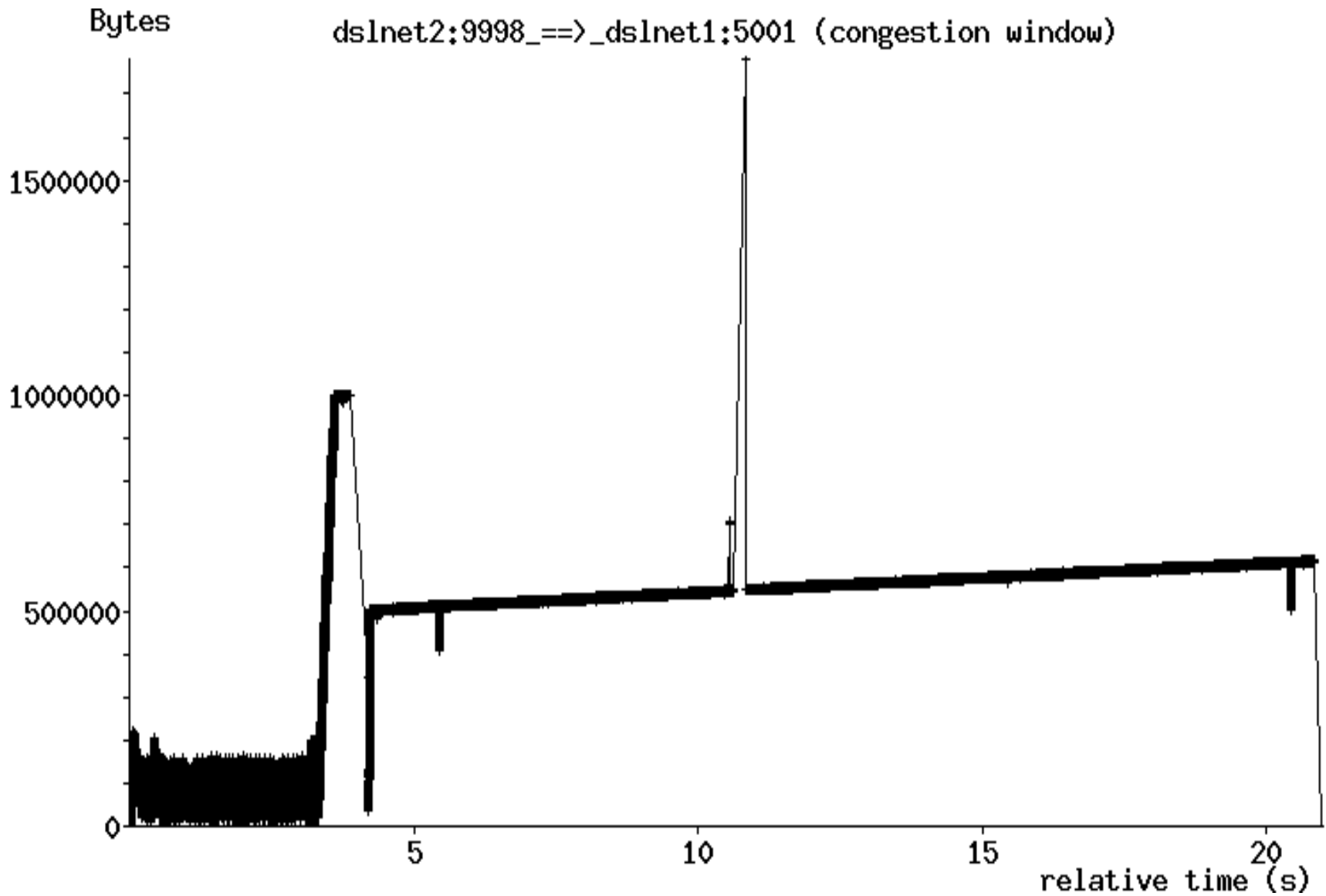
- For TCP the token bucket epth should be calculated by the BB conforming to the estimated RTT and the reservation rate
- The burstiness introduced through TCP can only be addressed through:
  - With Traffic Shaping:
    - ➔ Configure TS only for TCP and non-Latency/Jitter sensitive flows on output of the ingress router
  - Without: Use PQ style scheduling in core network
    - ➔ 99% WFQ BW results in a maximum increase of RTT by  $RTT/2$  (assuming 33% EF-Traffic)



