



GARA: An Architecture for QoS

Alain Roy

alain@cs.uchicago.edu



A note on Globus

- Globus is software toolkit for building distributed applications.
- GARA is part of Globus.
- More on Globus at:

www.globus.org



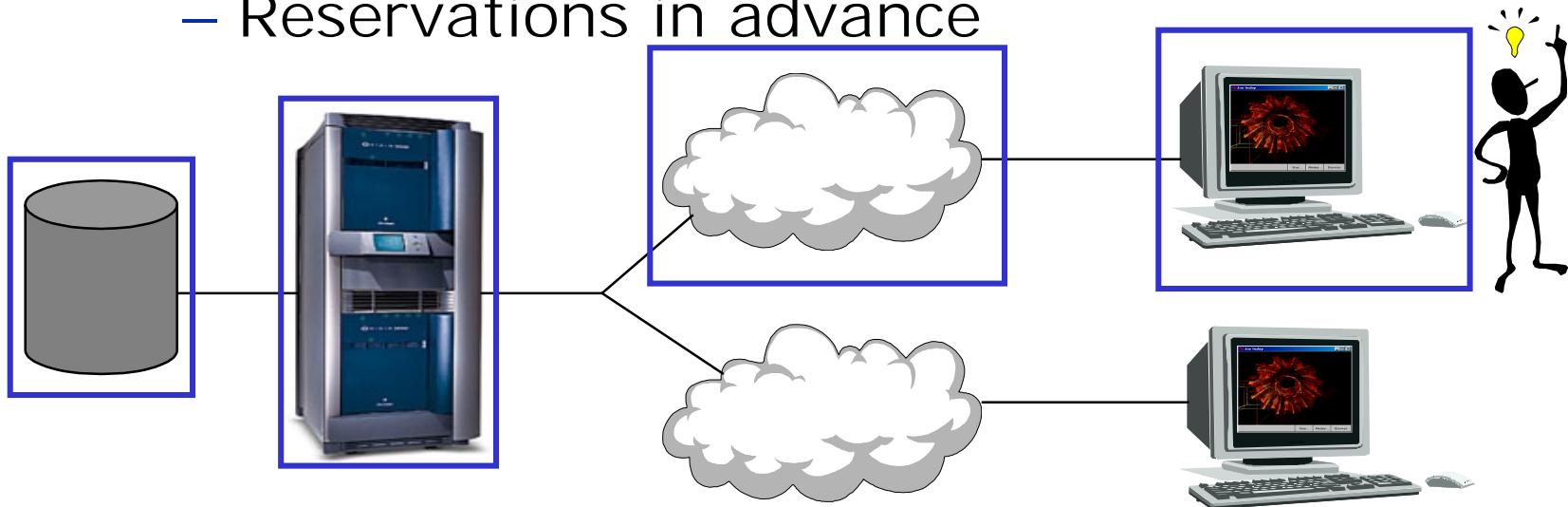
Goals

- Provide end-to-end Quality of Service to applications.
- Target: high-end applications
 - Audio/Video
 - Remote I/O
 - Scientific applications
 - Teleimmersion (virtual reality)
- Requirements:
 - TCP & UDP
 - Low & High Bandwidths—even 10s of MB/s



Example Application

- Scientific simulation and visualization
 - Different types of resources/QoS
 - Reservations in advance



Note: Disk, Computer, Network, and Workstation reservations!



A Note on Design Motivation

- We believe it is important to integrate different types of QoS.
- Our design choices were influenced by this desire.
 - Later, notice co-reservation and end-to-end network reservations.



Expanded Goals

- Providing end-to-end Quality of Service to applications requires:
 - Discovery and selection of resources.
 - Advance reservation of resources.
 - > Scarce resources are in demand
 - > Collaboration, demos, etc. require specific times
 - Convenient and useful for users
 - Allocation of resources.



Difficulties/Solutions

- Lack of support for advanced reservations
 - We can use existing advanced reservation mechanisms if available or supply our own.
- Heterogeneous resources
 - We provide uniform interfaces.
- Need to work with complex sets of resources
 - We use co-reservation “agents”.
- Wide variety of network flows
 - high and low bandwidth
 - **TCP** and **UDP**



Solution: GARA

Globus Architecture for Reservation and Allocation

- Four important big-picture contributions:
 - Advance reservations are first-class entities.
 - Uniform treatment of underlying resources.
 - > CPU, network, disk, graphic pipelines, etc...
 - Layered architecture enables generic co-reservation agents.
 - Monitoring to provide useful feedback to applications.



General QoS vs. Network QoS

- GARA tries to unify QoS systems.
- But not all the QoS systems are there!
- Therefore we are also working on the individual QoS systems
 - Especially networking QoS
- For this talk, I'll focus on networking



Reservations

- There is a generic “reservation”, which has several properties:
 - Start Time (“now” or future) and Duration
 - Resource type/Underlying resource identifier
 - Resource-specific (bandwidth, % CPU...)
- All reservations are treated uniformly:
 - Create/Modify (Given properties)
 - ➔ Returns “Reservation Handle”
 - Cancel
 - Monitor (Callbacks or Polling)



Kinds of network reservations

- Normal
 - User specifies specific bandwidth.
 - Most commonly used.
- Adaptive
 - Like normal, but with feedback about performance.
- Bulk Transfer
 - User is given all unused premium bandwidth
 - > shared proportionately for multiple bulk transfers.
 - When unused premium bandwidth changes, user is informed of amount available.



