Communication Facilities for Database Transaction Processing

by

Prof. Bharat Bhargava
Dept. of Computer Science
Purdue University
West Lafayette, IN 47907
USA

317-494-6013
317-494-0739 (Fax)
bb@cs.purdue.edu

Table of Content

- Statement of problem

- Assumption

- Transaction processing benchmark

- RAID distributed database system

- Related projects

- LAN communication for DTP

- Evaluation of RAID communication

- WAN communication for DTP

Legend

DTP	Distributed Transaction Processing
LAN	Local Area Network
WAN	Wide Area Network
IPC	Inter-Process Communication
RPC	Remote Procedure Call

Statement of the Problem

How to control and improve the performance of distributed transaction processing (DTP) system by improving the communication and by exploiting the knowledge about network characteristics, architecture, and dynamics?

Assumptions

- DTP is communication intensive.

- The conventional communication scheme cannot meet the performance requirement.

- Wide area networks are characterized by large number of sites, variable communication delay, and high message loss rate etc.

- Distributed transaction processing experiences a serious performance degradation during failures or changing of communication conditions.

- It's difficult to conduct experiments and testing in WAN environments.

LAN vs WAN

Issues	LAN	WAN
Scale	small (10-100 sites)	large (over 100 sites)
Geographic span	within a mile	over 100 miles
Topology	bus, ring	hierarchical interconnected irregular mesh
Routing	simple	multiple hops, dynamic
Speed	very high (10-1Gbps)	low (1Kbps-10Mbps)
Error rate	low	high
Variation*	small	large
Ownership	private	public
Access	controlled	no control shared

* variation of parameters such as delay, throughput, and losses.

Motivation

To improve the performance of DTP by efficient communication facility.

To provide structure as well as efficiency.

To control and improve the performance of DTP by surveillance that exploits the knowledge about network characteristics, architecture, and dynamics.

Transaction Processing Benchmark for Measurements

- Relation size: 100 tuples.
- Hot-spot size: 20% of the tuples.
- Hot-spot access: 80% of the actions access hot-spot tuples.
- Type of experiment: close (maintaining the concurrency level constant).
- Number of transactions: 250.
- Transaction length. Average: 6. Variance: 4
- Timeout: one second per action.
- Updates: 10% of the actions are updates.
- Restart policy: rolling restart backoff.
- Concurrency controller: two phase locking protocol.
- Atomicity controller: two phase commit protocol.
- Replication controller: ROWA.
- Concurrency level: one transaction at a time.

RAID Experimental Infrastructure

- RAID laboratory with 5 Sun3/50's and 4 Sparcstation1's, all with local disks, microsecond resolution timers connected by a 10Mb/s Ethernet

- Software to specify, create, and maintain replicated databases

- Extended DebitCredit benchmark (Anon 85):
  - Transaction length, and number of transactions to generate
  - Transaction arrival rate or maximum degree of multiprogramming
  - Ratio of small to large transactions
  - Fraction of actions that are updates
  - Hot spot size and access percent

- Action Driver Simulator:
  - Interprets commands written in our benchmark language

  - Generates *open*- and *closed*- system transactions

- Automated overnight experimental procedure:
  - Reboot machines at 3:00 am
  - Setup environment
    - Unmount non-essential file systems
    - Start up the Oracle (the naming server)
    - Initialize database relations
  - Execute experiments found on the Benchmark directory. For every specified experiment:
    - Start a RAID instance
    - Start the AD simulator to run transactions
    - Terminate the RAID instance
    - Upon termination, RAID servers dump statistics data
  - Next day: digest statistics data into condensed tabular format

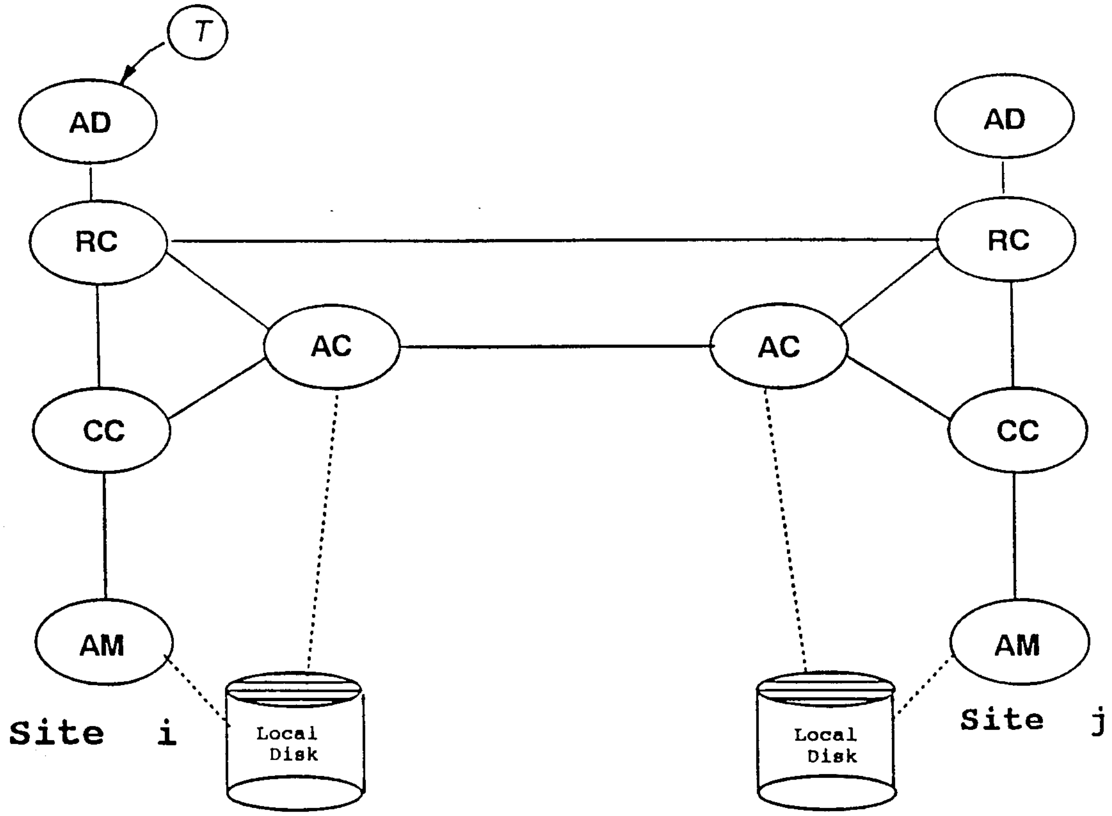
The RAID System

Robust and Adaptable Distributed Database System,
Versions 1 and 2

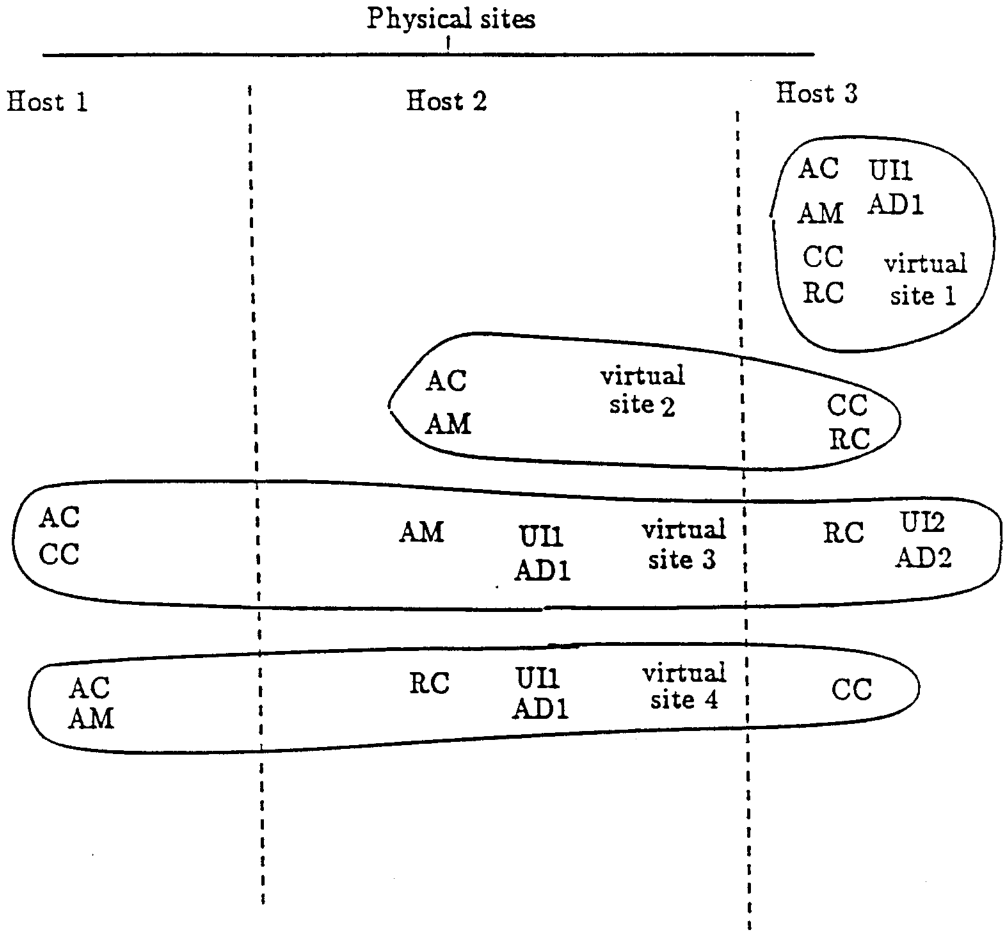
- functional distributed database system, 30k lines of C code, running on UNIX
- experimental system for investigating reliability and adaptability
- modular: independent servers connected by location-transparent UDP/IP-based communication package
- individual servers can adapt between algorithms

IEEE Transactions on Software Engineering, June 1989.