# Some enhanced TCP Issues...

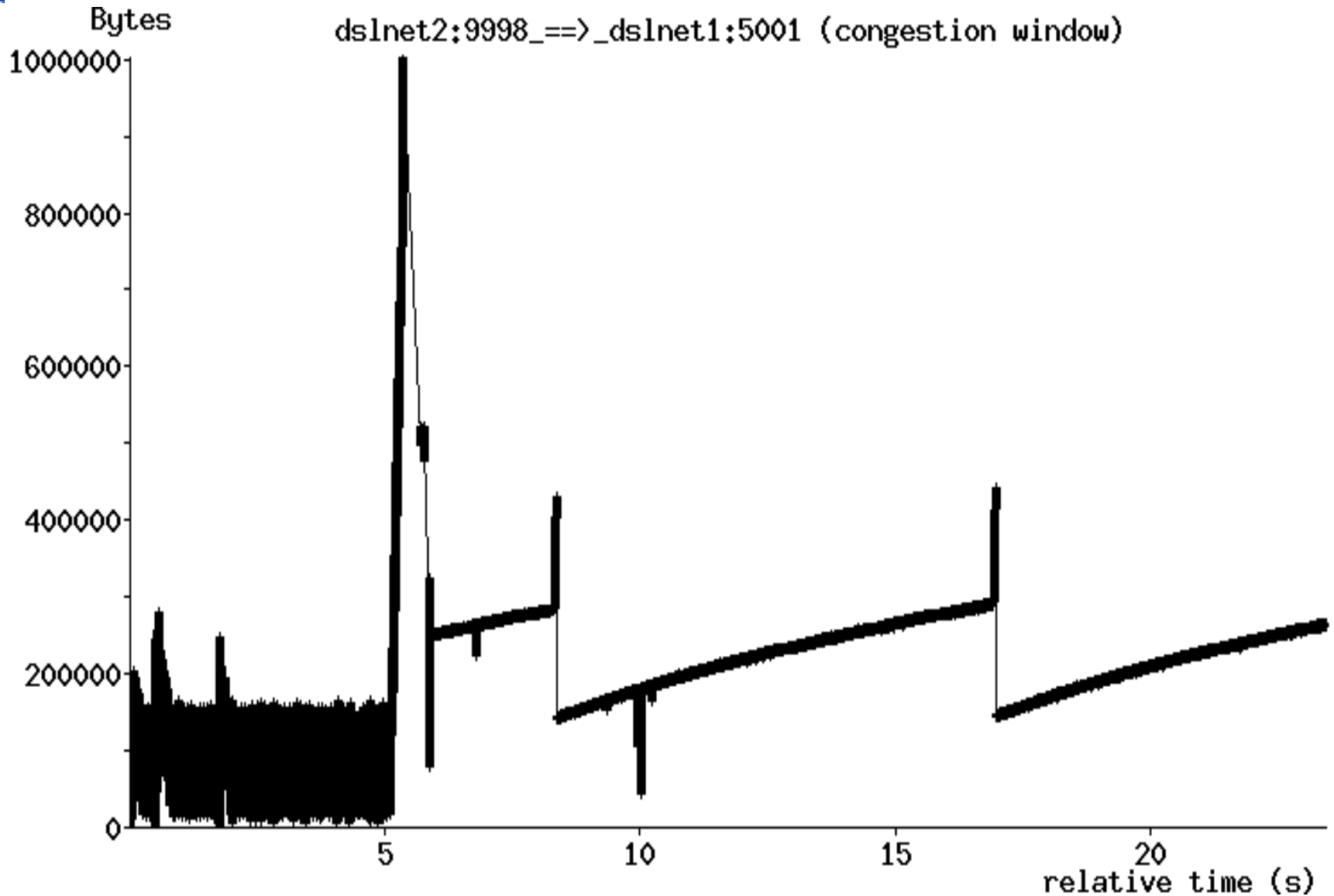


# Standard BE Behavior