Our uses of network reservations

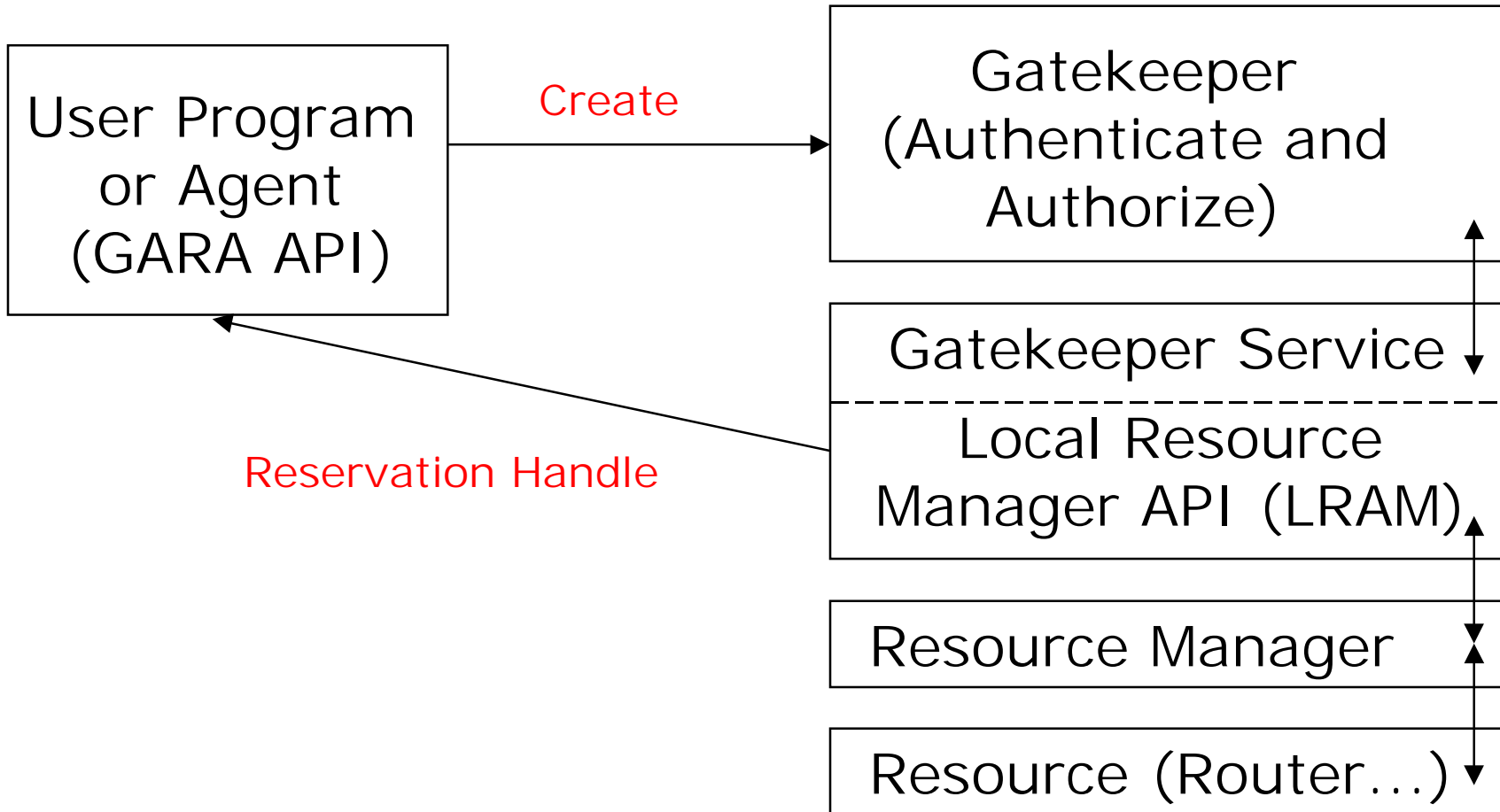
- Scientific Applications
 - Raw data for processing
 - Communication between distributed nodes
 - Processed data for visualization
- Remote I/O
- Virtual Reality and Collaboration
- Audio and Video

Note:

- Many of these use **lots** of bandwidth, even tens of MB/sec!
- TCP and UDP

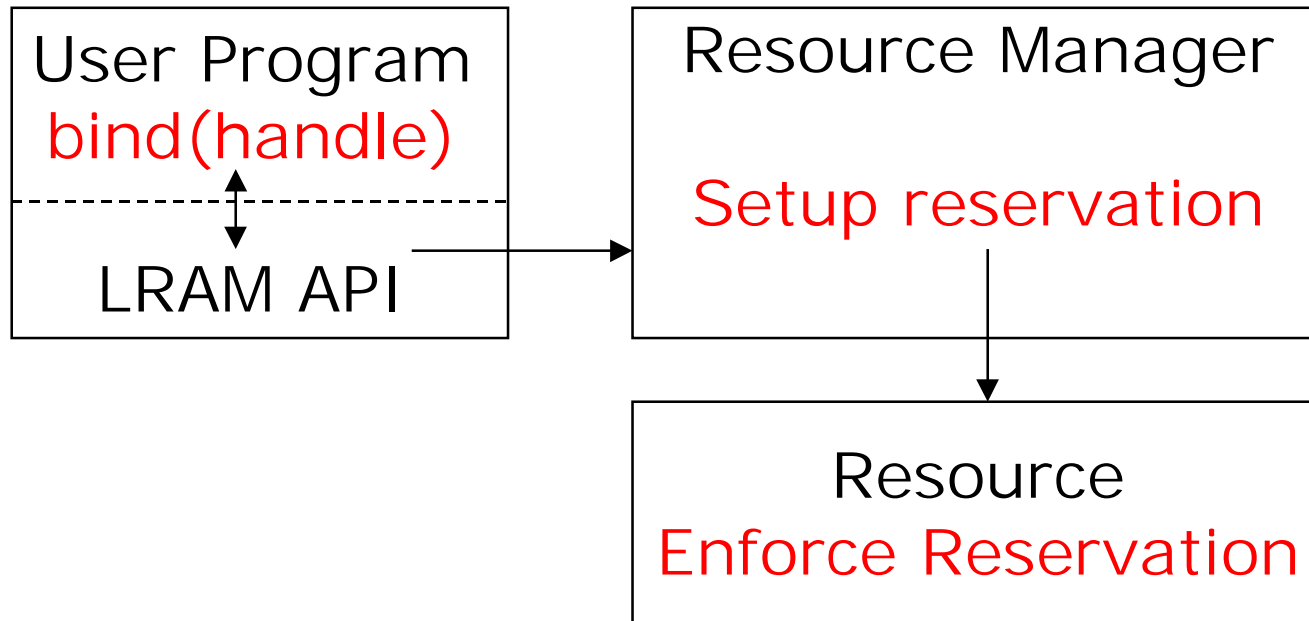


User→BB Communication Creating a Reservation





User→BB Communication Claiming a Reservation



Note: GARA supports third-party creation & claiming
(useful for legacy applications)



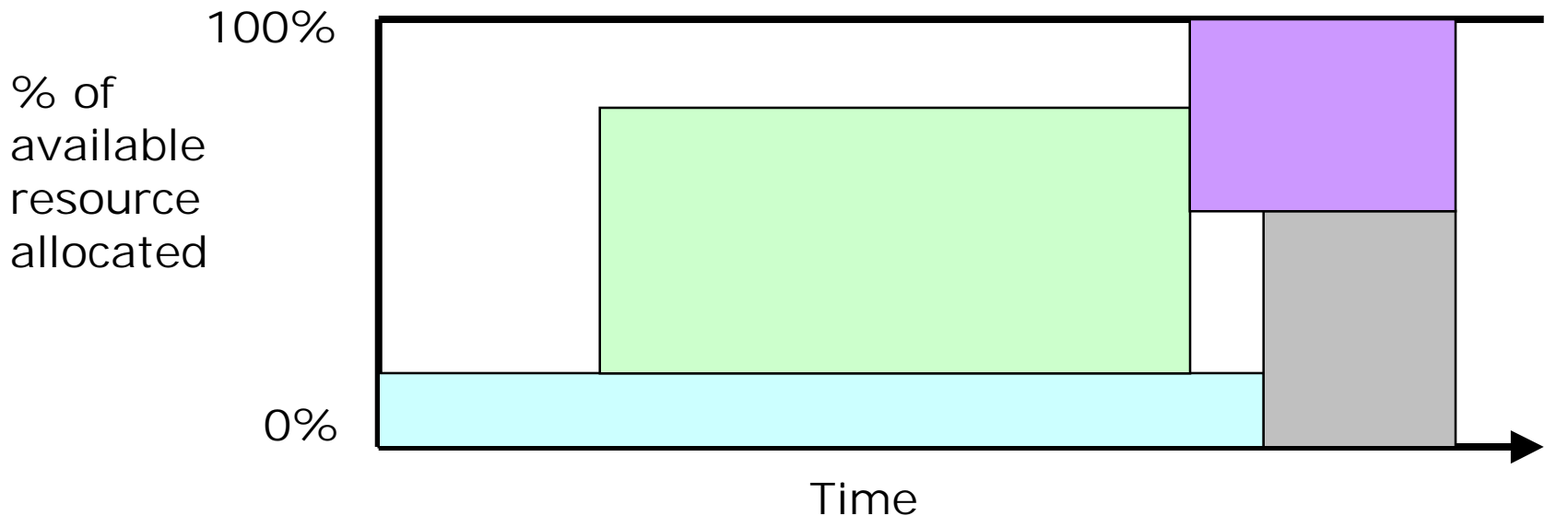
Example: Network reservation

- In advance, make a reservation:
 - ➔ Specify reservation:
& (reservation-type=network)
(start-time="now" + 1800) (duration=3600)
(endpoint-a=128.135.11.4) (endpoint-b=128.135.11.1)
(bandwidth=1000) (protocol=tcp)
 - ← Handle: Opaque string H_1
- At run-time, claim the reservation:
 - Bind(H_1 , port_A, port_B)



