Performance Analysis of Distributed Systems over ATM Networks

Brian Tierney (bltierney@lbl.gov),
William Johnston(wejohnston@lbl.gov),
Gary Hoo, Guojun Jin, Jason Lee

Imaging and Distributed Computing Group
(http://www-itg.lbl.gov)
Lawrence Berkeley National Laboratory
Berkeley, CA 94720

This work is jointly supported by ARPA - CSTO, and by the U. S. Dept. of Energy, Energy Research Division, Mathematical, Information, and Computational Sciences office, under contract DE-AC03-76SF00098 with the University of California.

Introduction

♦ The MAGIC testbed consists of a large-scale, high-speed, ATM network: It is a heterogeneous collection of

- ATM switches and computing platforms
- IP over ATM implementations
- "middleware" (distributed services), etc.,

all of which must cooperate in order to make a complex application operate at high speed

Introduction

♦ When building high-speed network-based distributed services, we often observe unexpectedly low network throughput and/or high latency - the reason for which is usually not obvious

The bottlenecks can (and have been) in any of the components:

- the applications
- the operating systems
- the device drivers, the network adapters on either the sending or receiving host (or both)
- the network switches and routers, and so on

Introduction

- **Difficult to track down performance problem because of the complex interaction between the many distributed system components resulting in problems in one place being most apparent somewhere else.**

- **Tools such at *ttcp* and *netperf* are somewhat useful but don't model real distributed applications, which are complex, bursty, and have more than one connection in and/or out of a given host at one time**

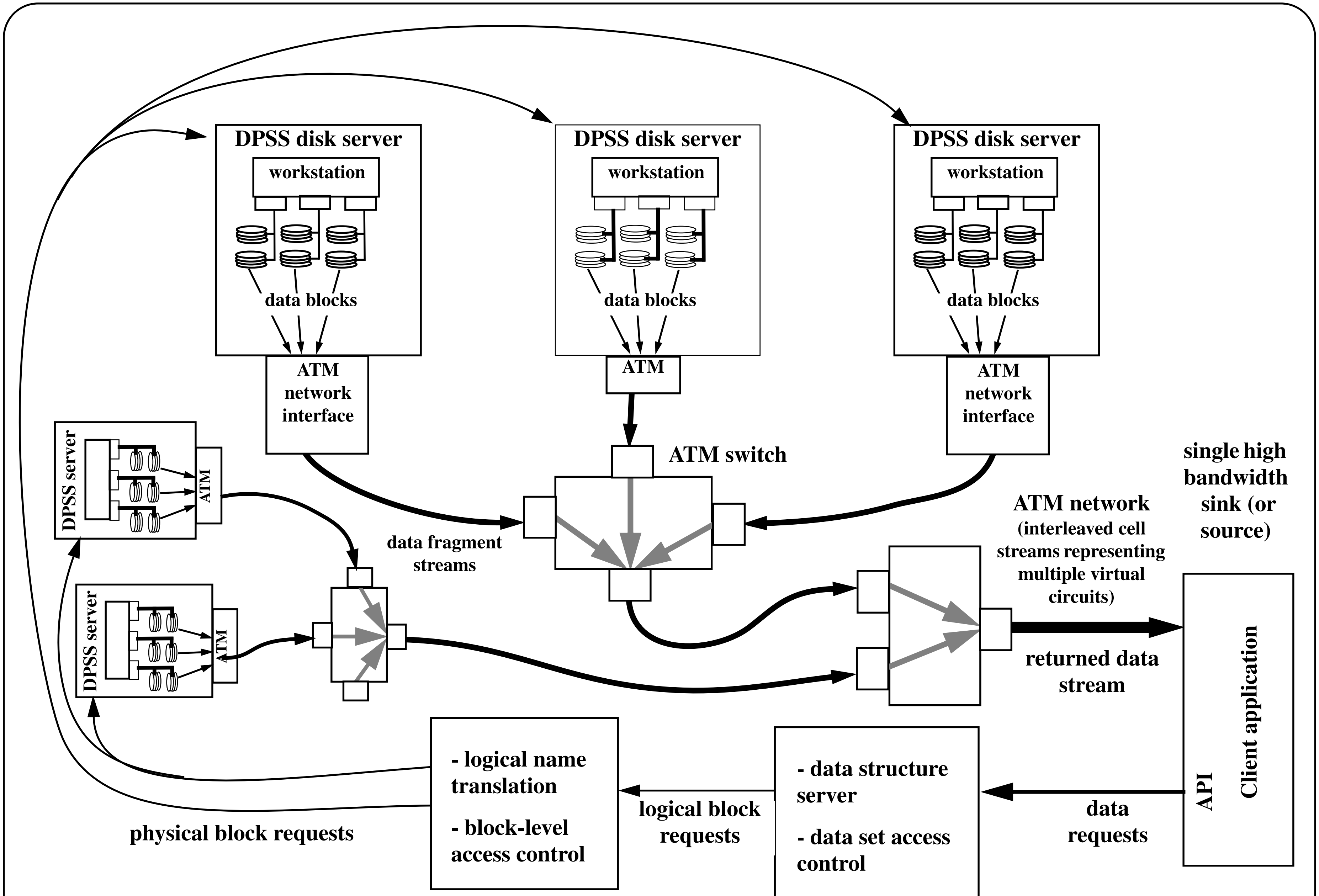
Introduction

- ♦ **Have developed a methodology and tools for monitoring, under realistic operating conditions, the behavior of all the elements of the application-to-application communications path in order to determine exactly what is happening within this complex system**
  - **we have instrumented our applications to do timestamping and logging at every critical point**
  - **modified some Unix network and operating system monitoring tools to log "interesting" events using a common log format**
- ♦ **The goal of this work is to produce predictable, high-speed components that can be used as building blocks for high-performance applications, rather than having to "tune" the applications top-to-bottom as is all too common today**

DPSS Architecture

- The Distributed-Parallel Storage System (DPSS, aka ISS) is a network-striped disk array designed to supply high speed data streams to other processes in the network
- DPSS uses parallel operation of, distributed servers to supply image streams fast enough to enable various multi-user, "real-time", virtual reality-like applications in an Internet / ATM environment
- At the application level, the DPSS is a persistent cache of named objects, at the storage level it is a logical block server (It is not a reliable tertiary storage system.)

