TCP Performance over a 2.5 Gbit/s Transatlantic Circuit

HNF-Europe Workshop 2002, ECMWF, UK
11 October 2002

J.P. Martin-Flatin, CERN
S. Ravot, Caltech

The DataTAG Project

http://www.datatag.org/

Project Partners

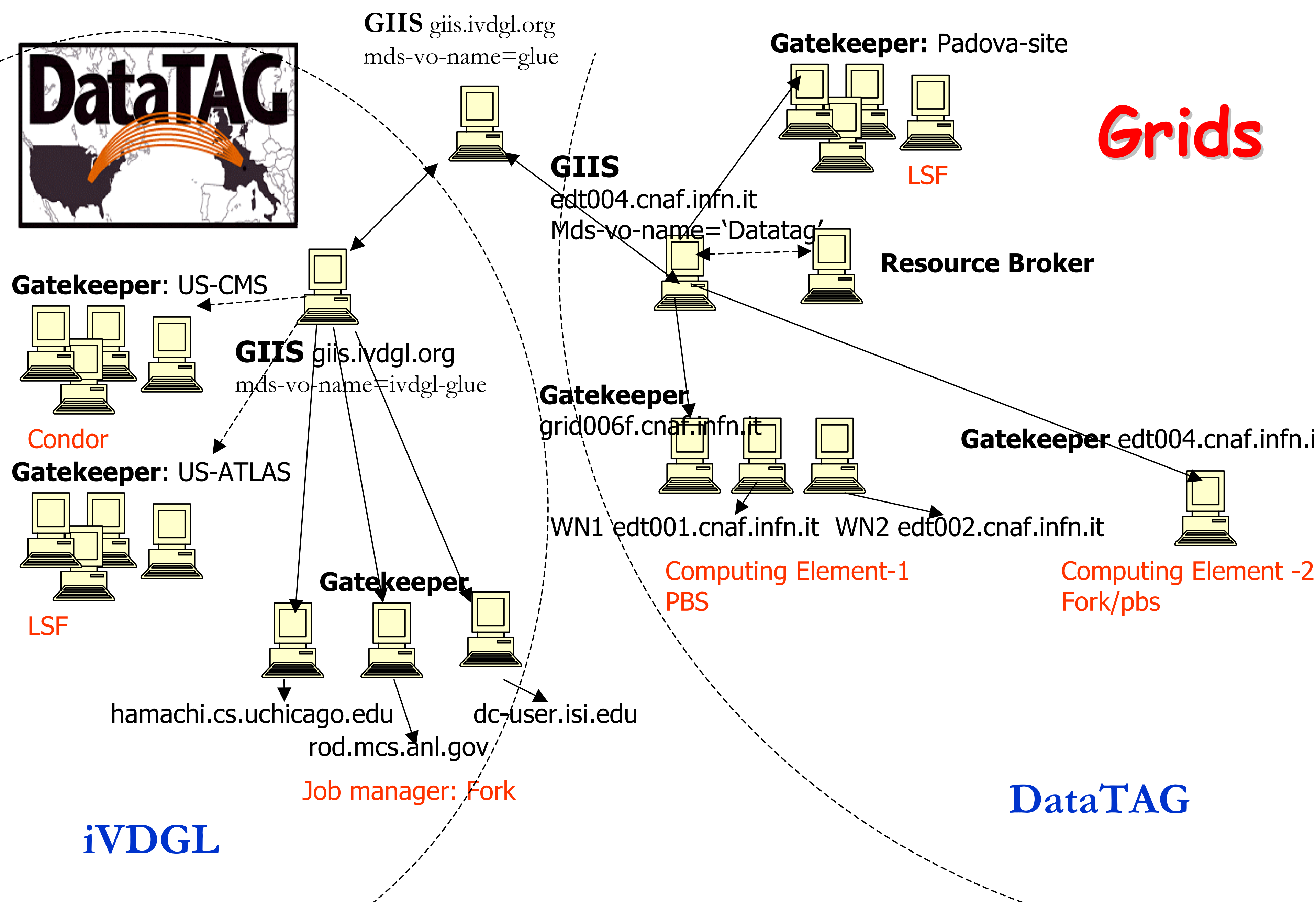
- EU-funded partners: CERN (CH), INFN (IT), INRIA (FR), PPARC (UK) and University of Amsterdam (NL)
- U.S.-funded partners: Caltech, UIC, UMich, Northwestern University, StarLight
- Collaborations: GEANT, Internet2, Abilene, Canarie, SLAC, ANL, FNAL, LBNL, etc.
- Project coordinator: CERN