RAID Version 2



RAID Virtual Sites



Distinguishing Features of RAID-V2

- Improved flow of control for adaptability and fault-tolerance
- Directory-based partial replication
- Adaptable quorum-based replication control:
  - An adaptable version of the Quorum Consensus (Gifford 79), in which quorum assignments are determined by a quorum relation
- Dynamically adaptable concurrency control
- Use of XDR to support communication in a heterogeneous system

Adaptability in RAID

- Concurrency Control
- Replication Control
- Distributed Commit
- Network Paritioning
- Reconfiguration after Site Failure
- Inter-server Communication
- Server Relocation
- Parallel vs. Merged Servers (P-Raid)
- Object-relation database model (O-Raid)
- Kernel Extensions (PUSH)

IEEE Transactions on Knowledge and Data Engineering, December 1989.

Related Projects

- Networking/Communication
  - internetworking
  - transport protocols
- Distributed File Systems
  - Cedar, Sprite, Locus, etc
  - Andrew/Coda file systems
- Distributed Operating Systems
  - Amoeba, Athena, Mach, V, *x*-kernel
  - authentication server (Kerberos), group communication
- Database/Transaction Processing
  - Camelot, Argus, Eden, ISIS, Raid
  - Federated databases, multi-databases (R*)
  - Advanced transaction models

Networking/Communication

- IPC in Distributed Systems
  - Accent - IPC in kernel
  - Mach - hand-off scheduling
  - Camelot - a DTP build on top of Mach
  - VMTP - support intra-system model of DTP
  - DUNE - dynamic binding in RPC
- Local IPC
  - Lightweight RPC
  - User-level RPC
- Communication Protocols and LAN
  - VMTP
  - Virtual/layered protocol (x-kernel)
  - atomic broadcast, multicast
- Communication Support for DTP
  - Camelot, RPC

Distributed Operating System

Amoeba

(Vrije, Tanenbaum)

- Employs transaction model.

- Improves performance by using efficient RPC and local protocols.
  (1.10ms on a 3-MIPS machine)

- Uses “Amoeba gateway” to provide transparent communication in WANs without affecting the performance of RPCs in LANs.

- Fast Local Internet Protocol (FLIP).

Distributed File System

Andrew/Coda File System

(CMU, Satyanarayanan)

- Scalability as the design goal
- Improve performance by volume replication and client caching
- Improve performance by segregating files and access by their semantics
- Improve reliability by distinguishing "disconnected" and "connected" modes

LANs Communication for DTP

- Approaches
- Problems
- Solutions
- Our experiences

General Approaches for LANs

- Multi-threaded, light-weight process
  (Camelot)

- Multi-tasking
  (for multiprocessor machines)

- Merged servers
  (sacrifice the structure)

- Efficient RPC
  (Amoeba, Sprite, V, *x*-kernel)

- Primitives for transaction management
  (Mach)

Our Approaches for LANs

- Lightweight protocol
- Simple naming scheme
- Memory mapping and shared memory
- Physical multicasting
- Explicit control passing
- Adaptable to the communication environment

Experiments with Conventional Communication Schemes

- Socket-based Interprocess Communication

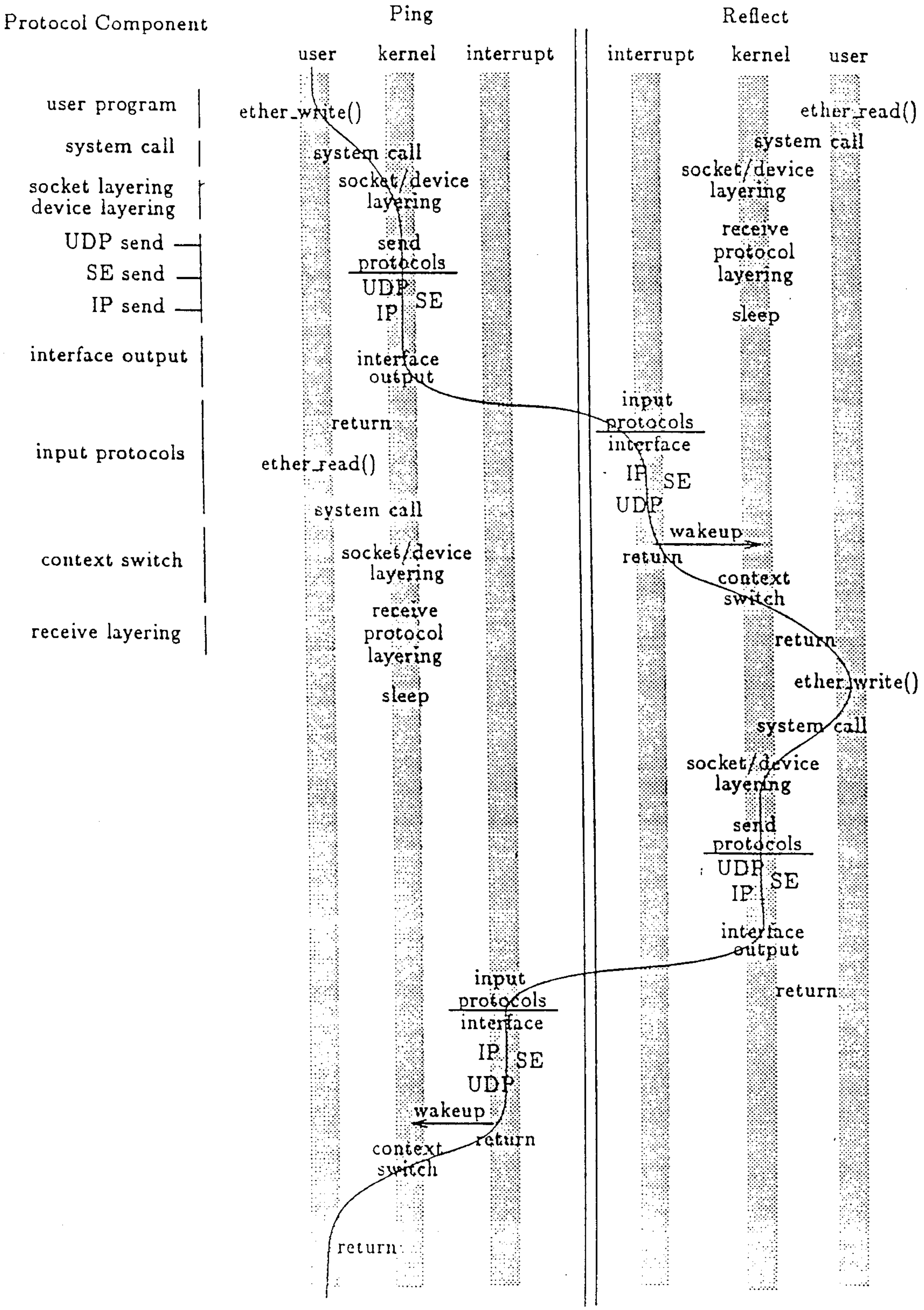
- Local Communication

- Multicast Communication

- Impact on Transaction Processing

Problem with Conventional Communication Schemes

- General purpose communication facilities are expensive
- DTP systems: communication intensive, most are local rather than remote
- Lack of communication support for DTP in some OS
- General purpose multicasting is unusable
- Name resolution is too expensive



Timing diagram for UDP and SE datagram services.

Local IPC Measurements

	MESSAGE SIZE	
	10 Bytes	1000 Bytes
METHOD	Time (ms)	Time (ms)
2 Q Message Passing	2.0	2.9
1 Q Message Passing	2.0	2.9
Named Pipes	2.3	3.9
Shared Memory	5.1	5.5
UDP Communication	4.3	9.6

- message queues are 1/4 the cost of UDP
  - less top-heavy
  - less data copying