DPSS Architecture



Architecture for Distributed-Parallel Storage System (MAGIC testbed, Image Server System)

MAGIC Testbed

- The DPSS and TerraVision are components of the MAGIC Gigabit Testbed project
- The monitoring work is a cooperative effort of the MAGIC participants:
  - Earth Resources Observation System Data Center, U.S. Geological Survey (EDC)
  - Lawrence Berkeley National Laboratory, U.S. Department of Energy (LBNL)
  - Minnesota Supercomputer Center, Inc. (MSCI)
  - SRI International (SRI)
  - University of Kansas (KU)
  - CNRI/MITRE Corporation (project coordination)
  - Sprint
  - U S WEST Communications, Inc.

MAGIC Testbed

♦ **Typical DPSS implementation**
  - **4 - 5 UNIX workstations (e.g. Sun SPARCStation, DEC Alpha, SGI, etc.)**
  - **4 - 6 fast-SCSI disks on multiple 2 - 3 SCSI host adaptors**
  - **an ATM network interface**

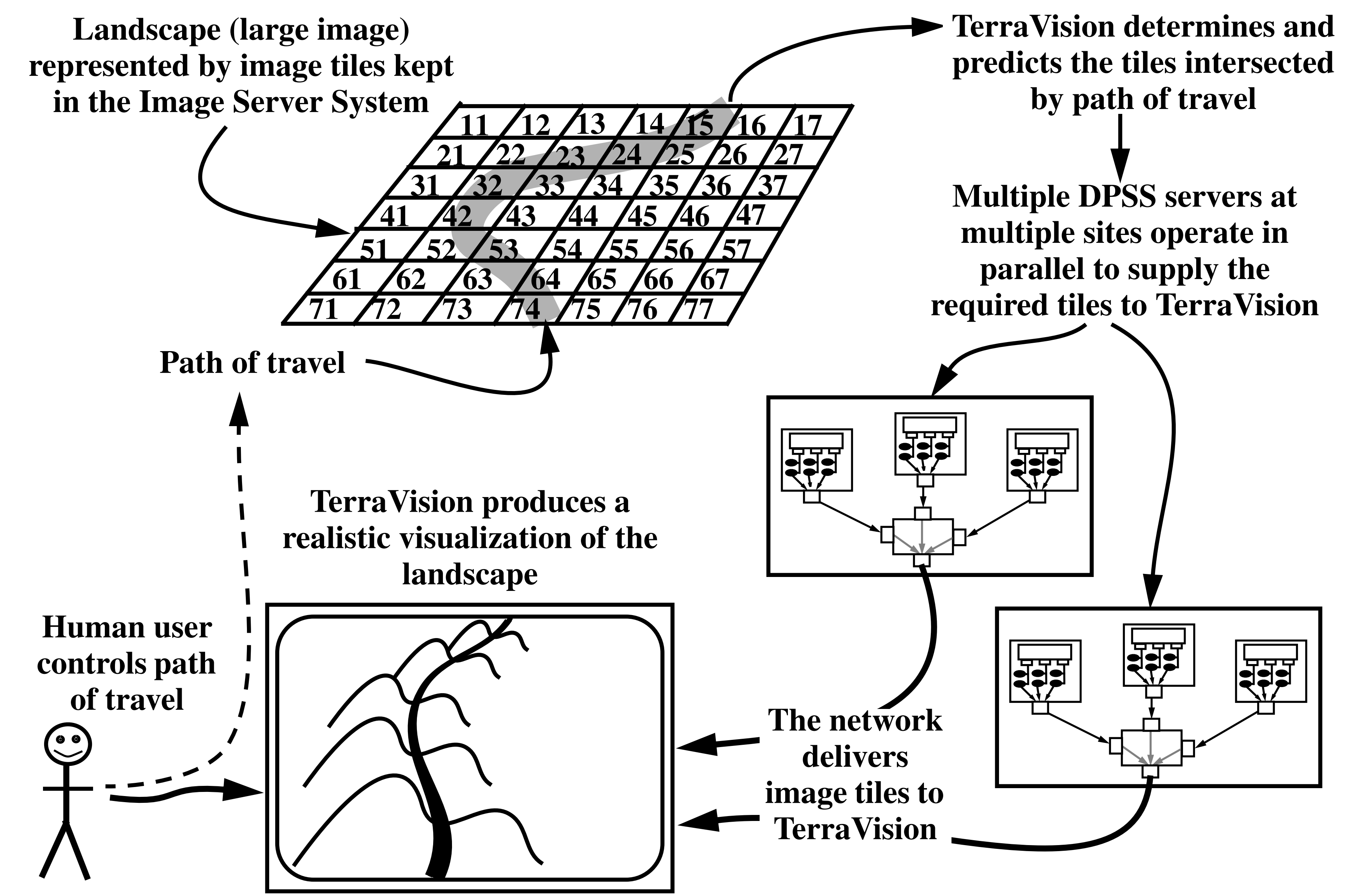
- **this configuration can deliver an aggregated data stream to an application at about 400 Mbits/s (50 Mbytes/s) using these relatively low-cost, “off the shelf” components by exploiting the parallelism provided by approximately five hosts, twenty disks, ten SCSI host adaptors, and five network interfaces**

- **transport is by either TCP or RTP, but most performance testing is currently on the TCP transport**