About DataTAG

- Budget: EUR 3.98M
- Funded manpower: 15 FTE/year
  - 21 people recruited
- 26 people externally funded
- Start date: January 1, 2002
- Duration: 2 years

Three Objectives

- Build a testbed to experiment with massive file transfers across the Atlantic
- High-performance protocols for gigabit networks underlying data-intensive Grids
- Interoperability between several major Grid projects in Europe and USA


Grids
DataTAG
GIIS giis.ivdgl.org
mds-vo-name=glue
Gatekeeper: Padova-site
LSF
GIIS
edt004.cnaf.infn.it
Mds-vo-name='Datatag'
Resource Broker
Gatekeeper: US-CMS
GIIS giis.ivdgl.org
mds-vo-name=ivdgl-glue
Condor
Gatekeeper: US-ATLAS
LSF
Gatekeeper
grid006f.cnaf.infn.it
Gatekeeper edt004.cnaf.infn.i
WN1 edt001.cnaf.infn.it
WN2 edt002.cnaf.infn.it
Computing Element-1
PBS
Computing Element -2
Fork/pbs
Gatekeeper
hamachi.cs.uchicago.edu
dc-user.isi.edu
rod.mcs.anl.gov
Job manager: Fork
DataTAG
iVDGL

Testbed

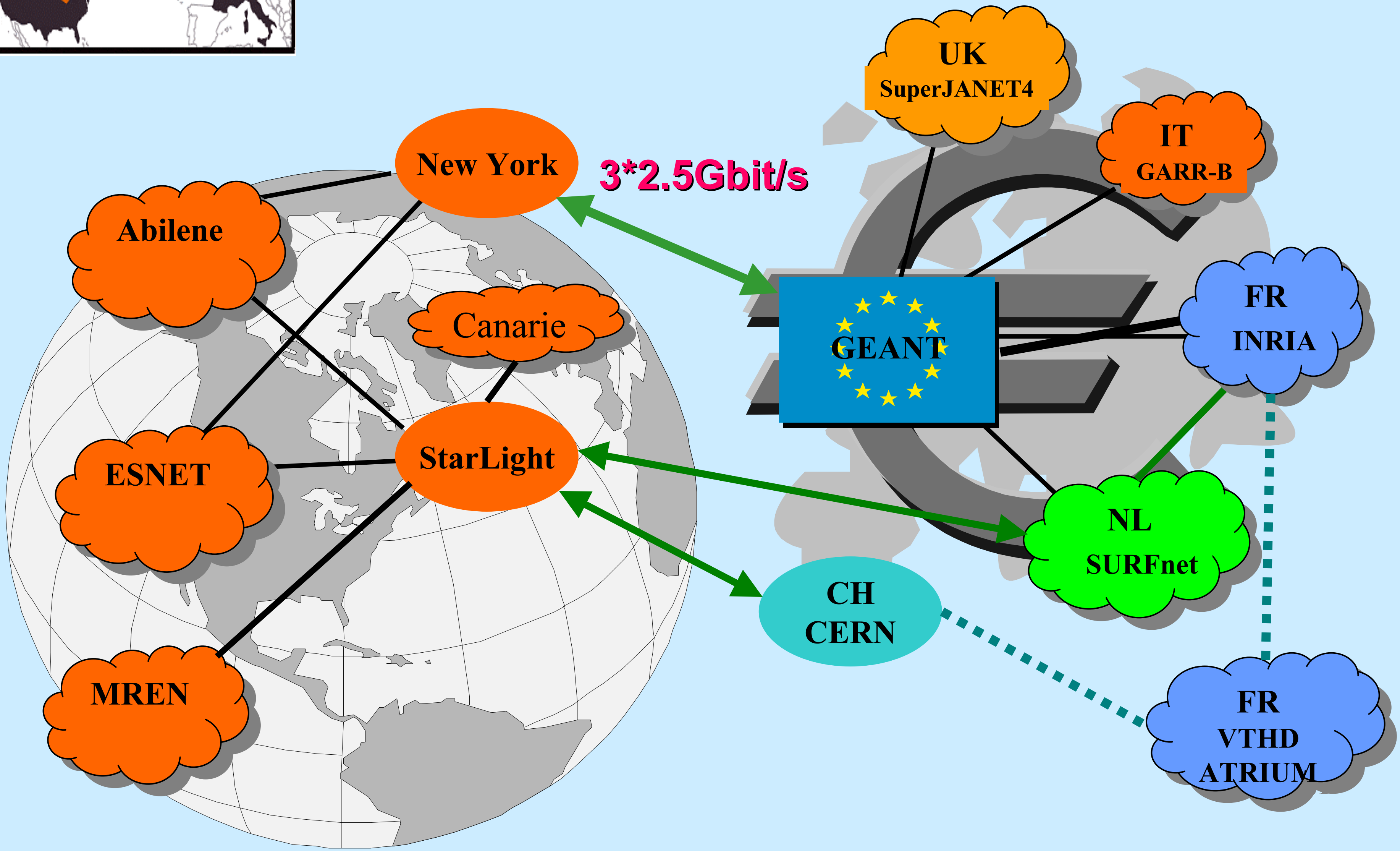
Objectives

- Provisioning of 2.5 Gbit/s transatlantic circuit between CERN (Geneva) and StarLight (Chicago)
- Dedicated to research (no production traffic)
- Multi-vendor testbed with layer-2 and layer-3 capabilities:
  - Cisco
  - Juniper
  - Alcatel
  - Extreme Networks
- Testbed open to other Grid projects

2.5 Gbit/s Transatlantic Circuit

- Operational since 20 August 2002 (T-Systems)
- Circuit initially connected to Cisco 7606 routers (layer 3)
- High-end PC servers at CERN and StarLight:
  - 6x SuperMicro 2x2.4 GHz
  - SysKonnect SK-9843 GbE cards (2 per PC)
  - can saturate the circuit with TCP traffic
  - ready for upgrade to 10 Gbit/s
- Deployment of layer-2 equipment under way
- Testbed fully deployed by 31 October 2002

R&D Connectivity Between Europe & USA



Network Research

Network Research Activities

- Enhance TCP performance for massive file transfers
- Monitoring
- QoS
  - LBE (Scavenger)
- Bandwidth reservation
  - AAA-based bandwidth on demand
  - lightpath managed as a Grid resource