Ethernet Experiment

Timing of Components of UDP Send

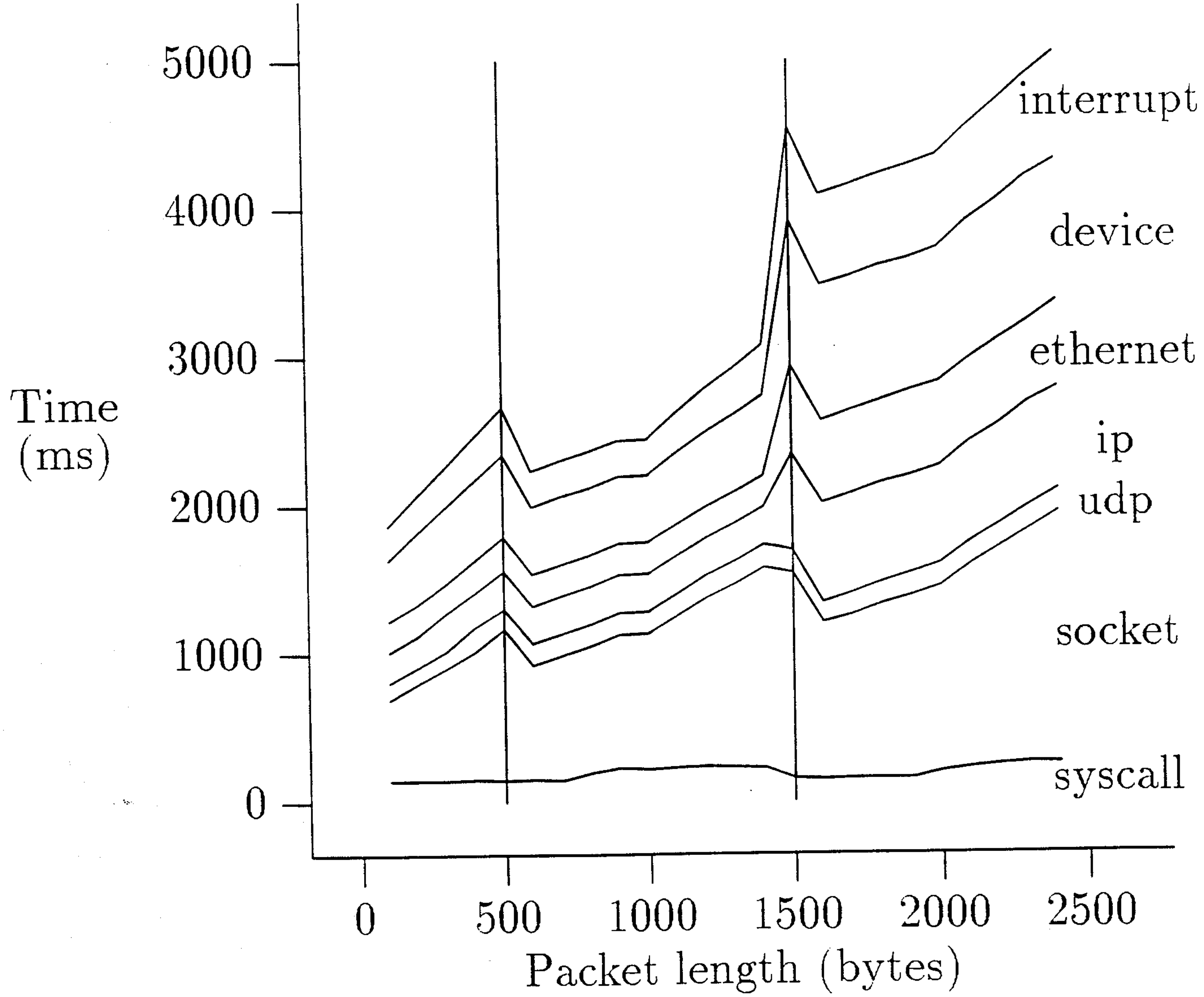
Measurement Instant	Time (in ms)	Protocol Component	Duration (in ms)
start	0	-	0
after socket layers	1.74	socket layers	1.74
after connection	2.28	connection	0.54
end of UDP output	2.58	UDP send	0.30
before IP checksum	3.10	IP send	0.52
end of IP output	3.22	IP checksum	0.12
entire round-trip including receive	7.2		

Timing of Simple Ethernet Components

Protocol Component	Time (in ms)	Measurement Tools
user-level code	0.08	user
system call	0.33	senull - user
device layering	0.61	null - senull
kernel transmit	1.36	kping
mbufs on read	0.45	se_fast_return
context switch	0.40	insomnia - senull
kernel-to-user copy	0.30	se_null_read

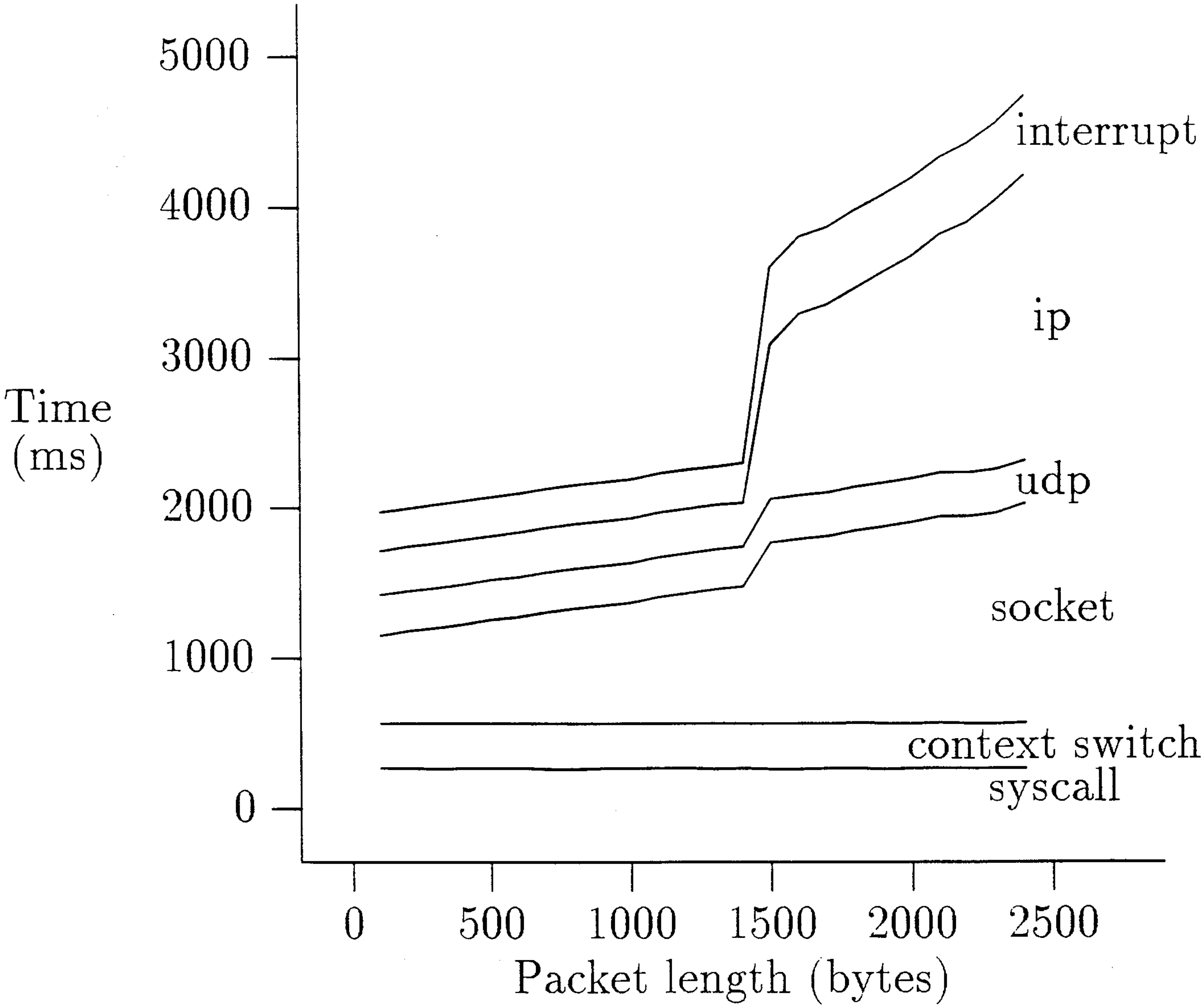
Evaluation of Unix IPC model (I)

Timing for UDP send

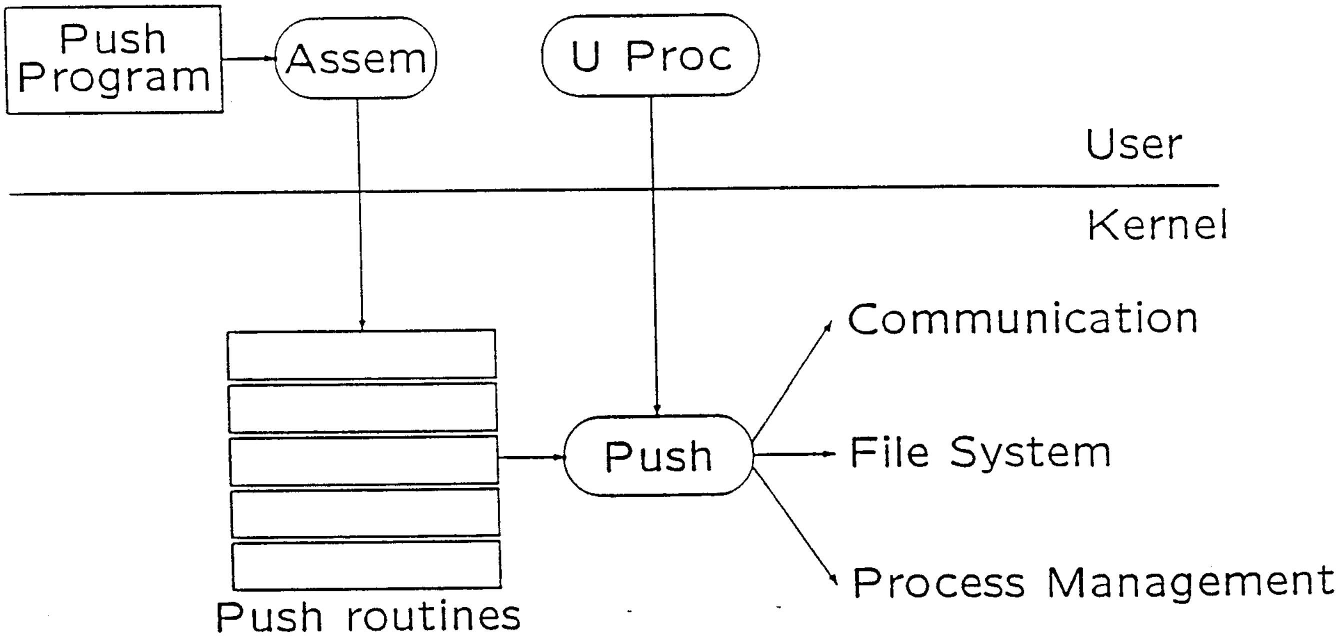


Evaluation of Unix IPC model (II)