The Resource Manager

- Performs admission control/controls resource.
- We assume exclusive access to resource through the resource manager.
- We aren't tied to this particular resource manager.
 - We could use anyone else's bandwidth broker





The Network Resource Manager (a.k.a. Bandwidth Broker)

- Has knowledge of network topology.
 - Controls a single domain.
 - Currently static, dynamic (OSPF) in future.
- Mostly independent of type of router.
 - Router knowledge localized in scripts.
- Monitors reservations
 - Queries routers for statistics on conforming and dropped packets.

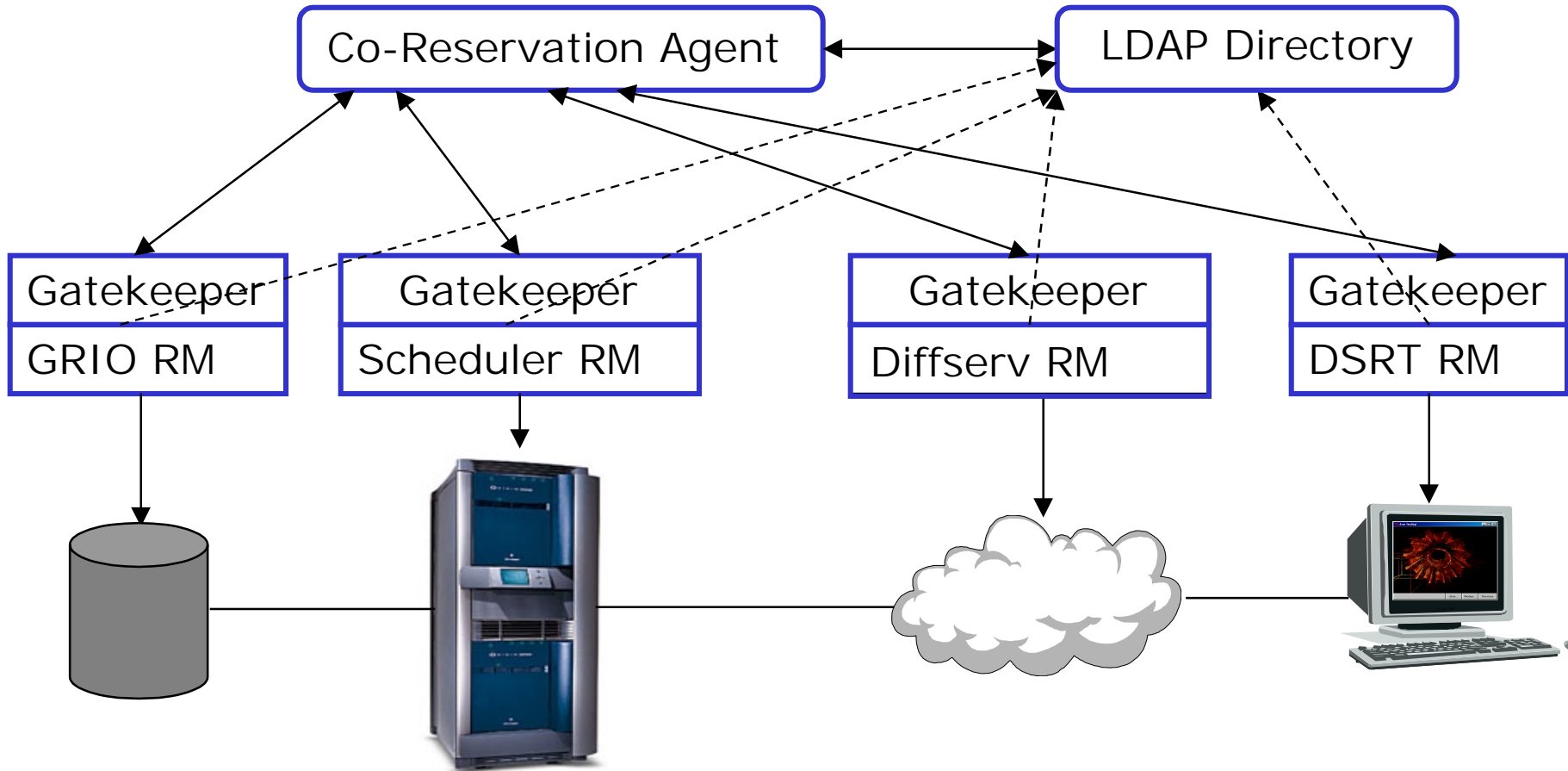


Co-Reservation Agents

- When multiple resources are needed, an agent:
 - Discovers applicable resources via a directory.
 - Discovers time when they can all be used.
 - Reserves resources.
 - Informs user when resources are ready.
- The uniform interface enables these agents to be created easily.



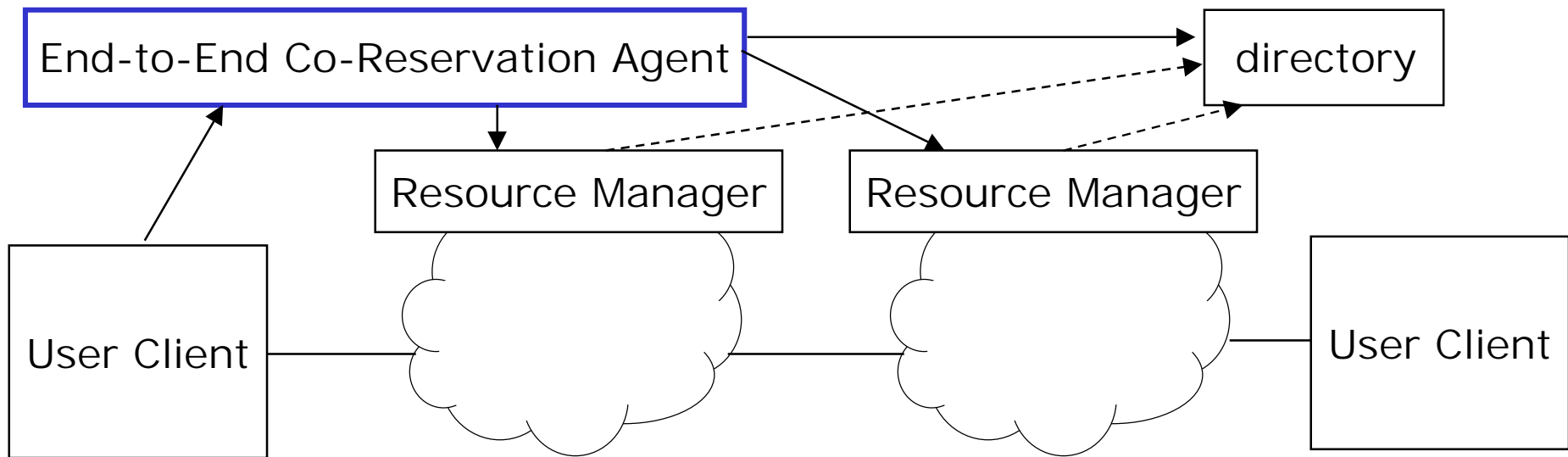
The Big Picture





End-to-End Network Reservations

- Just a special case of co-reservation
 - Make a reservation in each domain.
 - Agent coordinates reservation