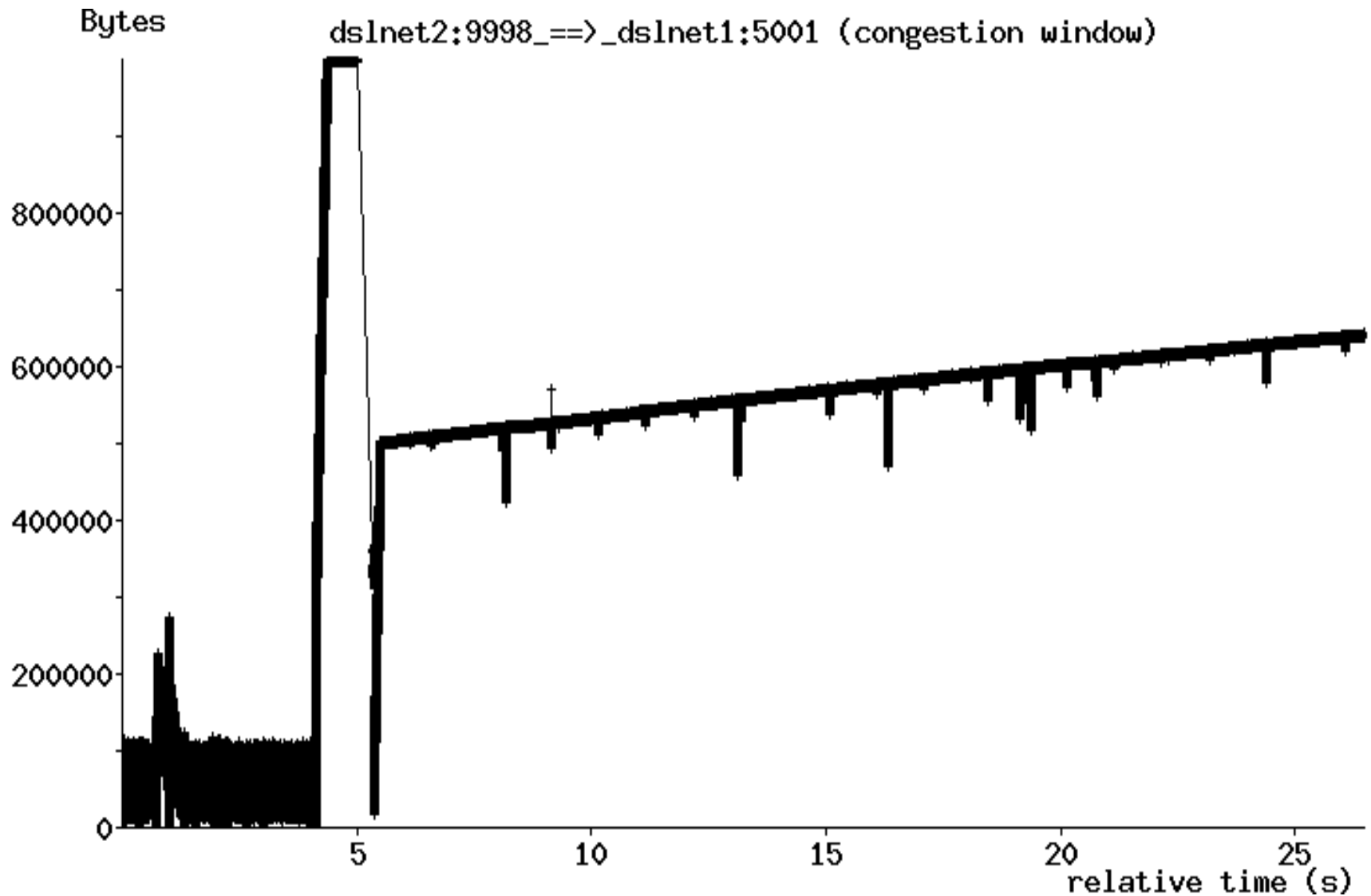


# WFQ: Default Buffer - Just BE