MAGIC Testbed

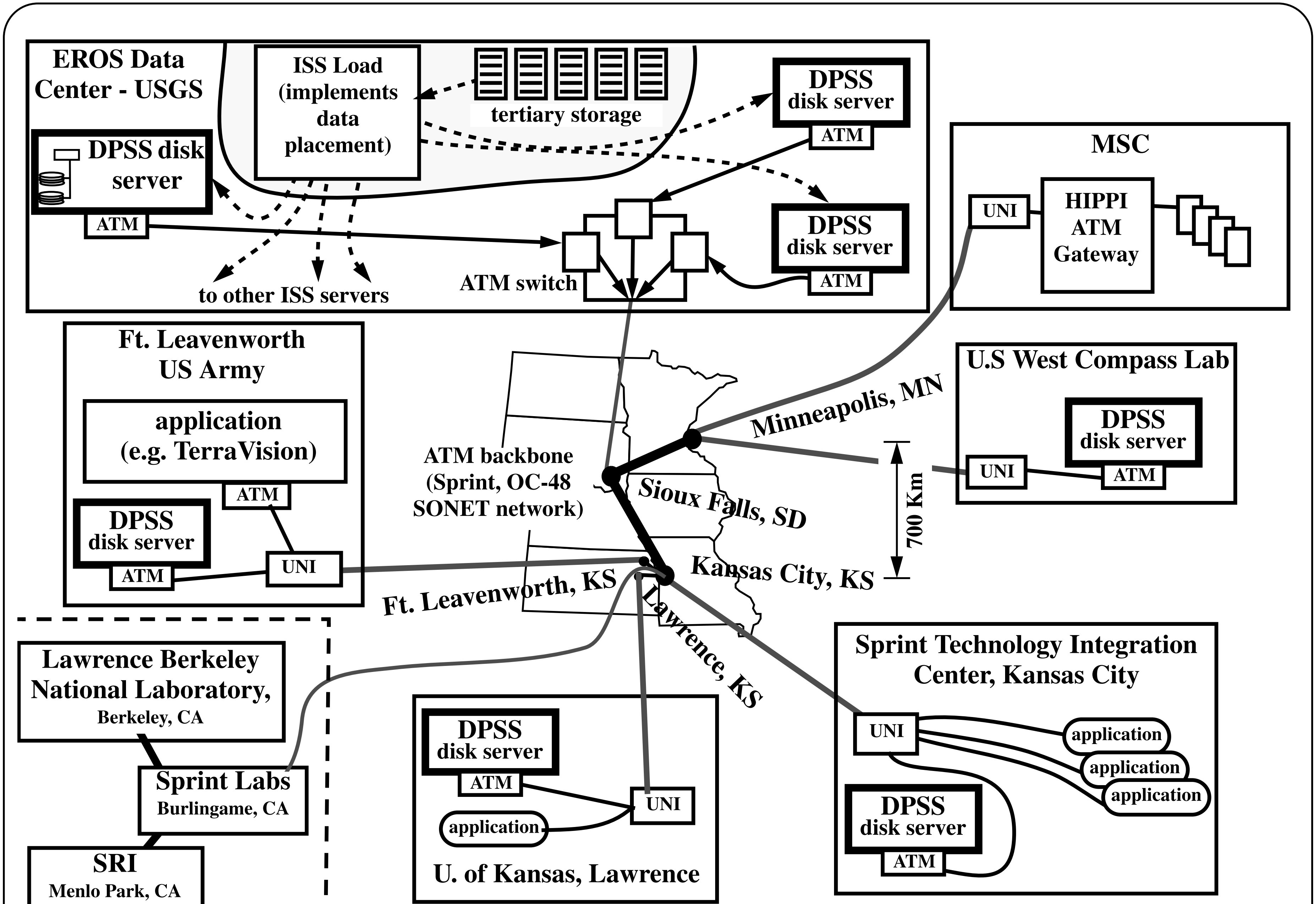
- TerraVision is the prototype application and is embedded in the MAGIC testbed environment

MAGIC Testbed



TerraVision and the DPSS Cooperate to Visualize Landscape

MAGIC Testbed

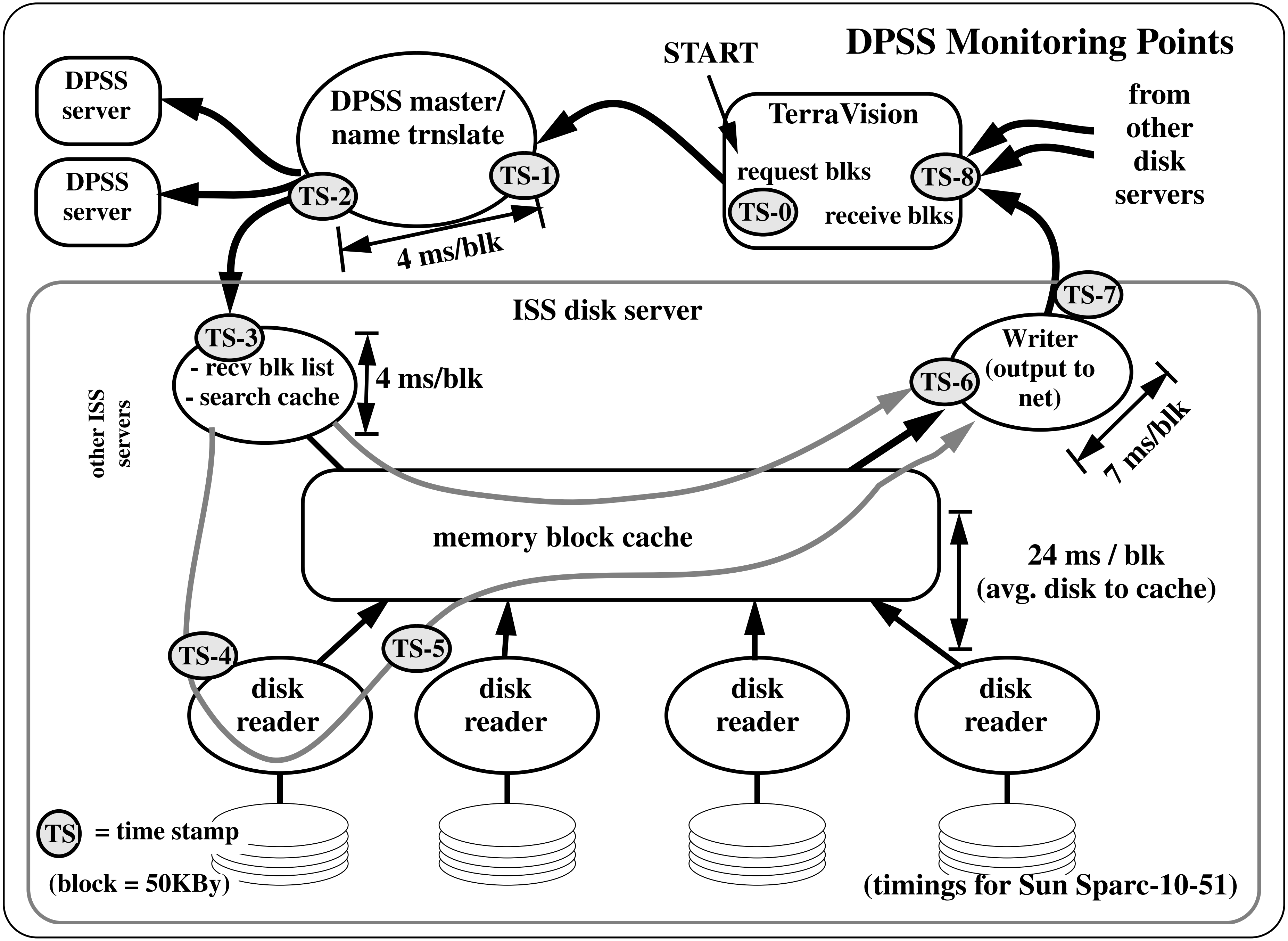


The MAGIC Testbed Distributed Application Environment

Monitoring

- Performance monitoring at the application level involves collecting time stamps at all of the critical points
- Time and other information are made available to the DPSS master through monitoring protocols, and carried as a defined part of the data block structure