More on End-to-End Network Reservations

- Right now, each reservation is made independently, in parallel.
 - No way to ensure reservation is made at each point.
- In the near future: Chained reservations.
 - Make reservation at first resource manager.
 - Answer: “Yes, if you get a signed confirmation from the next domain that you got a reservation.”
 - Repeat until last domain.
 - Present signed confirmations, in backward chain.

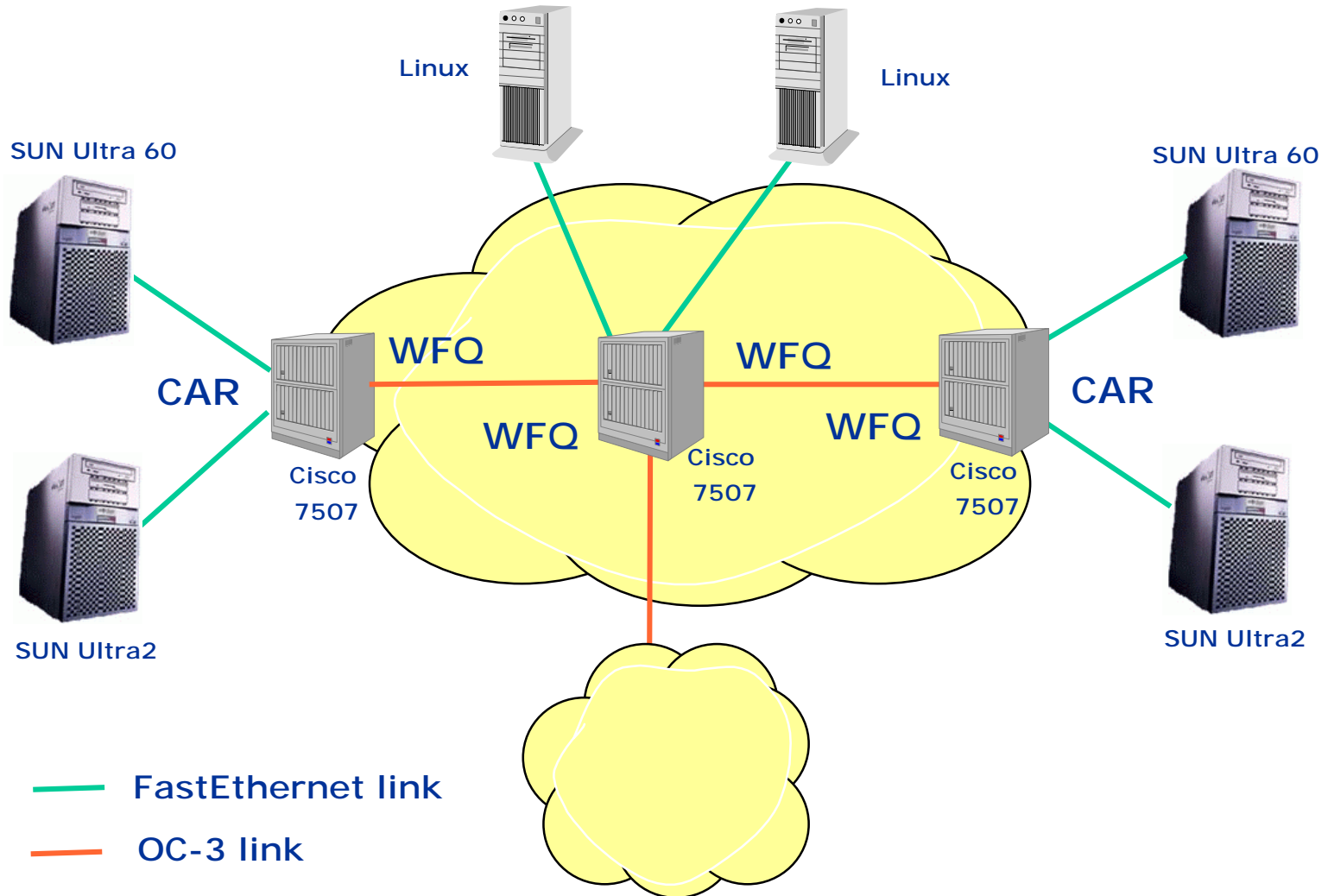


Network QoS implementation

- We use differentiated services.
- Expedited forwarding to implement “premium service”.
 - Emulated with Cisco 7507 routers
 - Edge routers controlled with resource manager, as described above.
 - Currently using expect + telnet for configuration. SNMP or COPS in the future.