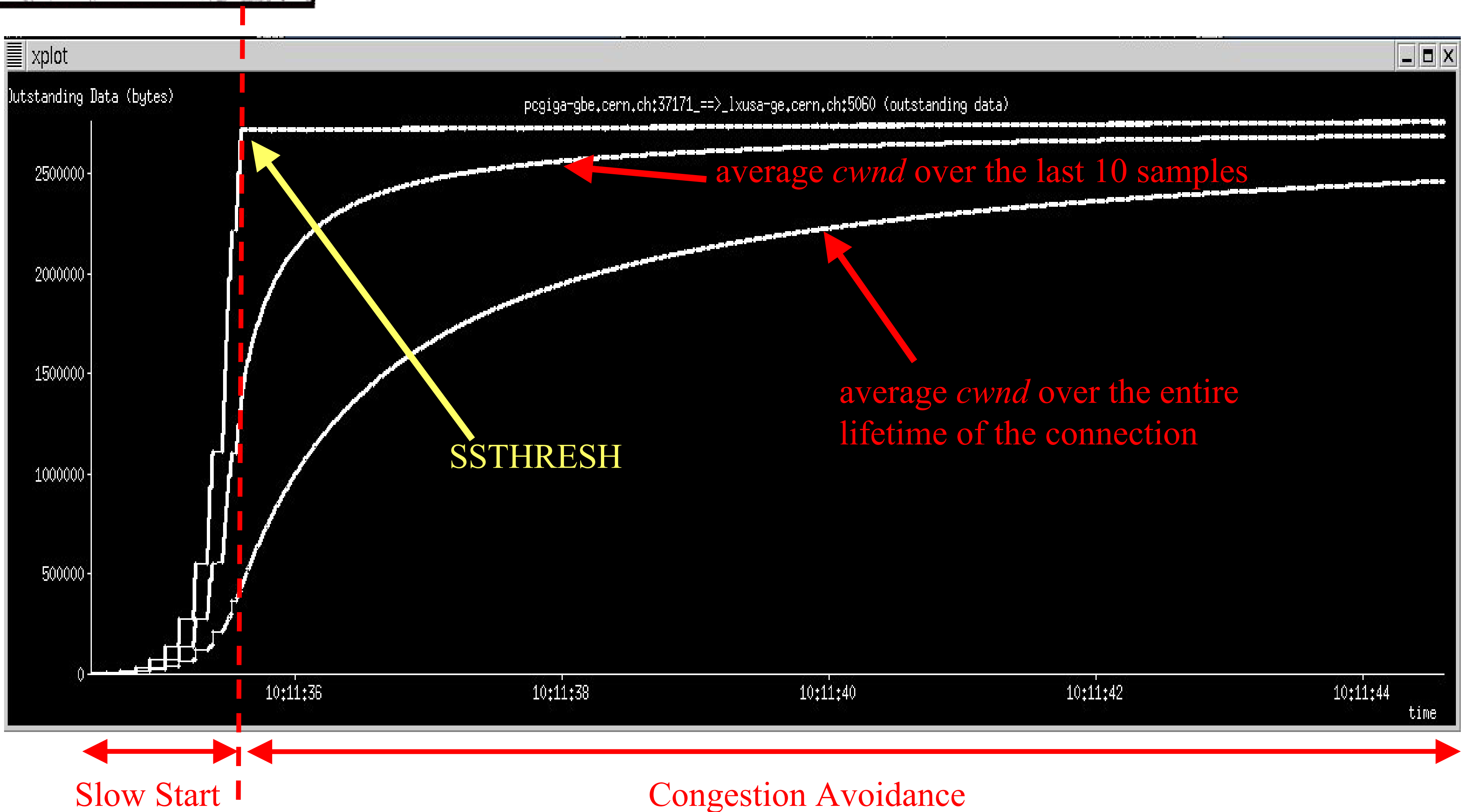
TCP Performance Issues: The Big Picture

- TCP's current congestion control (AIMD) algorithms are not suited to gigabit networks
- Line errors are interpreted as congestion
- Delayed ACKs + large *cwnd* + large RTT = problem

AIMD Algorithms

- Van Jacobson, SIGCOMM 1988
- Additive Increase:
  - a TCP connection increases slowly its bandwidth use in the absence of loss
    - forever, unless we run out of send/receive buffers
    - TCP is greedy: no attempt to reach a stationary state
  - Slow start: increase after each ACK
  - Congestion avoidance: increase once per RTT
- Multiplicative Decrease:
  - a TCP connection reduces its bandwidth use by half after a loss is detected