Timing for UDP receive



The PUSH system

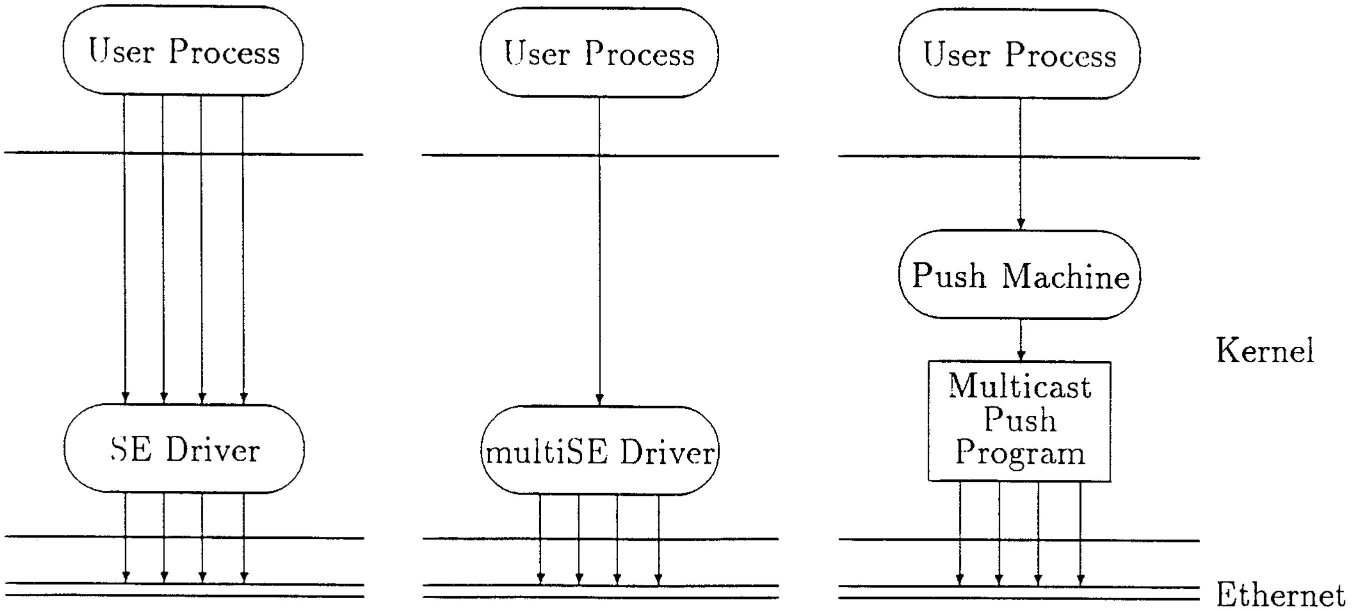


Multicasting Timing (in ms)

Number of destinations	kernel level SE	user level SE	Push
1	1.2	1.2	2.7
5	4.2	5.9	6.6
10	8.0	11.7	11.0
15	11.7	17.5	15.6
20	15.4	23.4	20.2

Evaluation of Multicasting (I)

Approaches for Multicasting



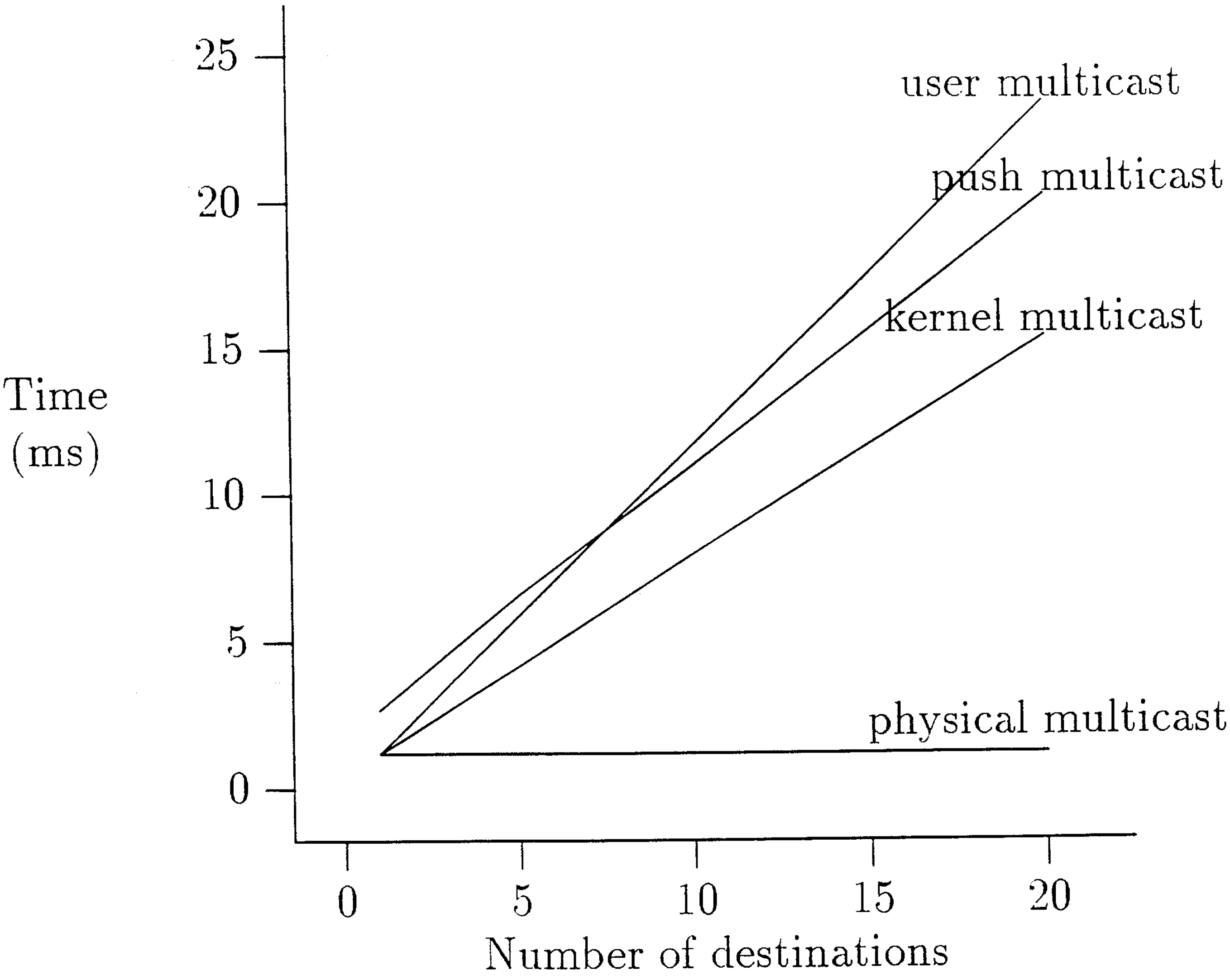
(a) User–level SE multicast

(b) Kernel–level SE multicast

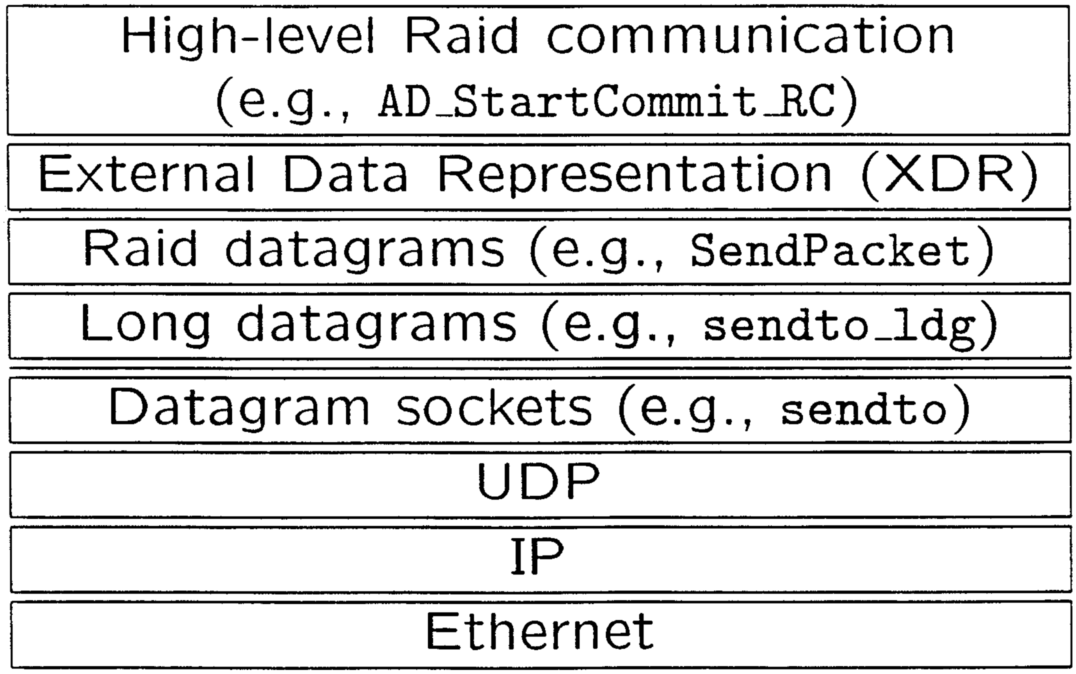
(c) Push multicast

Evaluation of Multicasting (II)