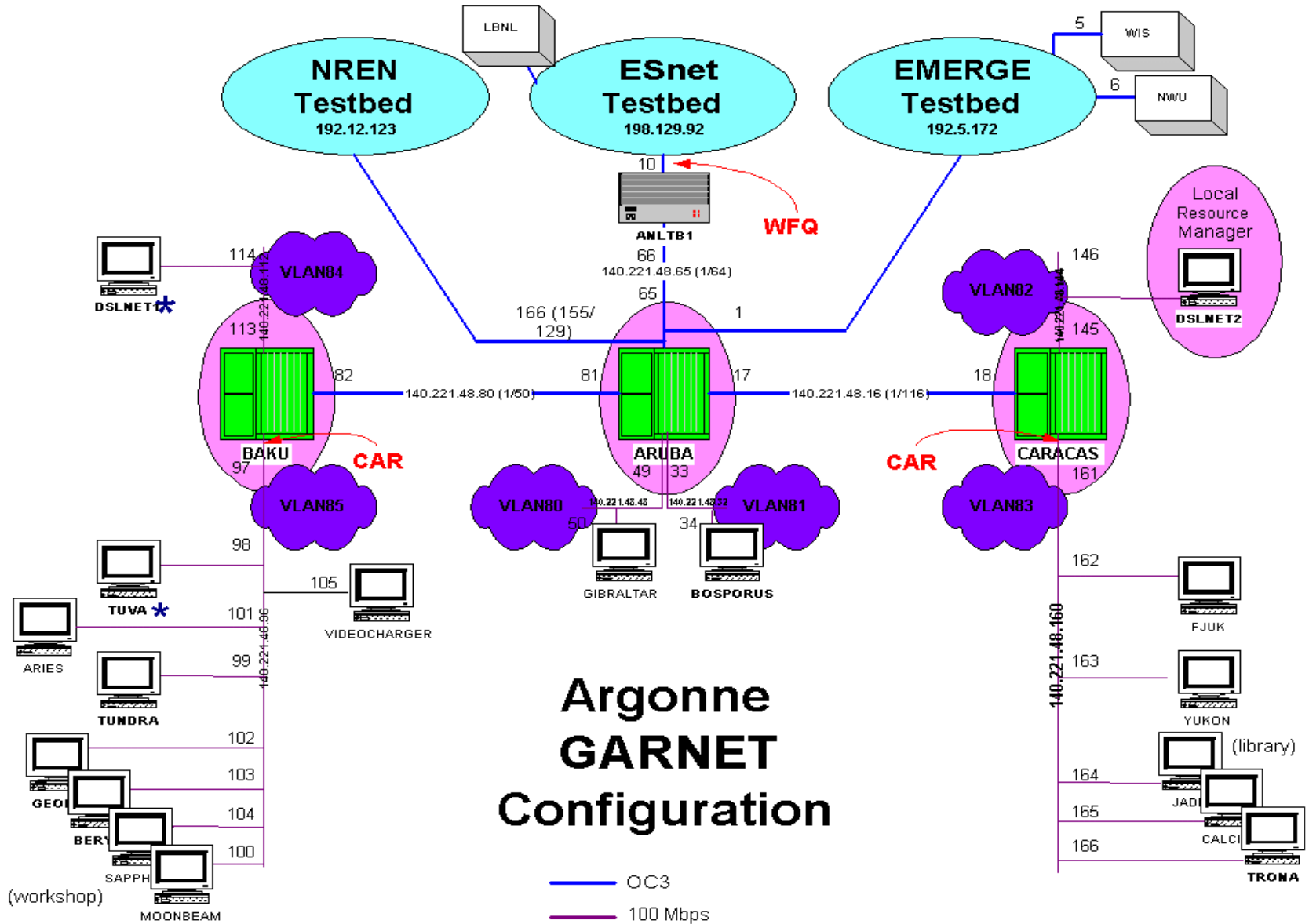


GARnet: Globus Advance Reservation Network Testbed





GARnet: The big picture



Argonne GARNET Configuration

— OC3

— 100 Mbps



QoS Mechanisms (Cisco)

- Modular QoS command-line interface (MQC)
 - Per flow based at first-hop routers
 - > Mark packets with precedence if they do not exceed rate limit.
 - > Drop packets beyond the limit (or transmit them via best effort).
 - Per Precedence between domains
 - > Statically configured—Corresponds to a static SLS.
 - > Resource manager will configure the limits on demand in future releases.
 - Allows short term bursts
 - > TCP is bursty!



QoS Mechanisms (Cisco)

- Weighted Fair Queuing (WFQ)
 - Used on interior interfaces
 - Configured “once” in a domain.
 - Divides available bandwidth into two classes:
 - > Best effort N%
 - > Premium 100-N%
 - > N is usually very small ~ 1%



Some Interesting Results...

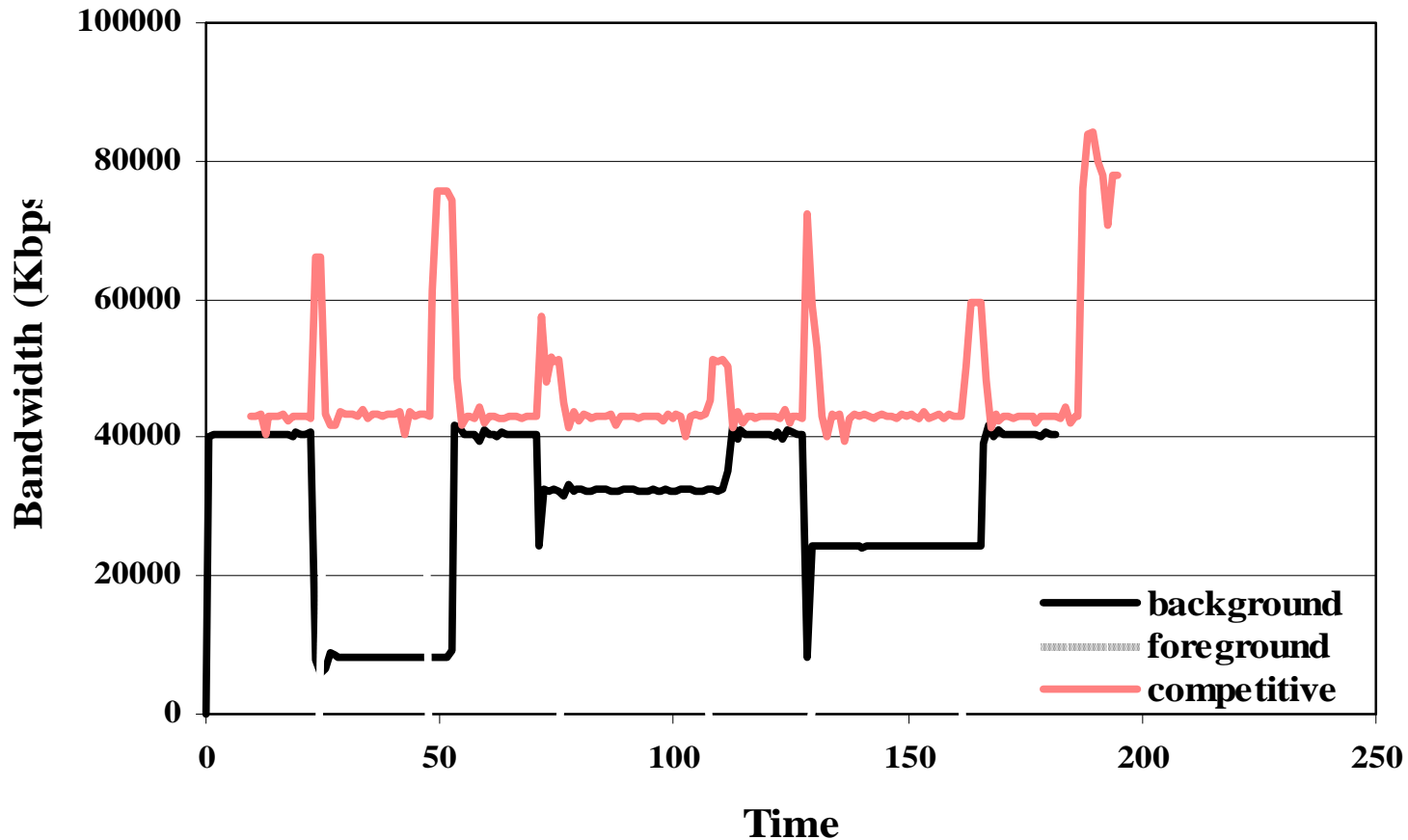


Bulk Transfer

- Recall: bulk transfers get the unused premium bandwidth
- We did experiments in both the local area and wide area (Argonne→LBNL)