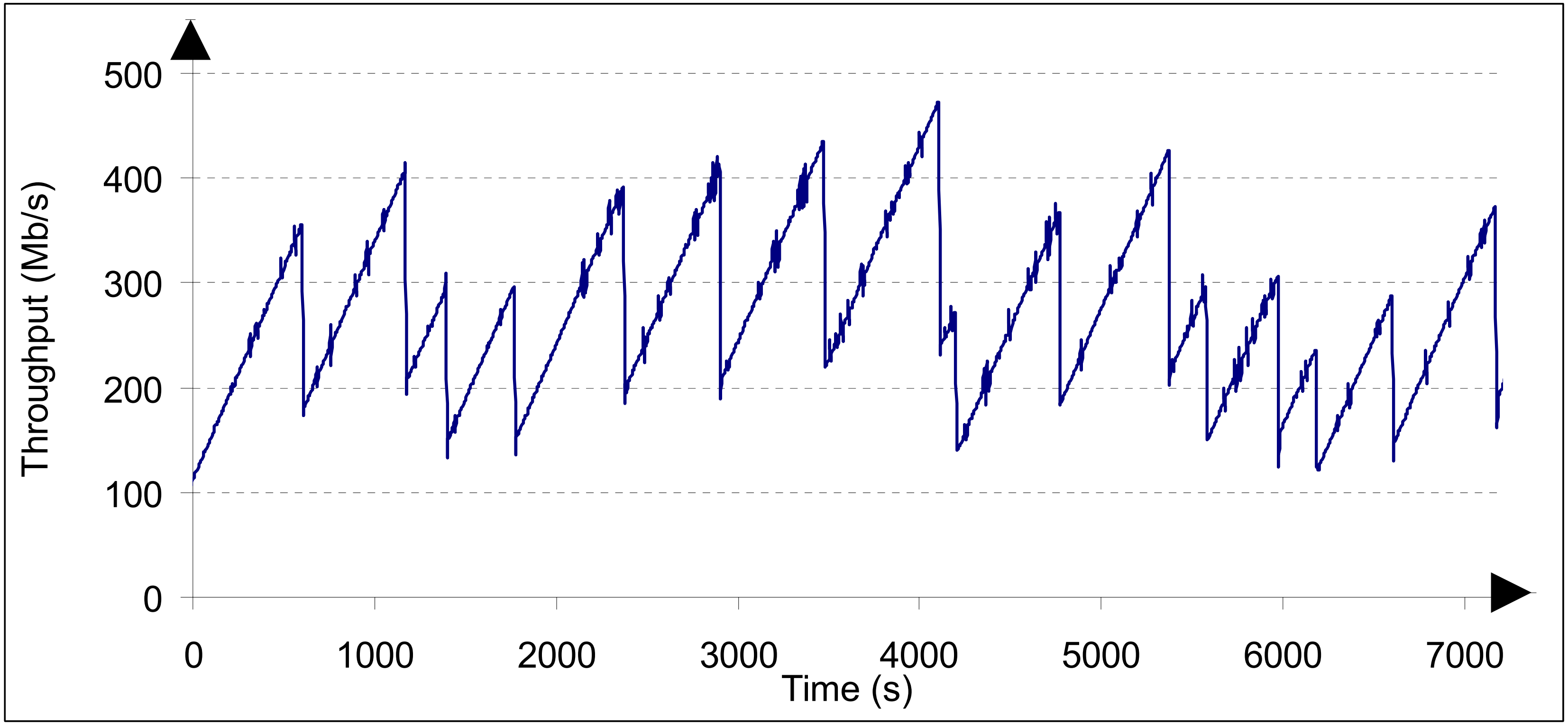
Congestion Window (*cwnd*)



tcptrace

Disastrous Effect of Packet Loss on TCP in WANs (1/2)

TCP throughput as a function of time **MSS=1460**



Disastrous Effect of Packet Loss on TCP in WANs (2/2)