Monitoring


DPSS Monitoring Points
START
DPSS server
DPSS server
DPSS master/
name trnslate
TS-1
TS-2
4 ms/blk
TerraVision
request blks
TS-0
receive blks
TS-8
from other disk servers
ISS disk server
TS-3
- recv blk list
- search cache
4 ms/blk
other ISS servers
Writer
(output to net)
TS-6
TS-7
7 ms/blk
memory block cache
24 ms / blk
(avg. disk to cache)
TS-4
TS-5
disk reader
disk reader
disk reader
disk reader
TS = time stamp
(block = 50KBy)
(timings for Sun Sparc-10-51)

Monitoring

- ♦ OS and network monitoring
  - TCP retransmits
  - CPU usage (user and system)
  - CPU interrupts
  - AAL 5 information
  - ATM switch buffer overflows
  - ATM hosts adapter buffer overflow

Monitoring

- **Common logging format:**

  - **keyword; hostname; seconds; nano-sec; data; data; data;......;**
  - **"keyword" - an identifier describing what is being logged - e.g. a reference to the program that is doing the logging: DPSS_SERV_IN, VMSTAT_SYS_CALLS, NETSTAT_RETRANSSEGS, TV_RQ_TILE**
  - **"data" elements (any number) are used to store information about the logged event - for example:**
    - **for NETSTAT_RETRANSSEGS event, one data element: the number of TCP retransmits since the previous event**
    - **for DPSS_START_WRITE event, data elements contain the logical block name, the data set ID, a "user session" ID, and an internal DPSS block counter**

Monitoring

NETSTAT_OUTDATASEGS; 806706009; 652917; 4817378;
NETSTAT_RETRANSSEGS; 806706009; 652917; 16395;
NETSTAT_RETRANSBYTES; 806706009; 652917; 111841701;
NETSTAT_OUTDATASEGS; 806706009; 760199; 0;

VMSTAT_INTR; 806706009; 546568; 71612733;
VMSTAT_SYS_CALLS; 806706009; 546568; 2794466576;
VMSTAT_USER_TIME; 806706009; 546568; 2;
VMSTAT_SYS_TIME; 806706009; 546568; 5;
VMSTAT_INTR; 806706009; 646444; 3;

APP_SENT; 824951202; 824949; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_MASTER_IN; 824951202; 832232; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_MASTER_OUT; 824951202; 865724; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_SERV_IN; 824951202; 877494; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_START_READ; 824951202; 885279; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_END_READ; 824951202; 909439; 34; 78; 45; 0; 6; 5; 198.207.141.6;
ISS_START_WRITE; 824951202; 910743; 34; 78; 45; 0; 6; 5; 198.207.141.6; 49264
APP_RECEIVE; 824951210; 914210; 34; 78; 45; 0; 6; 5; 198.207.141.6;
APP_SENT; 824951202; 824949; 34; 76; 45; 0; 6; 3; 198.207.141.6;

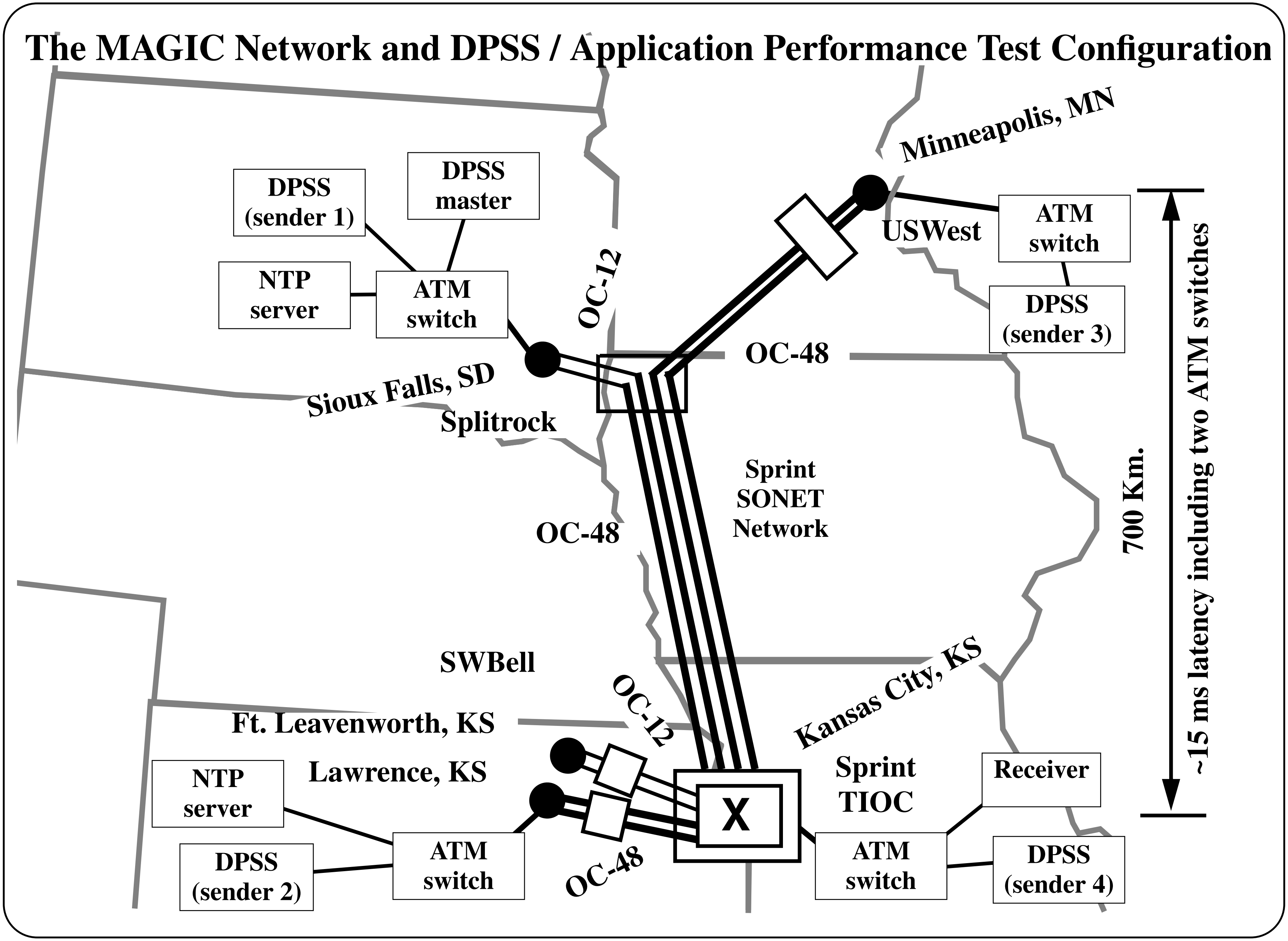
Monitoring

- **NTP is used to synchronize time stamps**
  - **All MAGIC hosts run *xntpd*, which synchronizes the clocks of each host both to time servers and to each other**
  - **The MAGIC backbone segments are used to distribute NTP data, allowing us to synchronize the clocks of all hosts to within about 250 microseconds of each other, but...**
    - **many different sys admins (harder to synchronize than clocks)**
    - **the systems have to stay up for a significant length of time for the clocks to converge**