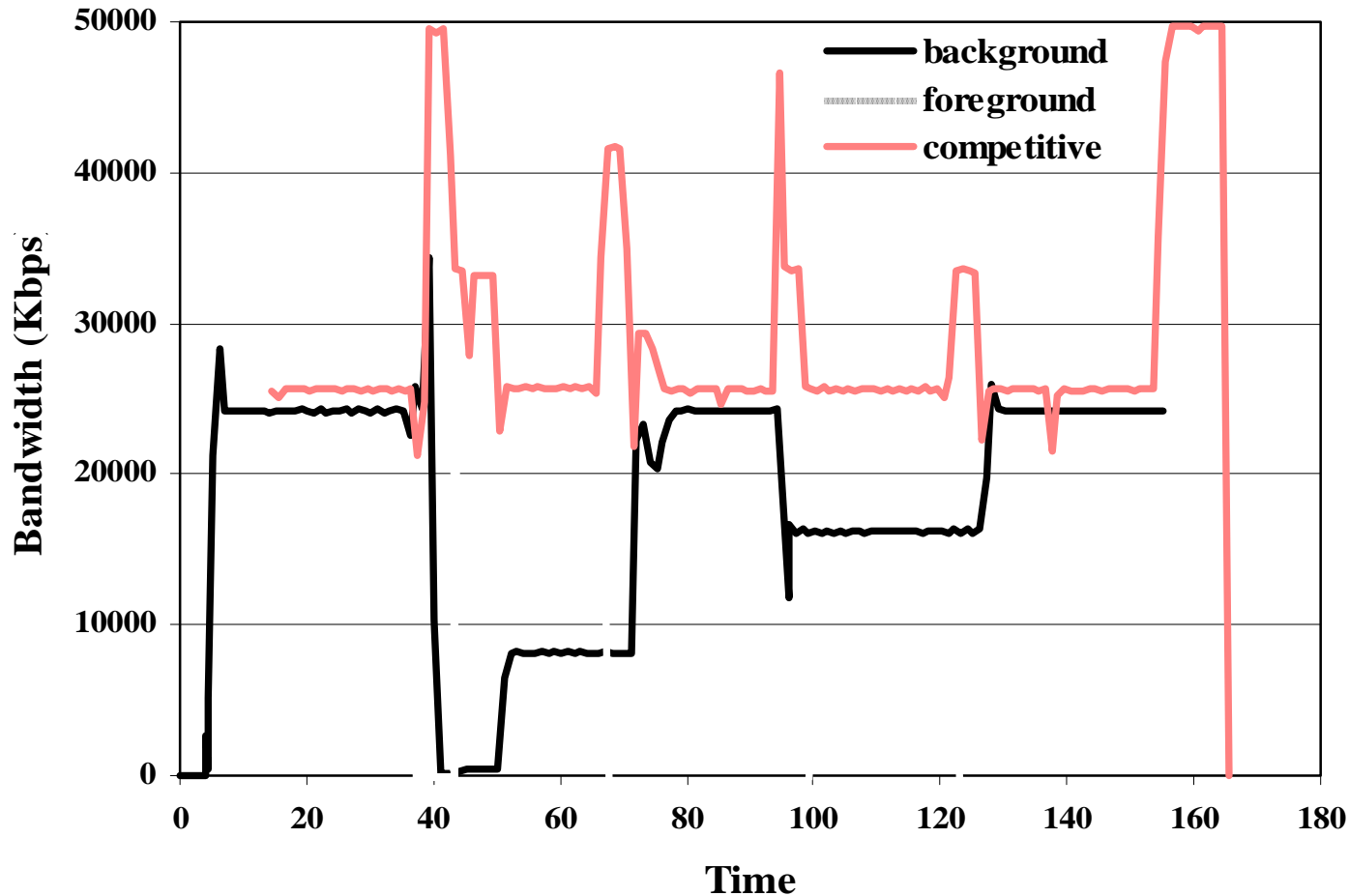


Example of bulk transfer (LAN)





Example of bulk transfer (WAN)



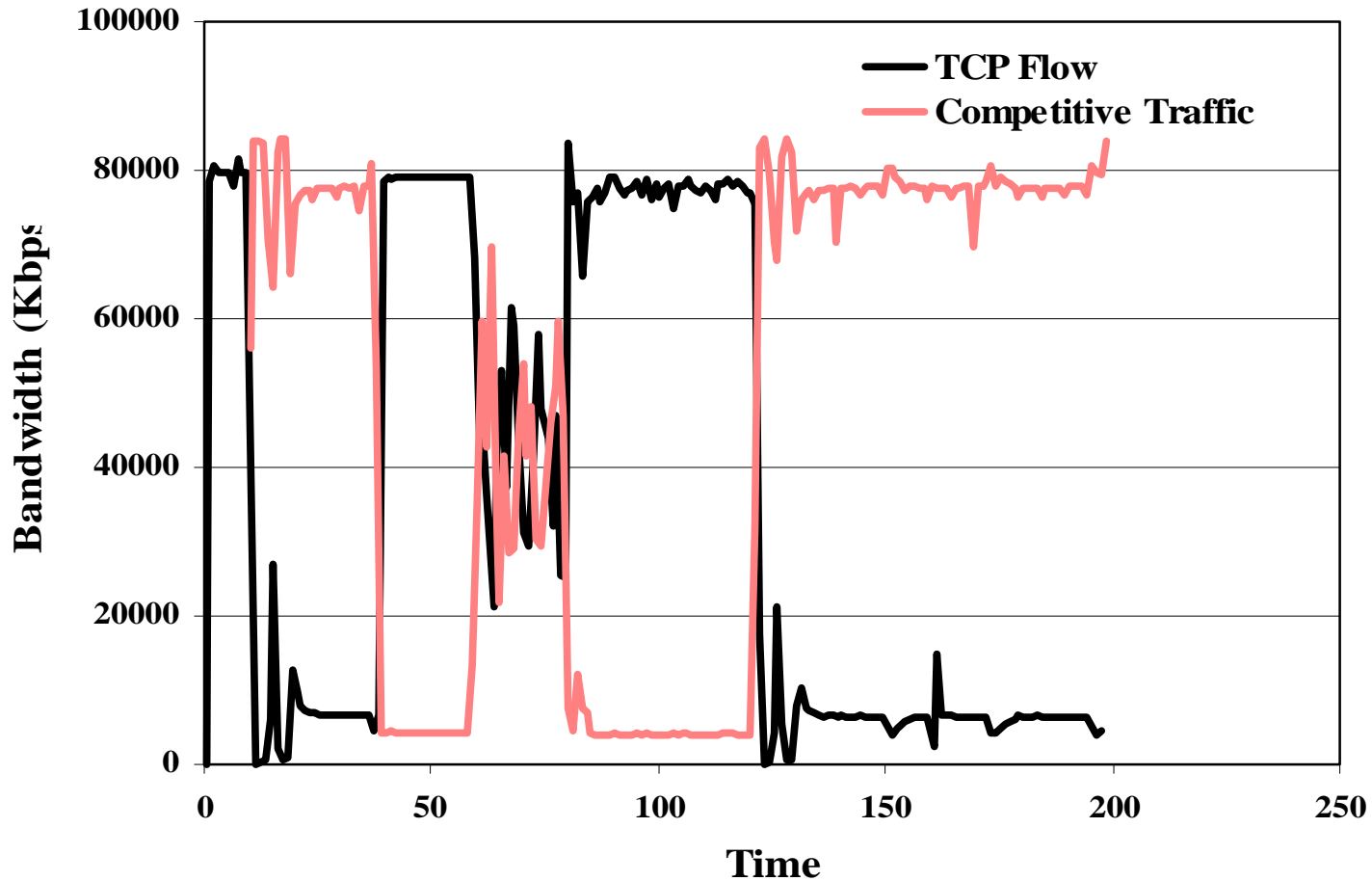


End-to-End QoS

- End-to-End QoS isn't just network QoS.
- We did experiments with a combination of:
 - Network QoS
 - CPU QoS: the Dynamic Soft Real-Time Scheduler