Multicasting cost



Raid Communication Subsystem, V.1



High-level Raid communication (e.g., AD_StartCommit_RC)

External Data Representation (XDR)

Raid datagrams (e.g., SendPacket)

Long datagrams (e.g., sendto_ldg)

Datagram sockets (e.g., sendto)

UDP

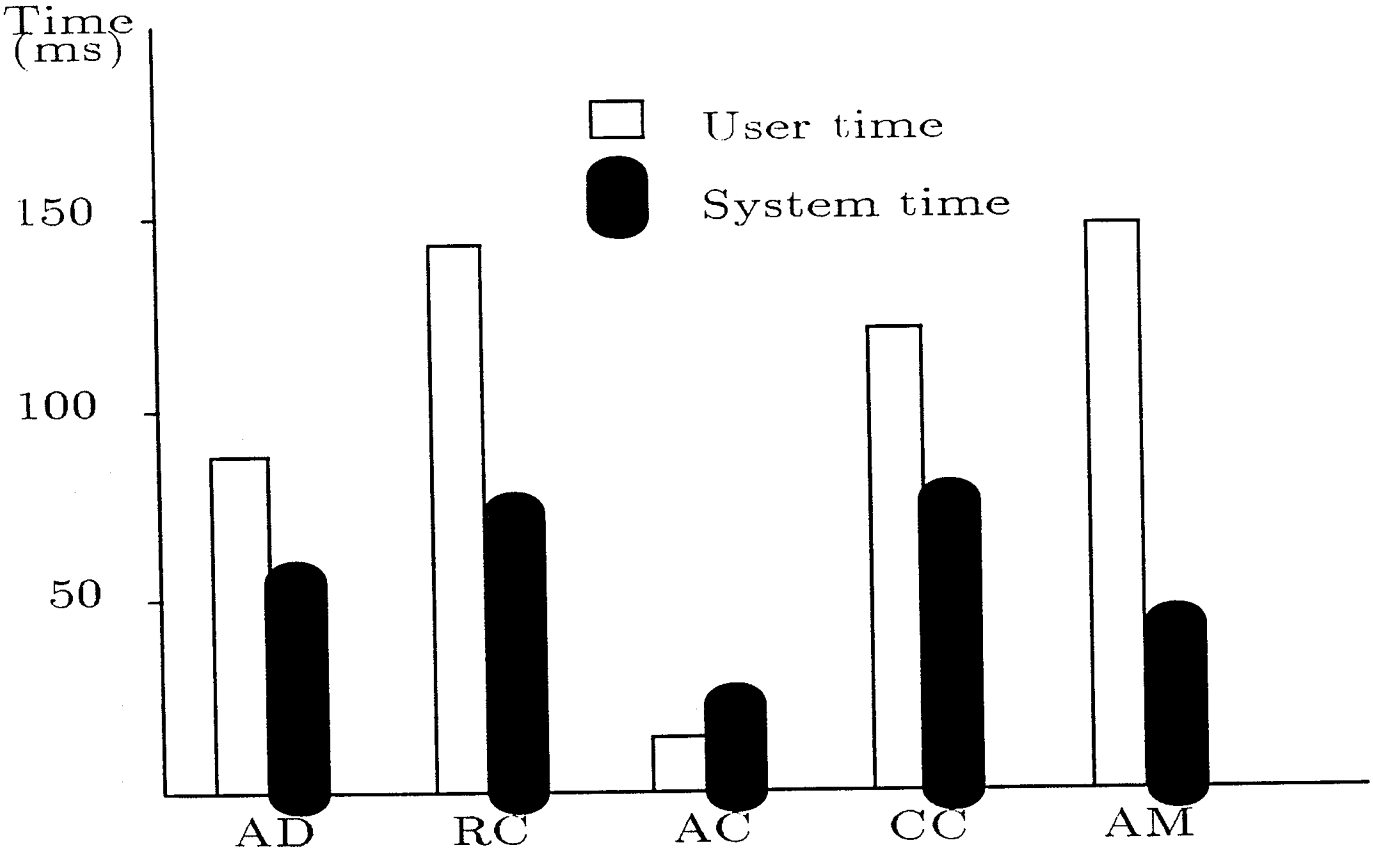
IP

Ethernet

plus a separate process: name server oracle

Evaluation of Raid communication V.1

Raid servers' time



Legend:

AC	=	atomicity controller
CC	=	concurrency controller
AM	=	access manager
RC	=	replication controller
AD	=	action driver

RAID Elapsed Time for Transactions

transaction	1 site	2 sites	3 sites	4 sites
select one tuples	0.3	0.3	0.4	0.4
select eleven tuples	0.4	0.4	0.4	0.4
insert twenty tuples	0.6	0.6	0.8	0.8
update one tuple	0.4	0.4	0.4	0.4

RAID Atomicity Control CPU Time

	1 site		2 sites		3 sites		4 sites	
transaction	user	sys	user	sys	user	sys	user	sys
select one tuples	0.04	0.14	0.06	0.14	0.04	0.10	0.08	0.24
select eleven tuples	0.04	0.08	0.02	0.04	0.06	0.12	0.06	0.10
insert twenty tuples	0.20	0.16	0.08	0.14	0.10	0.12	0.08	0.10
update one tuple	0.04	0.10	0.06	0.12	0.06	0.16	0.06	0.16

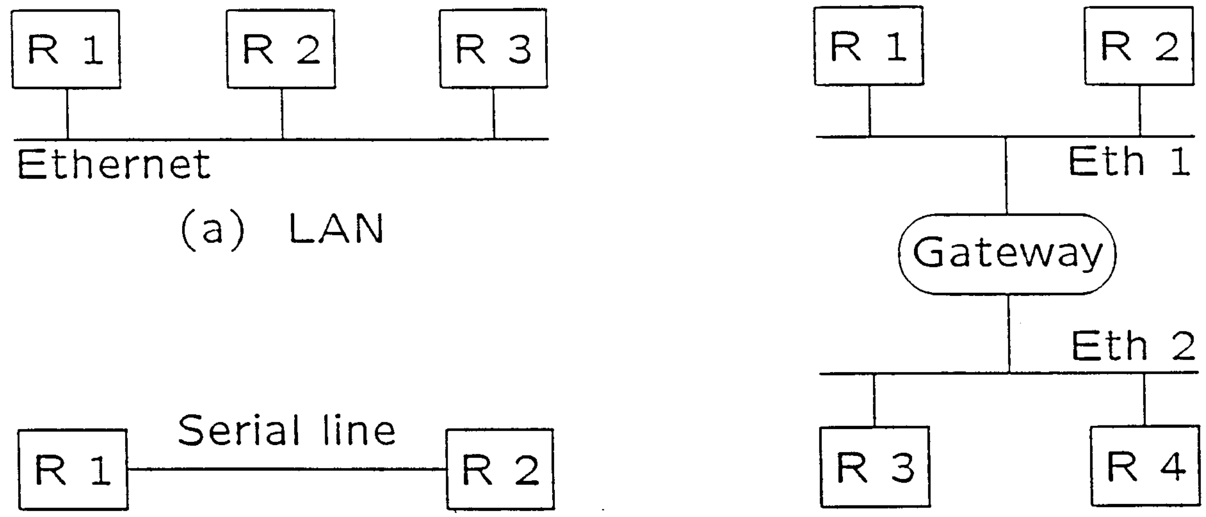
Times reported in seconds.

RAID Concurrency Control CPU Time

server	CC	
transaction	user	sys
select one tuple	0.04	0.06
select eleven tuples	0.02	0.02
insert twenty tuples	0.12	0.13
update one tuple	0.02	0.02

Times reported in seconds.