Monitoring

- ♦ **Tools**
  - ***netstat*** **and** ***vmstat*** **modified to poll and report continuously**
    - **(Currently poll at 100 ms, so this data is accurate to +/- 50 ms)**
  - **SNMP queries to switches and network interfaces**
    - **switch buffer overflows (!)**
- ♦ **The monitoring environment**

Monitoring



The MAGIC Network and DPSS / Application Performance Test Configuration
Minneapolis, MN
DPSS (sender 1)
DPSS master
NTP server
ATM switch
OC-12
USWest
ATM switch
DPSS (sender 3)
OC-48
Sioux Falls, SD
Splitrock
Sprint SONET Network
OC-48
700 Km.
~15 ms latency including two ATM switches
SWBell
OC-12
Kansas City, KS
Ft. Leavenworth, KS
Lawrence, KS
NTP server
Sprint TIOC
Receiver
X
DPSS (sender 2)
ATM switch
OC-48
ATM switch
DPSS (sender 4)

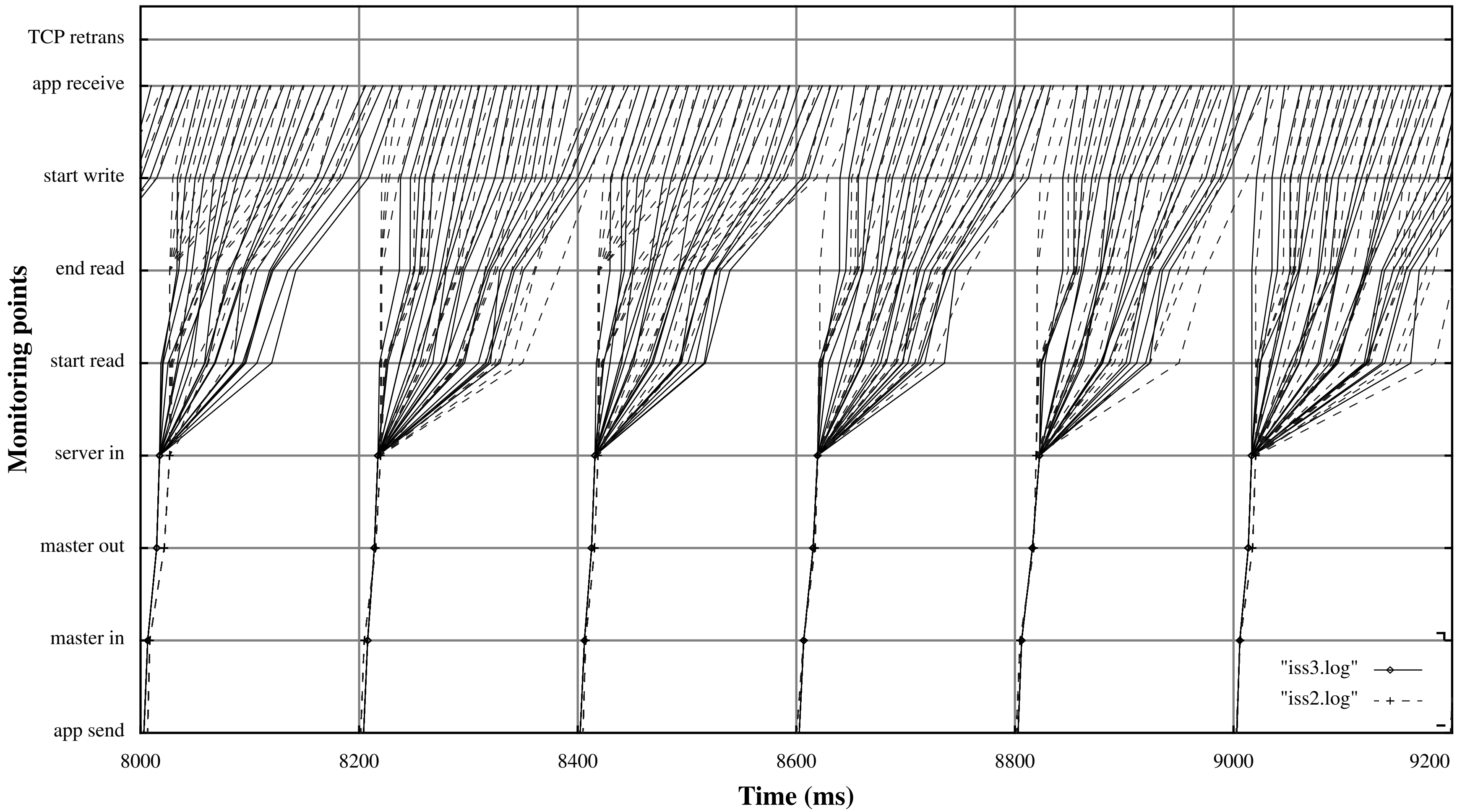
Monitoring

- **System performance experiments**
  - **TerraVision (the "real" application) is run interactively**
  - **traces of data requests are collected**
  - **tv_sim is both a parameter and trace driven simulator (in trace mode, the application can be - and usually is - "speeded up")**