Network + CPU Reservations





Monitoring and Feedback

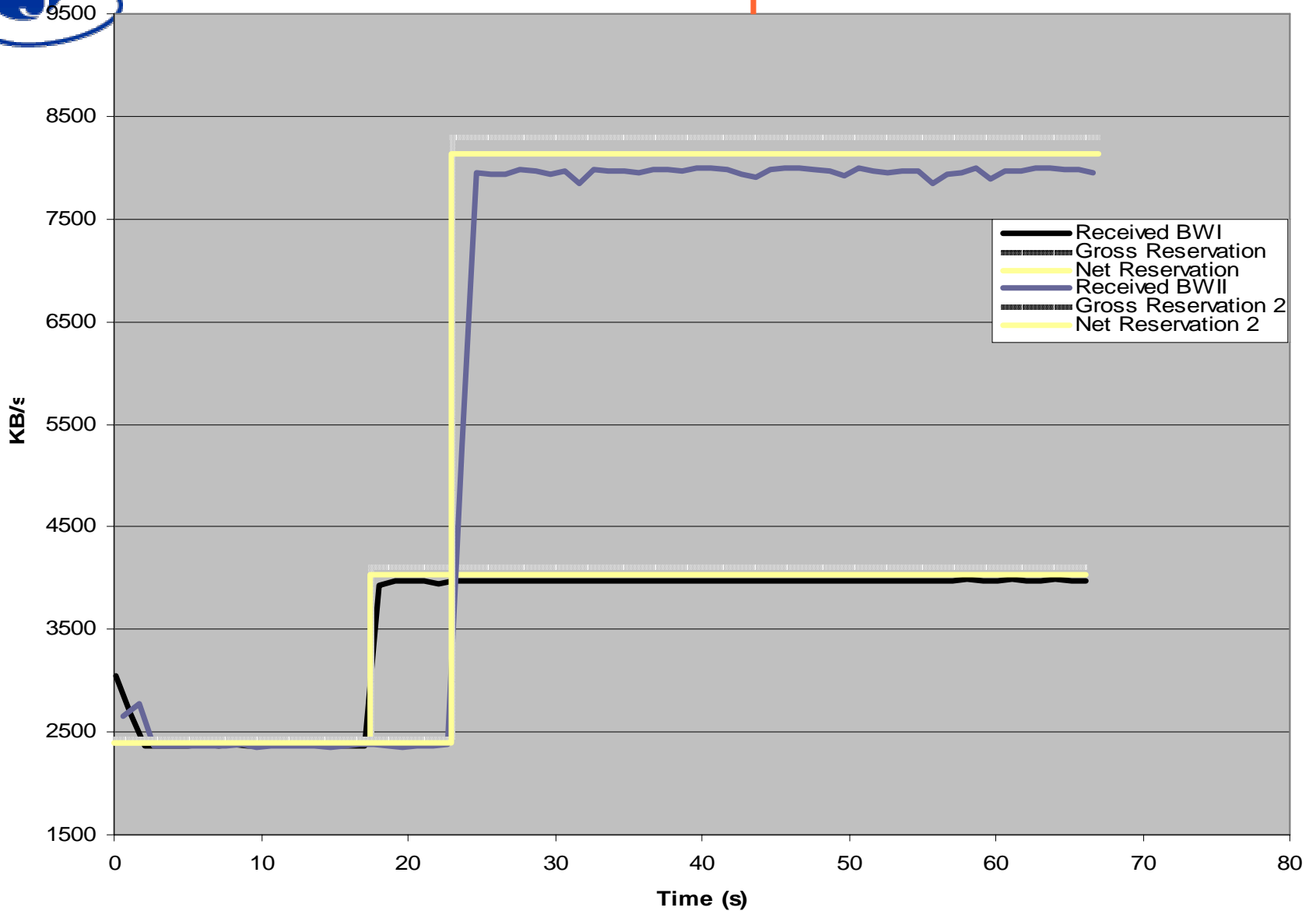


Monitoring and Feedback

- Resource Manager provides feedback.
 - Monitors and conforming and dropped packets on routers.
 - Informs application via “callback” mechanism.
- Application can adapt its reservation.
 - Easy calculation for UDP
 - Rather difficult for TCP.