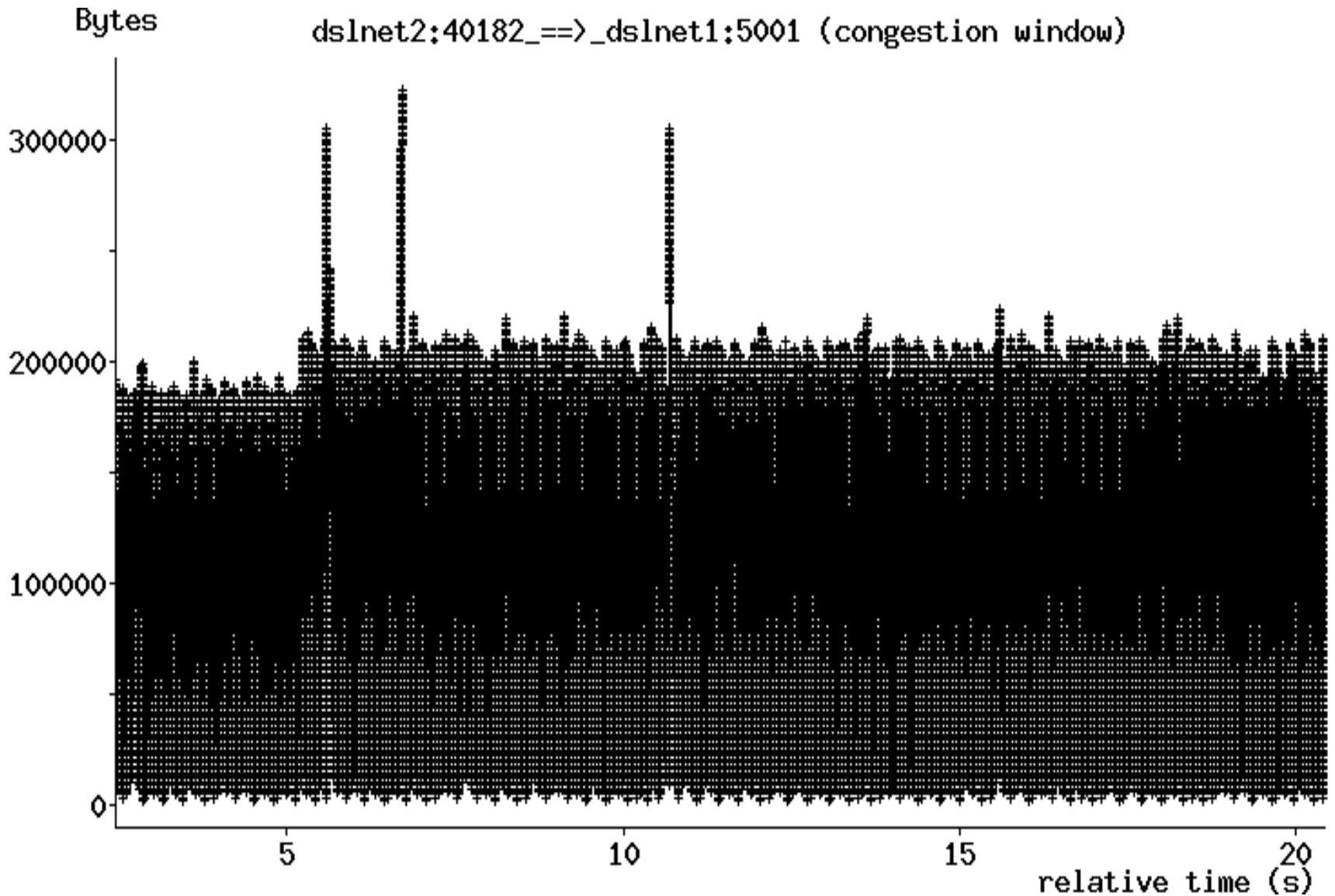


# WFQ: Modified Buffer - Just BE





# 10 MB/s EF TCP-Flow