End-to-End, Top-to-Bottom Analysis

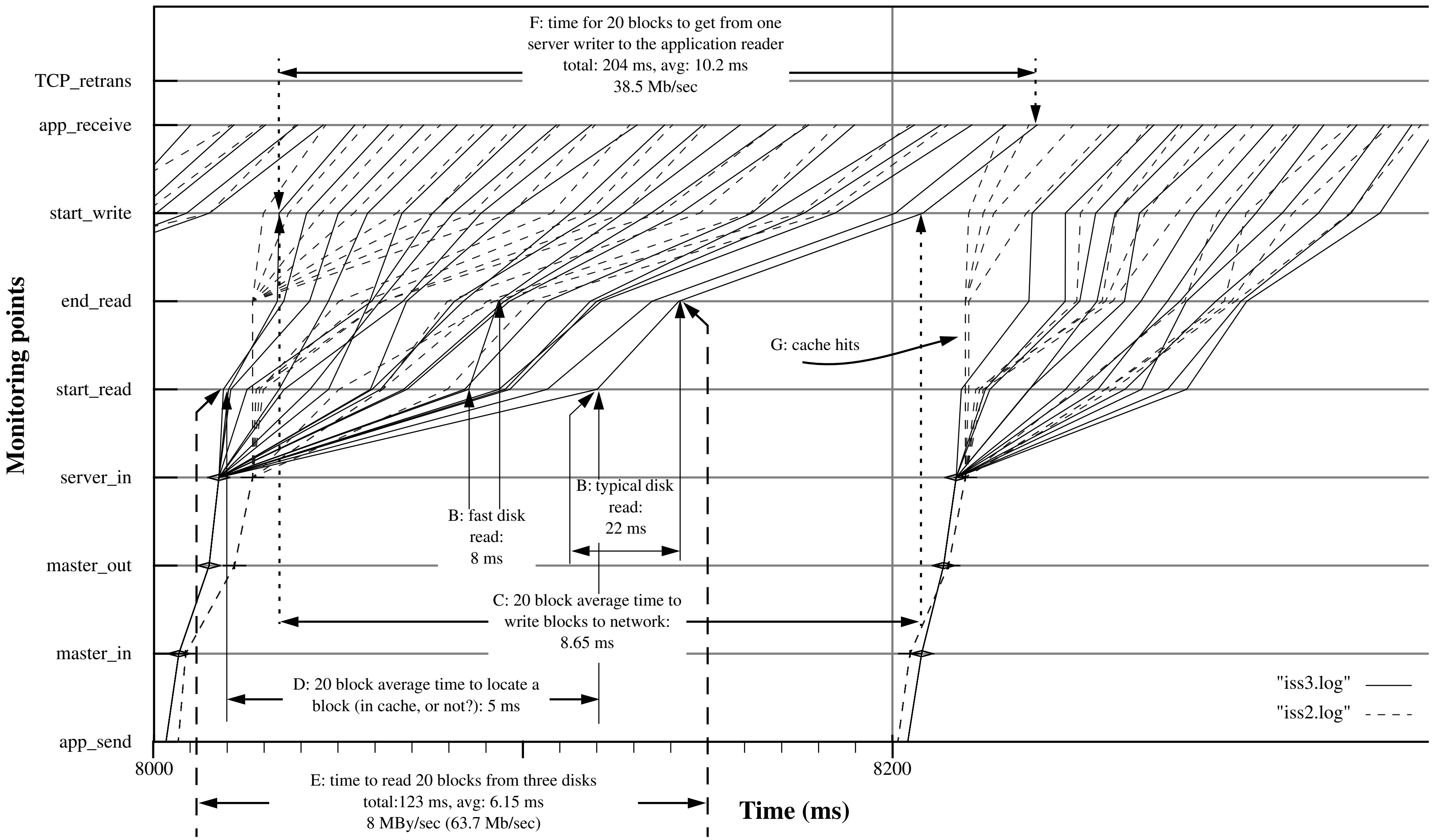
- **End-to-end, top-to-bottom analysis**
  - **Two server, ATM LAN case**
    - **data requests are sent from the application every 200 ms (the nearly vertical lines starting at *app send* monitor point)**
    - **each line represents the history of a data block as it moves through the end-to-end path**
    - **initial single lines fan out as the request lists are resolved into individual data blocks**
    - **each block is first individually represented as requests in the read queue**

End-to-End, Top-to-Bottom Analysis



Two Server Test (ATM LAN, two SS-20s as servers, tv_sim on DEC 3000/600)

End-to-End, Top-to-Bottom Analysis



Detail From a Two Server, ATM LAN, Experiment

End-to-End, Top-to-Bottom Analysis

- If two lines cross in the area between *start read* and *end read*, this indicates a read from one disk was faster that a read from another disk

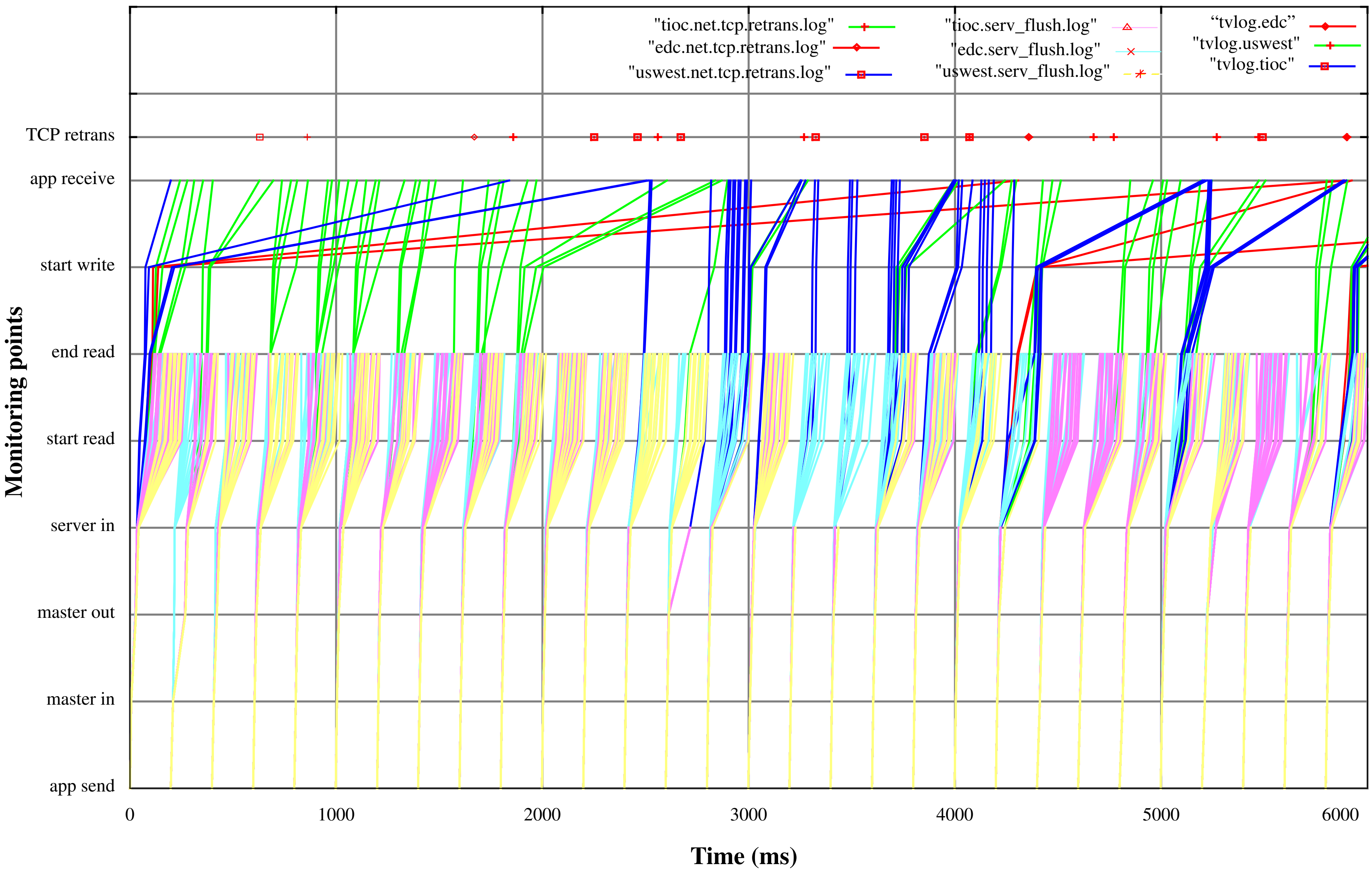
- lines terminate at *end read*, a few also end at *start read*
  If a data request is not satisfied before the next request list arrives, it is flushed (discarded), from all the queues, but not from the memory cache.
  (Assumption is that a new data block request is more important than any unsatisfied previous requests, so discard all pending requests. This approach ensures that the entire data pipeline stays full, and that disk server resources are never idle.)