Network Configurations for Raid



(a) LAN

(b) 19200 bps serial line

(c) Internetwork of LANs

Transaction Execution Time on Different Communication Topologies (in ms)

Transaction	a-1	a-2	b	c
select one tuple	100	100	240	120
insert twenty tuples	320	380	520	400
update one tuple	100	120	260	120

Improvement Emphasis for Raid Communication subsystem Version 2

- Avoid top-heavy communication abstractions

- Reduce kernel interaction

- Minimize message copying

- Use a simple IPC memory management

- Adopt same mechanism for local and remote

- Exploit the nature of DTP

Raid Communication Subsystem V.2

- Port: shared by process and kernel

- Protocol: SE for LAN

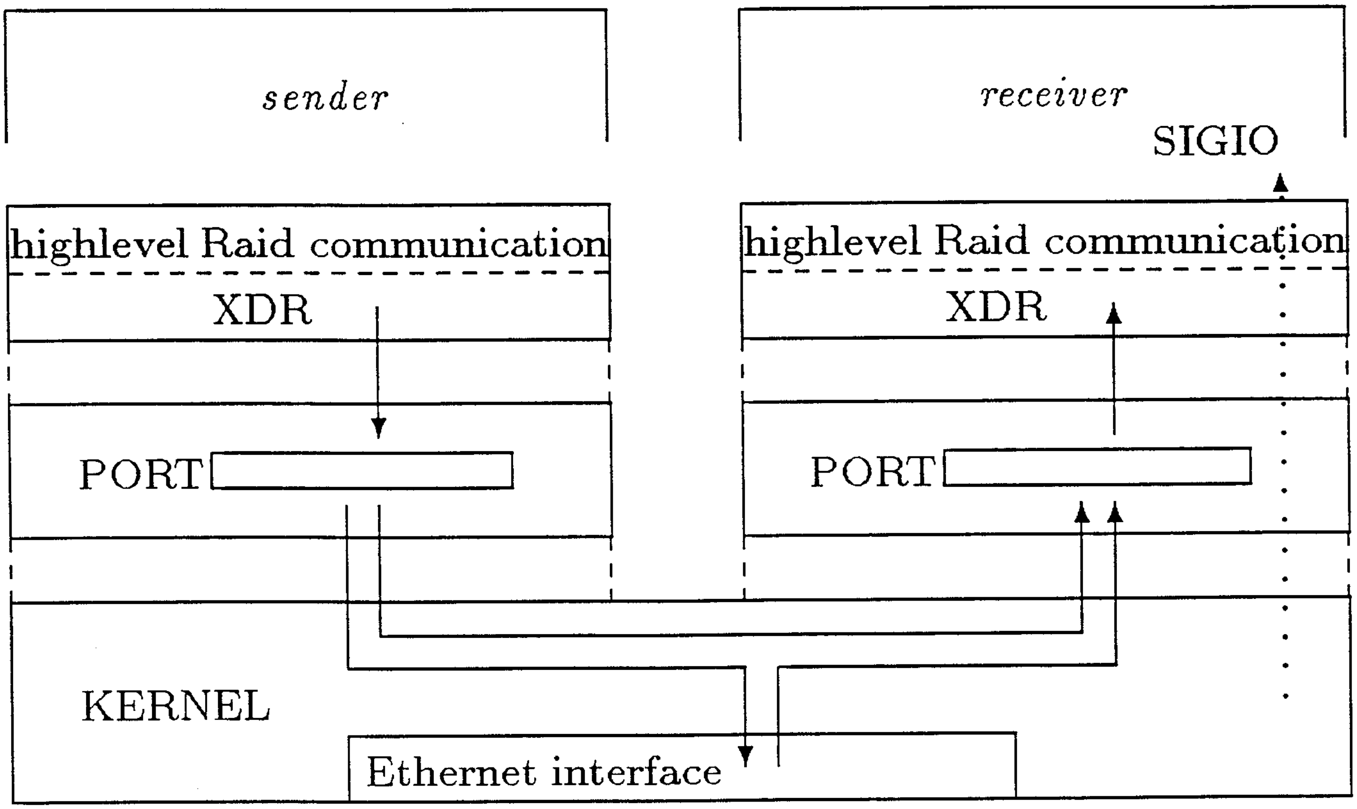
- Naming: simple naming scheme:

  ⟨Raid instance, server, server instance⟩ ⇔ port number

- Multicasting: each site sets a multicasting address using the transaction ID.

- Communication Primitives

Structure of Raidcomm V.2



Structure of a Communication Port

trmlen	Transmission Buffer	Active Buffers
len1	Receive Buffer 1	
len2	Receive Buffer 2	
len3	Receive Buffer 3	
	. . . .	
lenN	Receive Buffer N	

Performance: Communication Primitives

- Goal: to evaluate the performance of Raidcomm Version 2

- Experiments: measure the local and remote round trip times

  - Add socket-based IPC and two SYS V local IPC methods for comparison

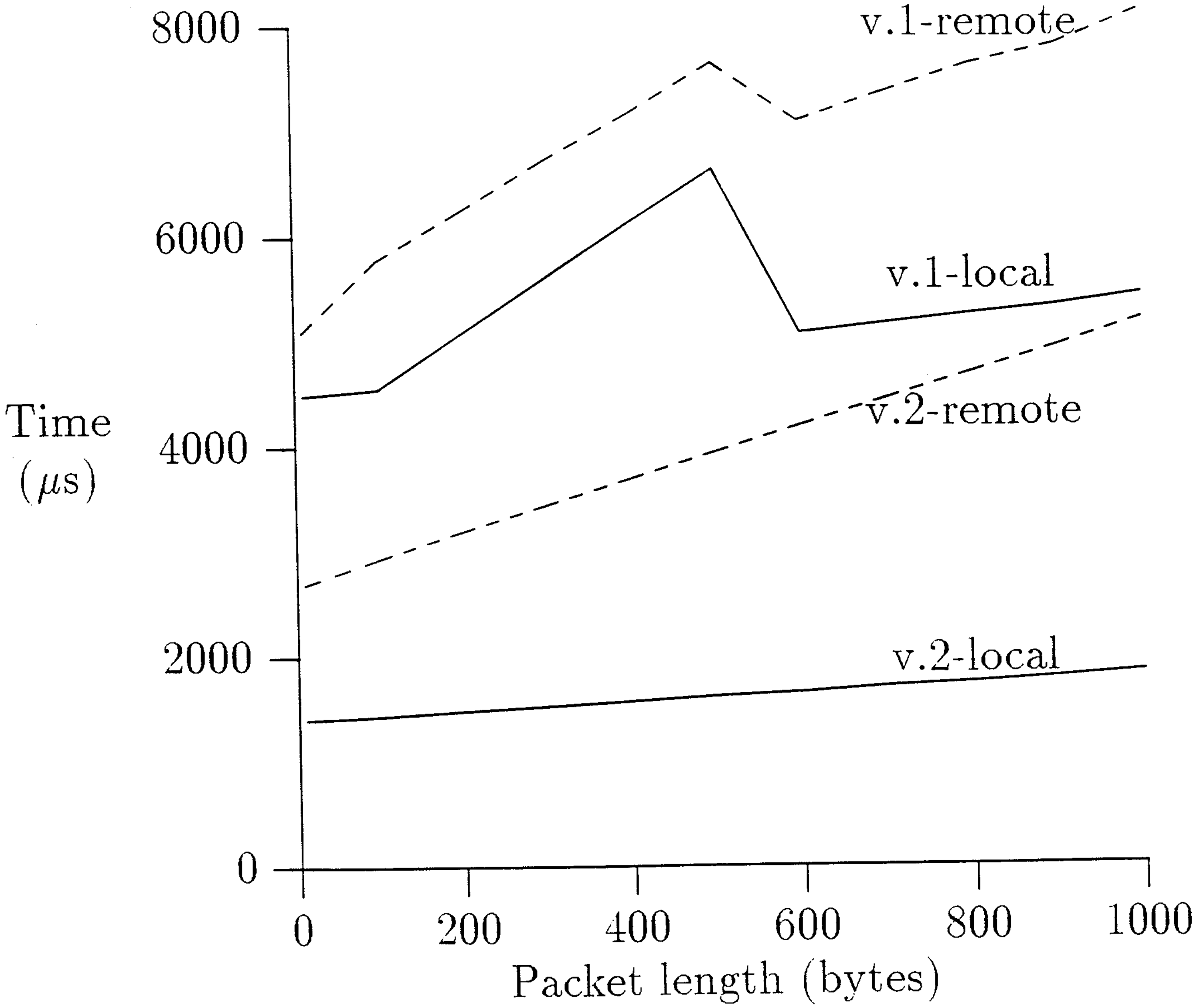
- Conclusions:

  - Our protocol is extremely lightweight

  - Most of the local round trip time is due to context switch overhead

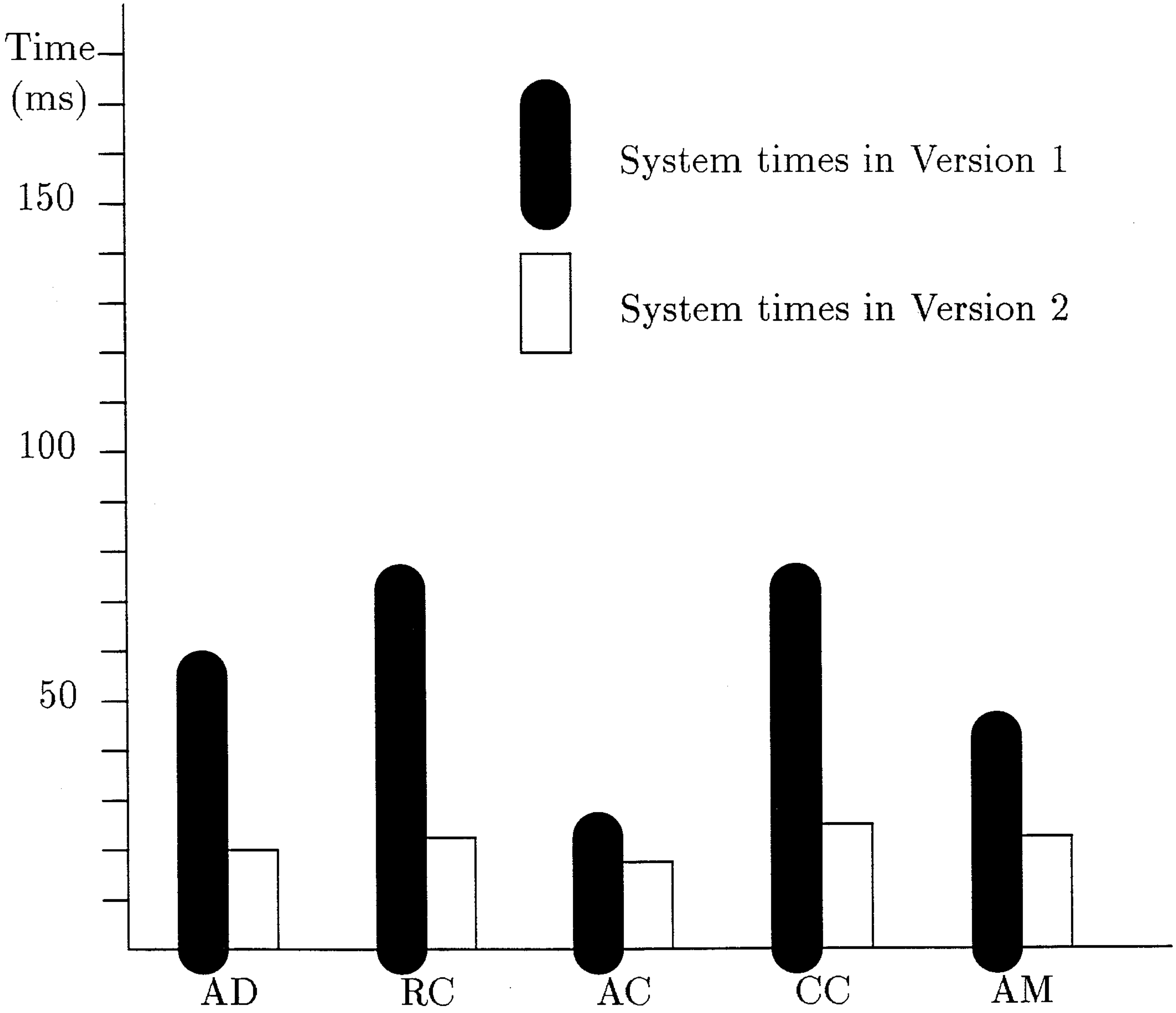
Evaluation of Raidcomm V.2

Round-trip times



Evaluation of Raidcomm V.2

Round-trip times



Problem with XDR

- A data representation standard

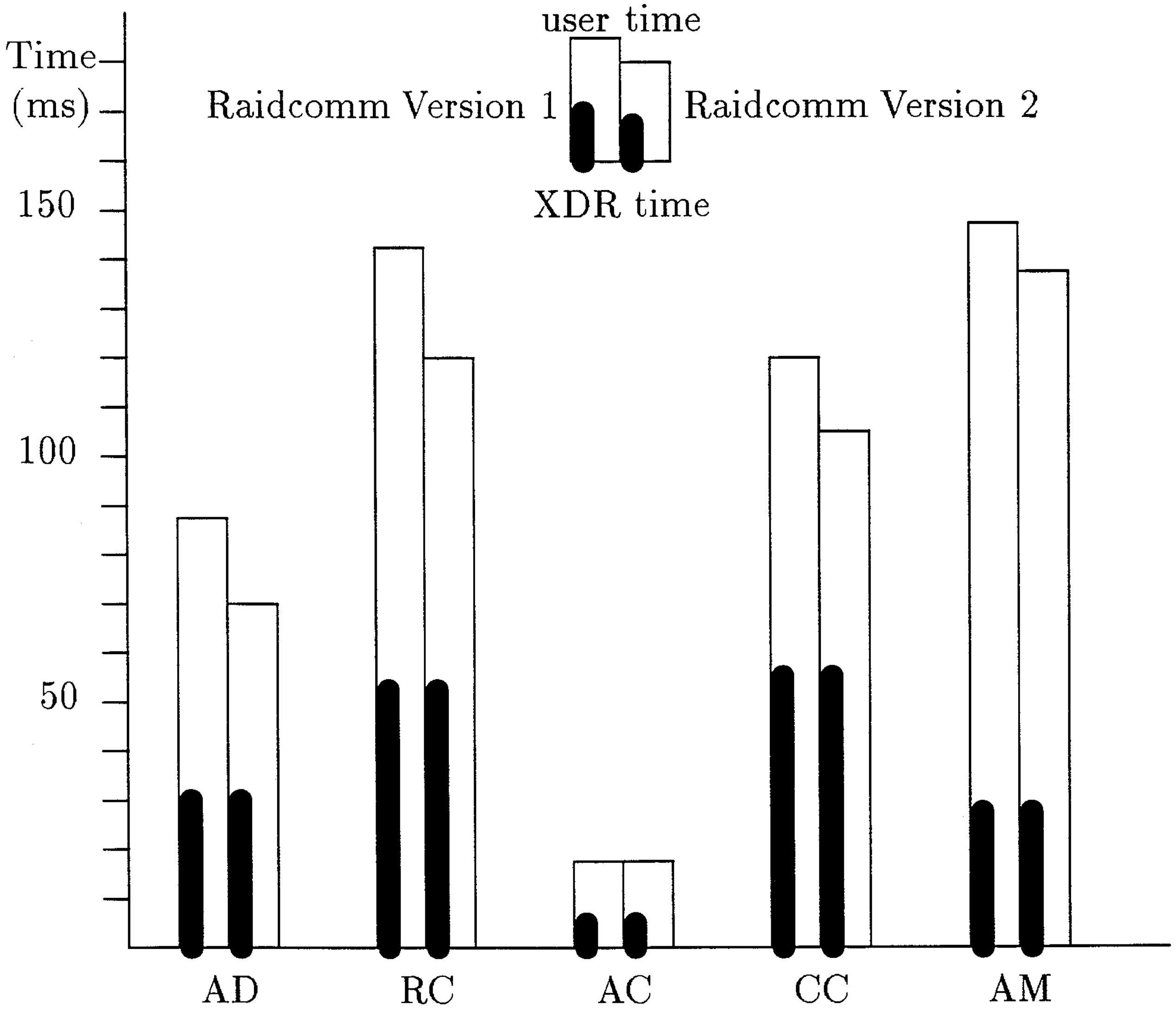
- Doesn't take into account the high-level demands

- Encoding and decoding are expensive

- Unnecessary in most cases

Evaluation of XDR

Average user times for a transaction
in a single machine



Problem with context switch and scheduling

- OS scheduling policies do not consider high level relationships in a group of processes.

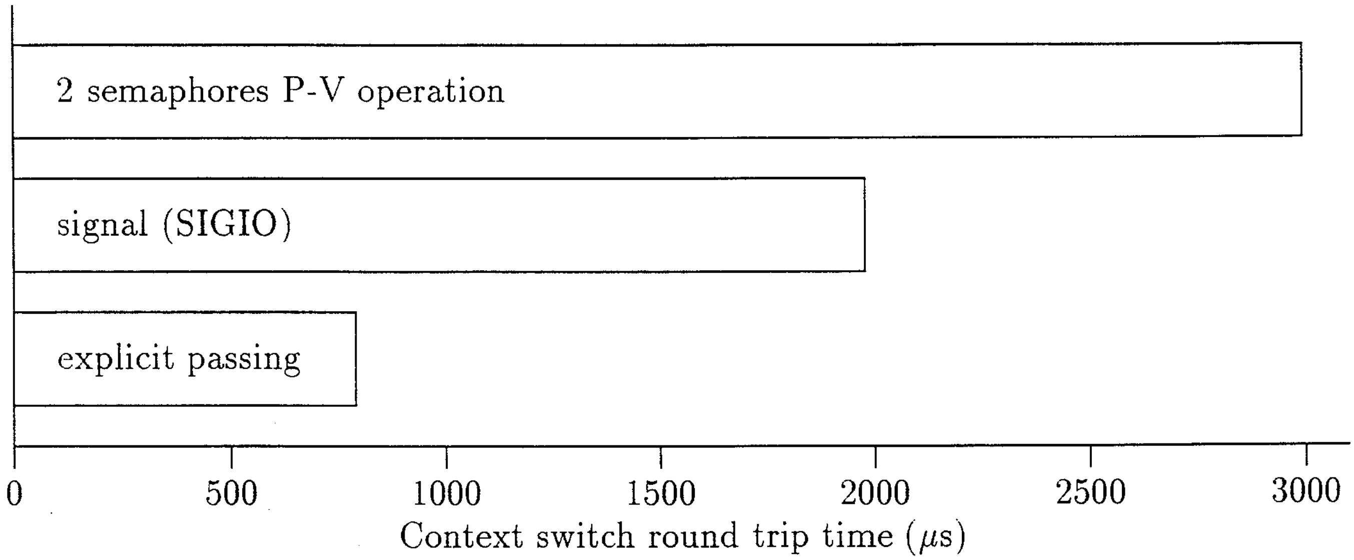
- Raid scenario:

  CC is CPU intensive and its priority is decreasing after some times. This forces CC is give up CPU after processing only one message, even though its time slice has not yet expired.

- context switches caused by synchronization are expensive.