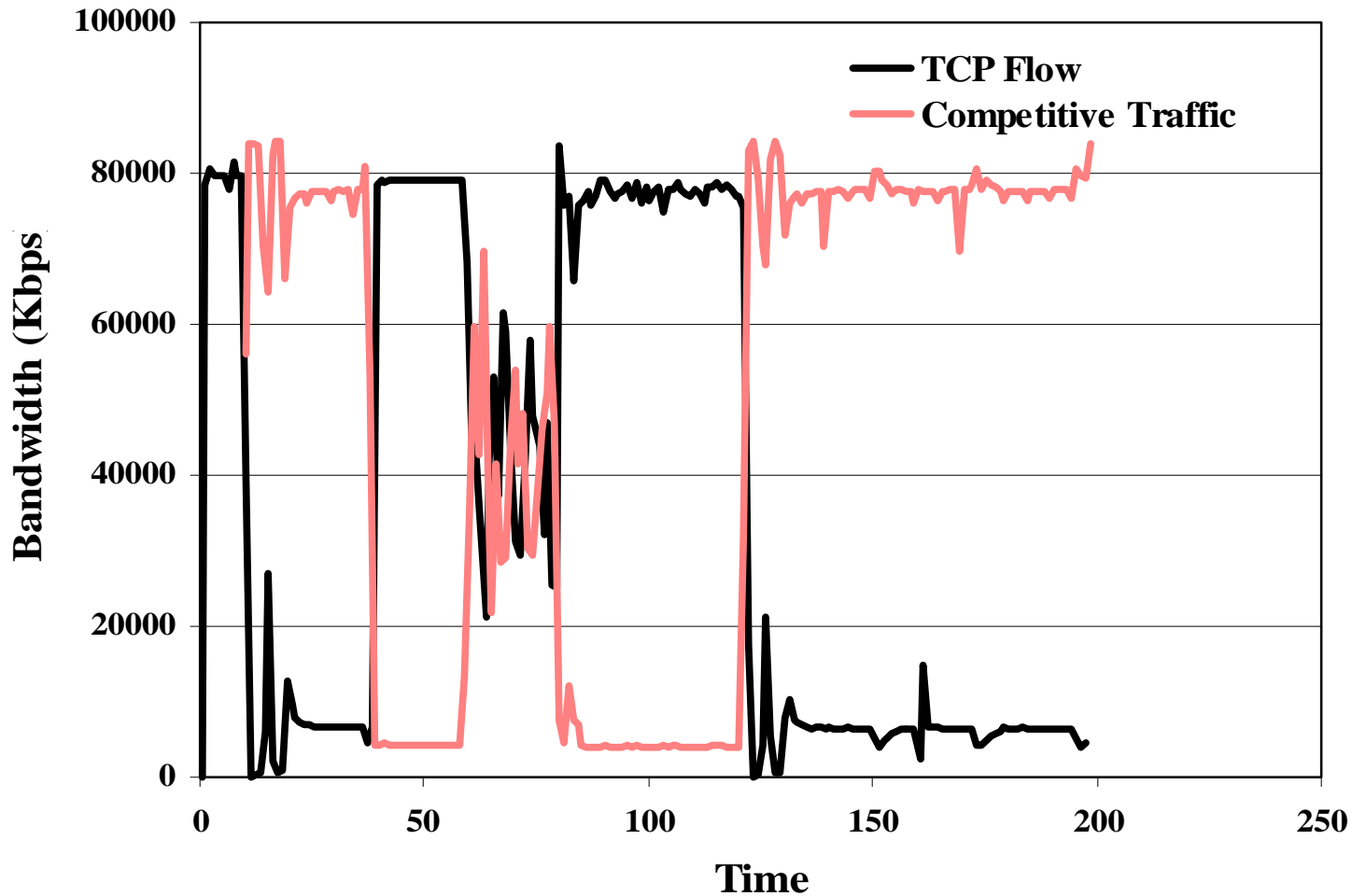
the globus project

[www.globus.org](http://www.globus.org)