- some requested data are found in the cache
  (The nearly vertical lines on the left of groups of requests)

End-to-End, Top-to-Bottom Analysis

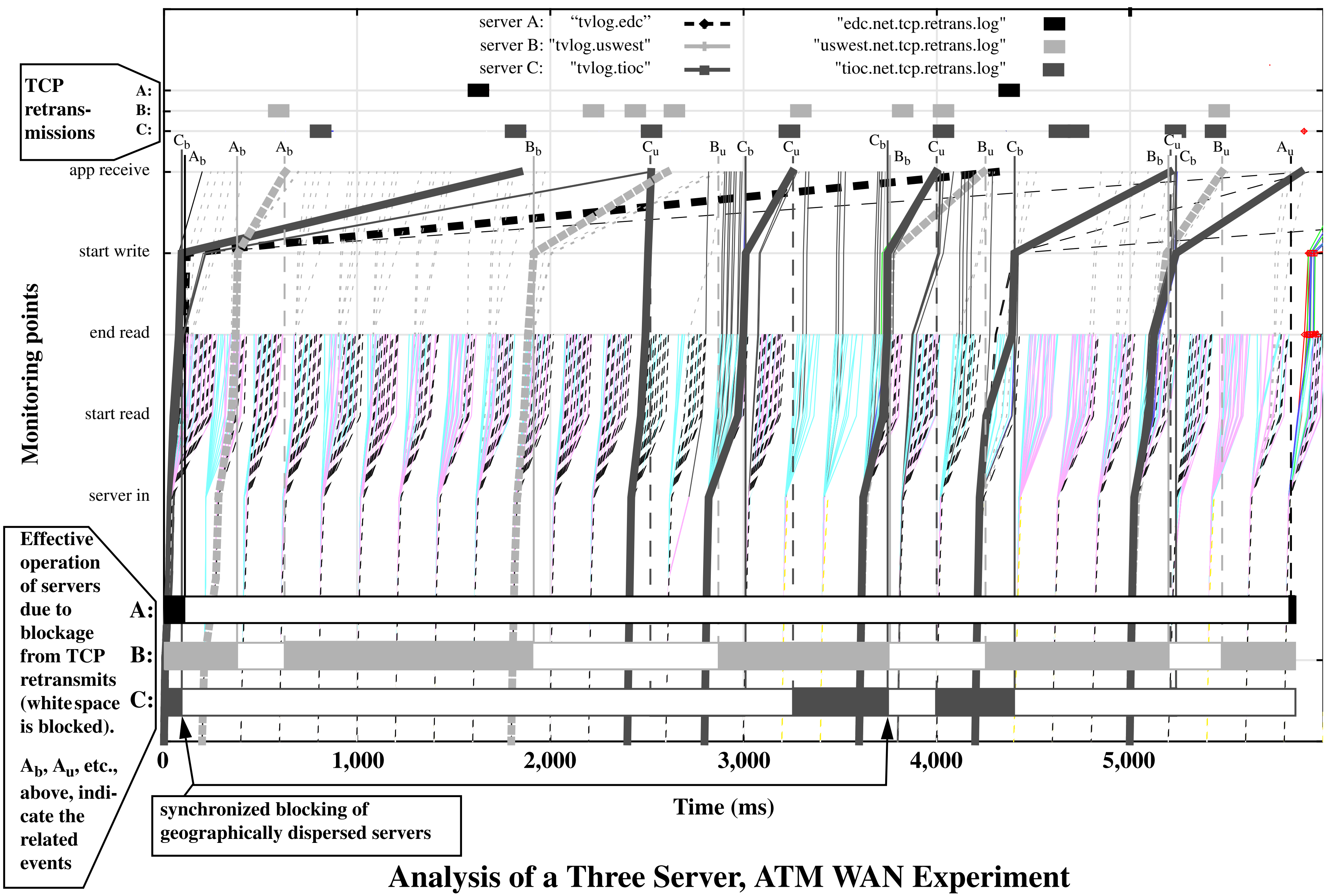
- **Three server, ATM WAN case**

End-to-End, Top-to-Bottom Analysis



Three Server Test (MAGIC ATM WAN, three SS-10s as servers, tv_sim on SGI Onyx)

End-to-End, Top-to-Bottom Analysis



Analysis of a Three Server, ATM WAN Experiment

End-to-End, Top-to-Bottom Analysis

- **Notes on the 3 server graph:**
  - **TCP retransmissions and some very long delays (up to 5500 ms)**
    (Once a block is written to the TCP socket, the user level flushes have no effect, and TCP will re-send the block until transmission is successful, even though the data is likely no longer needed and is holding up newer data.)
  - **in this particular case we suspect local ATM switch congestion, but at the time this experiment was done we could not get cell drop statistics from the switch**
  - **seems to be some synchronization among the long-delay situations which might indicate the interleaved cell problem reported by Romanow and Floyd**

(If so, an implementation of the Romanow and Floyd, Early Packet Discard (EPD) algorithm in the switches might help - none were available in early-mid 1995 when this test was done.)