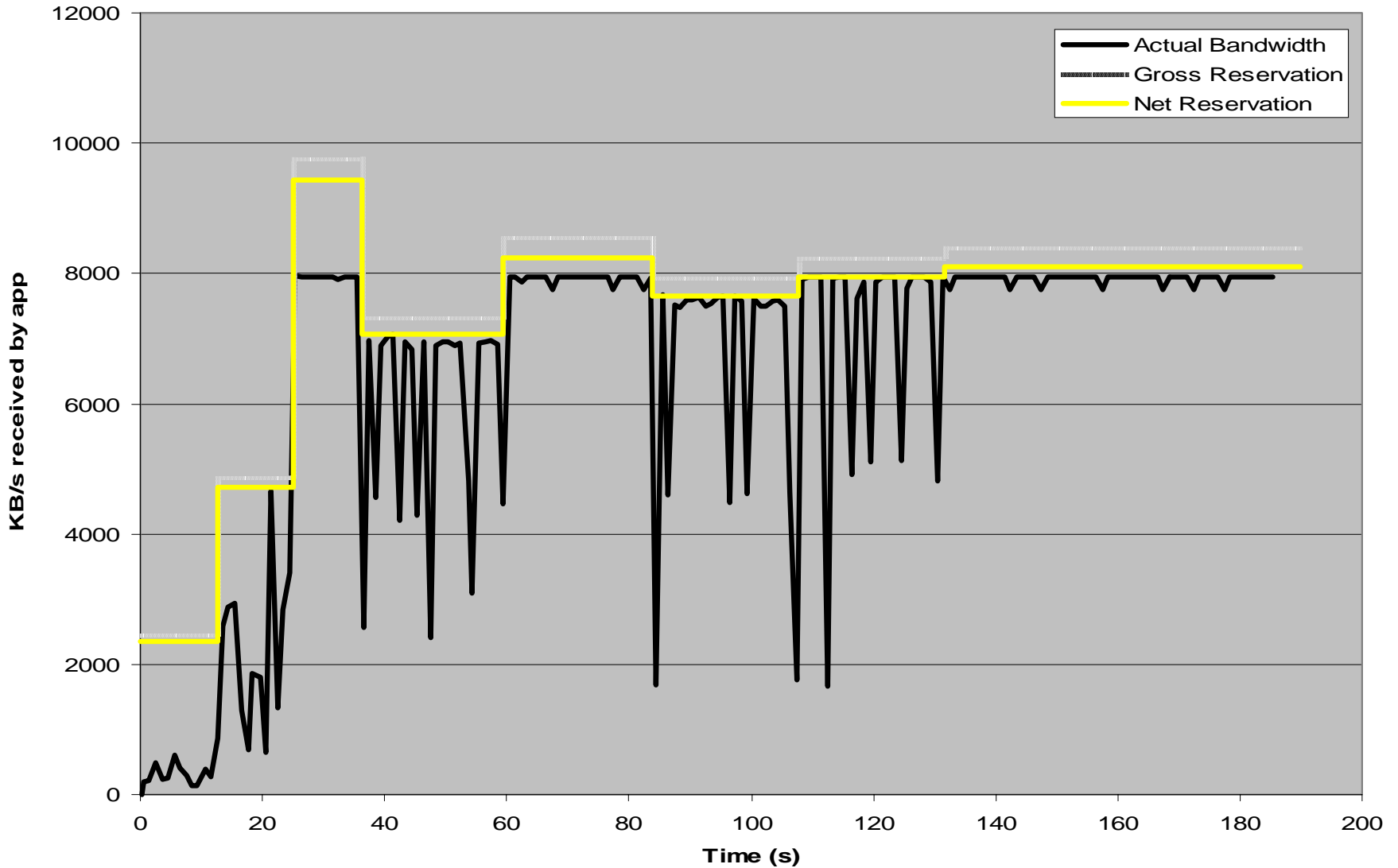
UDP Adaptation





TCP Adaptation

Receiver Bound Bandwidth





Current Status

- A *working* prototype of GARA exists
 - Differentiated Services
 - Real-Time CPU Scheduling (DSRT)
 - DPSS Disk Access
- An *working* prototype of the end-to-end network reservation agent exists.
- Validation and experiments are being done concurrently on a testbed.
- Work with early adopters has started.



Future Work

- Improved handling of inter-domain reservations.
- Working with policies
 - Who has access when, costs, accounting
 - Priorities/Preemption.
- Supporting more resource types
 - Job Schedulers, disk, graphic pipelines...
- Experimentation with more real applications
- High-level agents to simplify usage.



Questions?

- Feel free to email us:
 - Alain Roy: roy@mcs.anl.gov
 - Volker Sander: sander@mcs.anl.gov
- Check our web site:
 - <http://www.mcs.anl.gov/qos/>



Extra Slides

(just in case)



The GARA API

- `globus_gara_reservation_create()`
 - ➔ Gatekeeper contact
 - ➔ RSL reservation specification
 - ← Reservation Handle or Error
- `globus_gara_reservation_modify()`
 - ➔ Old Reservation Handle
 - ➔ RSL reservation specification
 - ← New Reservation Handle or Error



The GARA API (continued)

- `globus_gara_reservation_cancel()`
 - ➔ Reservation Handle
 - ← Error
- `globus_gara_reservation_status()`
 - ➔ Reservation Handle
 - ← Status/Error
- `globus_gara_reservation_callback_register()`
 - ➔ Reservation Handle
 - ➔ Callback Function/User Parameter
 - ← Error

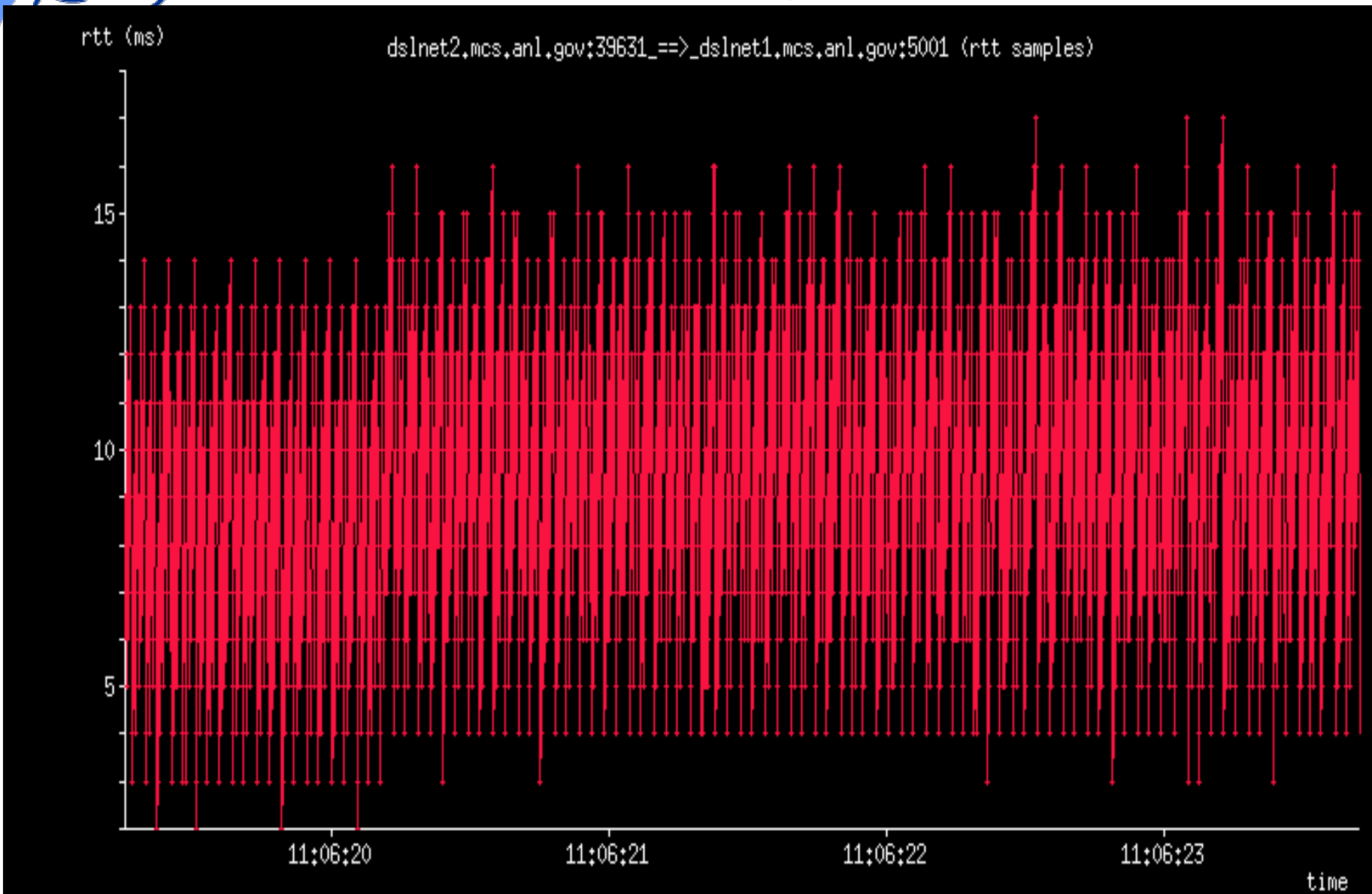


The End-to-End Agent API

- For making network reservations
 - Deals with co-reservation along path.
- Very similar to GARA API:
 - Ex: `globus_gara_end2end_create()`
- Finds resource managers.
- Makes multiple reservations, but provides user with single reservation handle.
- The most common API for network reservations.



Using WFQ



Using WFQ