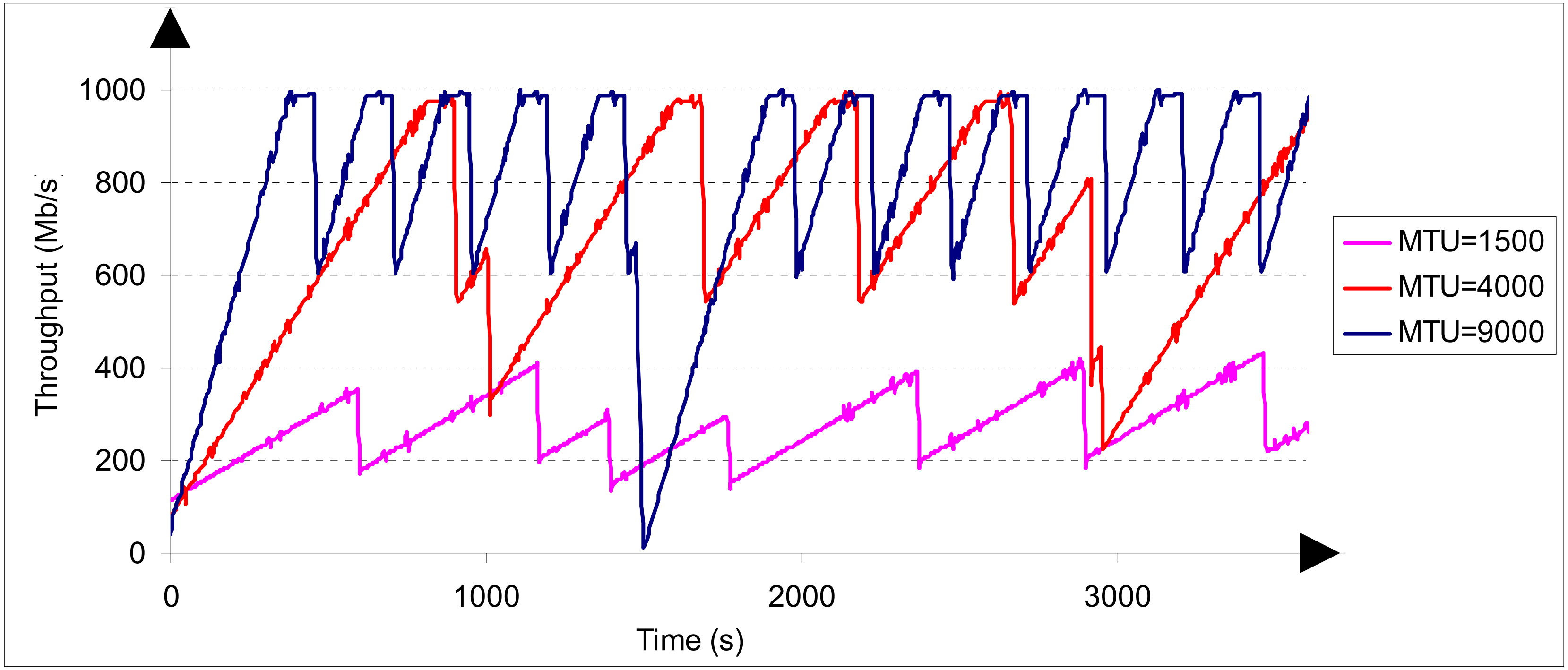
- Long time to recover from a single loss:
  - TCP should react to congestion rather than packet loss (line errors, transient fault in equipment)
  - TCP should recover quicker from a loss
- TCP is much more sensitive to packet loss in WANs than in LANs

Measurements with Different MTUs (1/2)

- **Experimental environment:**
  - Linux 2.4.19
  - Traffic generated by iperf
    - average throughout over the last 5 seconds
  - Single TCP stream
  - RTT = 119 ms
  - Duration of each test: 2 hours
  - Transfers from Chicago to Geneva
- **MTUs:**
  - set on the NIC of the PC (*ifconfig*)
  - POS MTU set to 9180
  - Max MTU with Linux 2.4.19: 9000

Measurements with Different MTUs (2/2)

TCP max: 990 Mbit/s (MTU=9000)
UDP max: 957 Mbit/s (MTU=1500)