- the RTP transport model does not suffer from this type of retransmission problem because a single block is never retransmitted by the disk server
  (In the coming year we will investigate the performance of RTP transport, and its interaction with the network.)
- this graph also shows an overly ambitious prediction strategy on the part of the client, which requests a lot of data that is never used

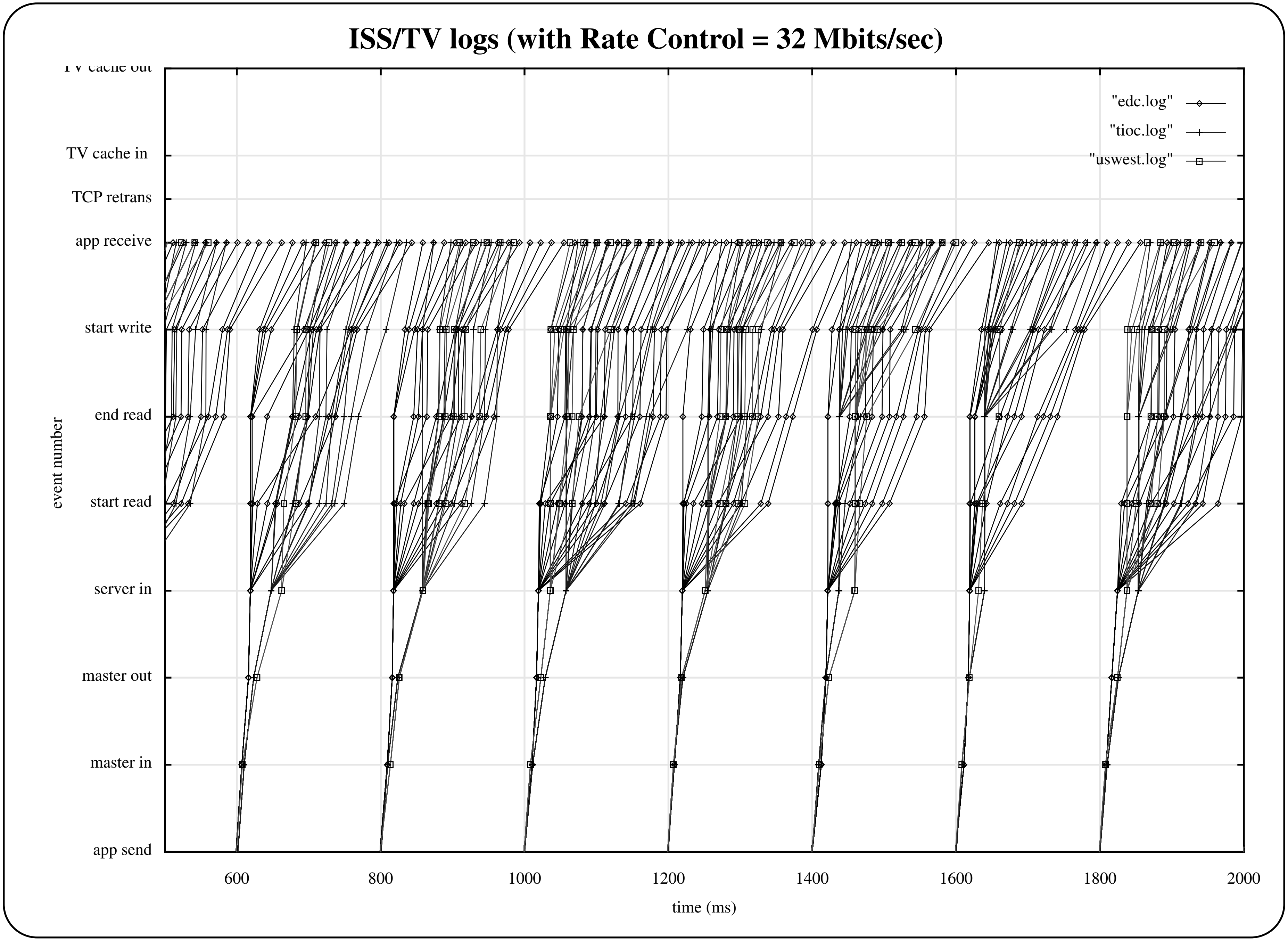
End-to-End, Top-to-Bottom Analysis

- The evolution of the IP over ATM infrastructure
  - 1994
    - single interfaces were slow
    - multiple interfaces were no faster (poor independence of data paths - nothing was multi-threaded)
    - switches dropped cells (silently)
  - 1995
    - single interfaces are faster
    - multiple interface data path independence is improving
    - switches still drop cells (silently)
  - 1996
    - single interfaces and systems are approaching wire speed
    - multiple interfaces better still, but not ideal
    - switches do report cell loss
    - cell pacing now critical for all platforms

End-to-End, Top-to-Bottom Analysis

- **Three server WAN case revisited**
  - **all three servers are now cell paced to ensure that the switch output port is never overrun**

End-to-End, Top-to-Bottom Analysis



End-to-End, Top-to-Bottom Analysis

♦ References

- Cavanaugh, John, Timothy Salo, “Internetworking with ATM WANs”, (available as http://www.msci.magic.net/docs/magic/ip-atm.ps)
- Evans, Joseph B., Victor S. Frost, Gary J. Minden, “TCP and ATM in Wide Area Networks”, CNRI Gigabit Network Workshop ‘94. (http://www.magic.net/tcp/overview.html)
- Jonkman, Roelof J.T., “An Overview of NetSpec”, Telecommunications & Information Sciences Laboratory, University of Kansas. (http://www.tisl.ukans.edu/Projects/AAI/products/netspec/)
- Lau, S, and Y. Leclerc, “TerraVision: a Terrain Visualization System,”, Technical Note 540, SRI International, Menlo Park, CA, Mar. 1994. Also see: http://www.ai.sri.com/~magic/terravision.html
- Richer, I. and B. B. Fuller, “An Overview of the MAGIC Project,” M93B0000173, The MITRE Corp., Bedford, MA, 1 Dec. 1993. (Available from http://www.magic.net/MAGIC_Summary.ps.)
- Romanow, A., and Floyd, S., “Dynamics of TCP Traffic over ATM Networks.” IEEE JSAC, V. 13 N. 4, May 1995, p. 633-641. (An earlier version appeared in SIGCOMM ‘94, August 1994, pp. 79-88.) See http://ftp.ee.lbl.gov/floyd/epd.html
- Tierney, B., Johnston, W., Chen, L.T., Herzog, H., Hoo, G., Jin, G., Lee, J., “Using High Speed Networks to Enable Distributed Parallel Image Server Systems”, Proceedings of Supercomputing ‘94, Nov. 1994, LBL-35437. Available from http://www-itg.lbl.gov/ISS/papers.html.)

End-to-End, Top-to-Bottom Analysis

- **For more information see**

  http://www-itg.lbl.gov/DPSS

  or

  http://www.magic.net

  on the World Wide Web.

Image
Server
System