# Advanced BB Features...

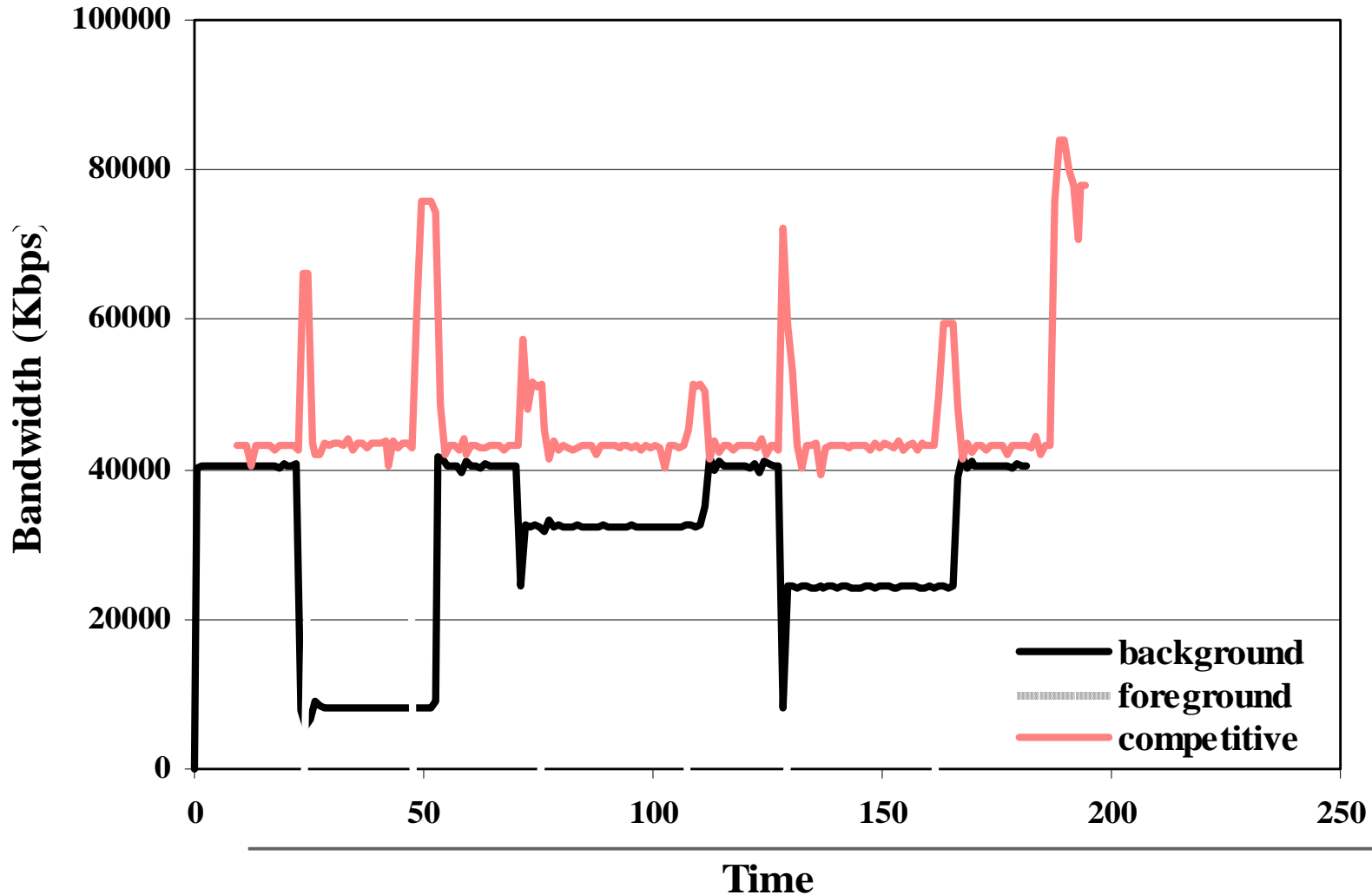


# Network + CPU Reservations





# Bulk Transfer with Deadline Support (LAN)







# Conclusions and Future Work

- A Bandwidth Broker supporting High-Performance TCP Flows should:
  - Allow bursts of one Window-Size
  - Support a Low-Latency Class
    - > Still set the EF DSCP
    - > Ingress router should use different Queuing Class (PQ)
    - > Configure Shaping on Ingress Router based on actual reservations!
  - Provide advanced Functionalities and Policies
    - > Feedback to the application
    - > Online Monitoring of Edge Devices
- Traffic Shaping is Work in Progress



# Questions...???

- Now or later... (sander/roy @ mcs.anl.gov)