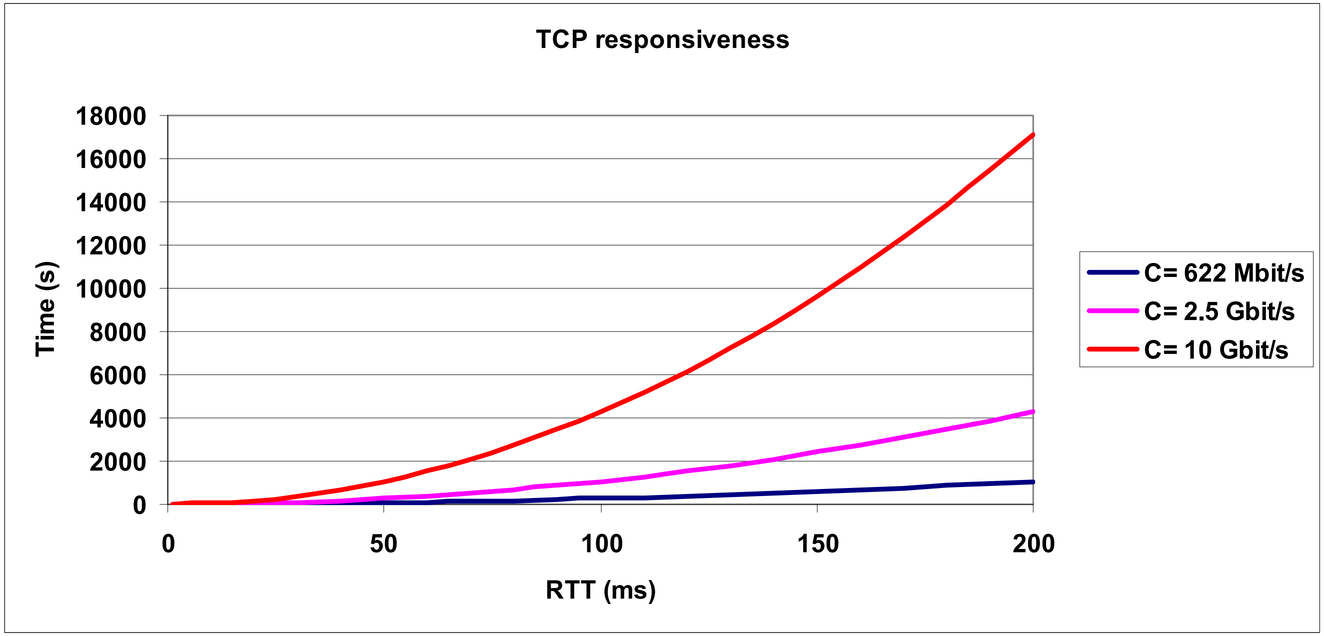
Tools Used to Perform Measurements

- We used several tools to investigate TCP performance issues:
  - Generation of TCP flows: *iperf* and *gensink*
  - Capture of packet flows: *tcpdump*
  - *tcpdump* output=> *tcptrace* =>*xplot*
- Some tests performed with SmartBits 2000
- Currently shopping for a traffic analyzer
  - Suggestions?

Responsiveness

The responsiveness ρ measures how quickly we go back to using the network link at full capacity after experiencing a loss

$$\rho = \frac{C \, . \, RTT^2}{2 \, . \, inc}$$



Characterization of the Problem

inc size = MSS = 1,460

Capacity	RTT	# inc	Responsiveness
9.6 kbit/s (WAN 1988)	max: 40 ms	1	0.6 ms
10 Mbit/s (LAN 1988)	max: 20 ms	8	~150 ms
100 Mbit/s (LAN 2002)	max: 5 ms	20	~100 ms
622 Mbit/s	120 ms	~2,900	~6 min
2.5 Gbit/s	120 ms	~11,600	~23 min
10 Gbit/s	120 ms	~46,200	~1h 30min

What Can We Do?

- To achieve high throughput over high latency/bandwidth network, we need to:
  - Set the initial slow start threshold (*ssthresh*) to an appropriate value for the delay and bandwidth of the link
  - Avoid loss
    - by limiting the max size of *cwnd*
  - Recover fast in case of loss:
    - larger *cwnd* increment => better responsiveness
    - larger packet size (Jumbo frames)
    - Less aggressive decrease algorithm

Delayed ACKs

- RFC 2581:

$$cwnd_{i+1} = cwnd_i + \frac{SMSS \bullet SMSS}{cwnd_i}$$

- Assumption: one ACK per packet
- Delayed ACKs: one ACK every second packet
- Responsiveness multiplied by two
- Disastrous effect when RTT large and *cwnd* large

Research Directions

- New fairness principle
- Change multiplicative decrease:
  - do not divide by two
- Change additive increase
  - successive dichotomies
  - local and global stability
- More stringent definition of congestion
- Estimation of the available capacity and bandwidth*delay product:
  - on the fly
  - cached

Related Work

- Sally Floyd, ICIR: Internet-Draft "High Speed TCP for Large Congestion Windows"
- Steven Low, Caltech: Fast TCP
- Tom Kelly, U Cambridge: Scalable TCP
- Web100 and Net100
- PFLDnet 2003 workshop:
    - http://www.datatag.org/pfldnet2003/