Evaluation of Context Switch

The performance of context switch



Improvement Emphasis for Raid Communication subsystem Version 3

- Avoid unnecessary data formatting
- Provide fast complex data transfer
- Eliminate kernel's involvement in communication activities
- Use shared-memory as communication channel
- Reduce context switch overhead
- Use explicit control passing scheme

Raid Communication Subsystem V.3

- Improve interprocess communication inside one machine
- Port: shared by receiver, sender process and kernel
- Communication channel: pairwise shared memory segment
- Explicit control passing
- Communication Primitives

Evolution of Raid communication subsystems

- Version 1
  - based on UDP only
  - user-level multicasting

- Version 2
  - designed for LAN and local environment
  - use SE protocol and physical multicasting
  - mapping memory between kernel and process

- Version 3
  - improve local communication
  - support efficient complex data objects transfer
  - shared memory between processes
  - explicit control passing

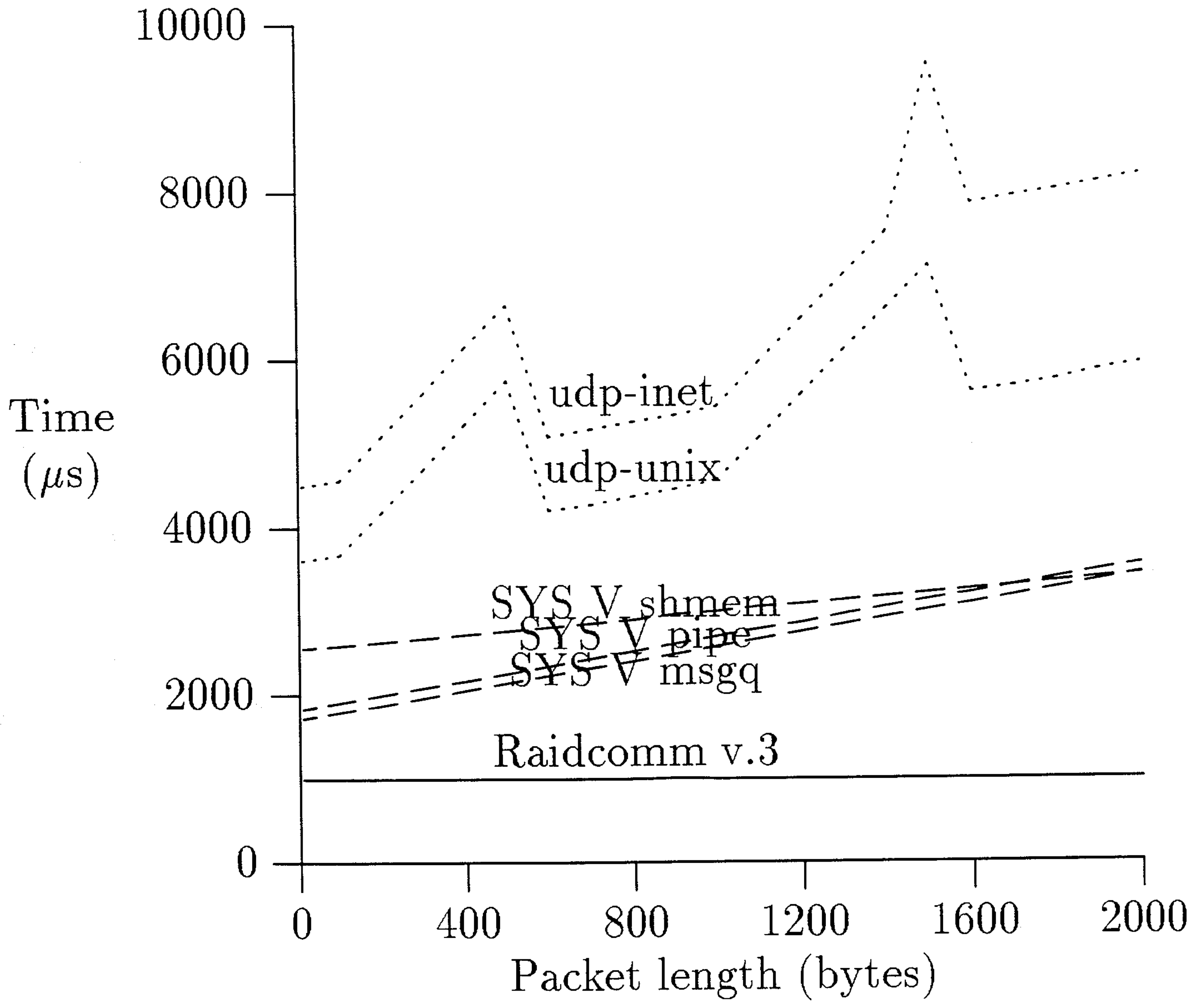
Evaluation

The performance comparison of commucation library

Message	Length	*Raidcomm*		
		V.1	*V.2*	*V.3*
(† multicast dest = 5)	(bytes)	(μs)	(μs)	(μs)
SendNull	44	2462	1113	683
MultiNull †	44	12180	1120	782
SendTimestamp	48	2510	1157	668
SendRelationDescriptor	76	2652	1407	752
MultiRelationDescriptor †	72	12330	1410	849
SendRelation	156	3864	2665	919
SendWriteRelations	160	3930	2718	1102

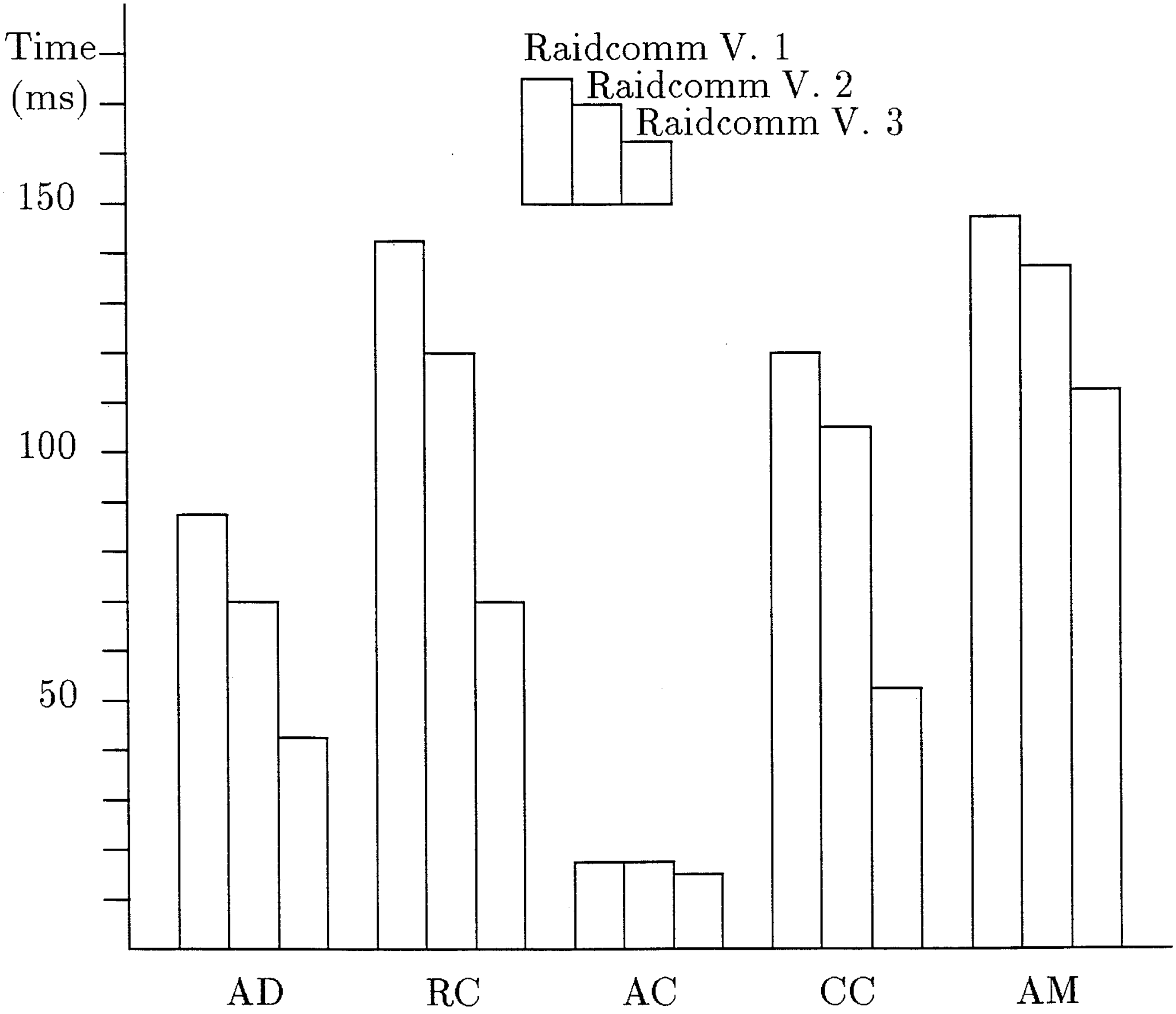
Evaluation

Comparison of message round trip times



Evaluation

User times for Raid servers



WANs Communication for DTP

- Approaches
- Problems
- Solutions
- Our experiences

General Approaches for WANs

- Dedicated links

- Bandwidth reservation (ATM network)

- Reliable transport service (by retransmission)

  - expensive

  - acknowledgement and retransmission may worsen the congestion problem

  - connection-mode is not scalable

Our approaches for WANs

- Clustering
  - A *cluster* is a set of sites that the link between each pair of members are reliable enough to use datagram protocols without error control.
  - Occasionally packet loss is remedied by timeout and rolling back the transaction to the previous checkpoint.
  - It reflects the network characteristics and architecture.
- Two-level communication scheme
- Surveillance

Communication performance of the Internet

- Objective:
  message delivery is essential to DTP.

- Measurements:
  round-trip time, message loss rate.

- Experimental method:
  the Internet echo services: IP/UDP/TCP echo.

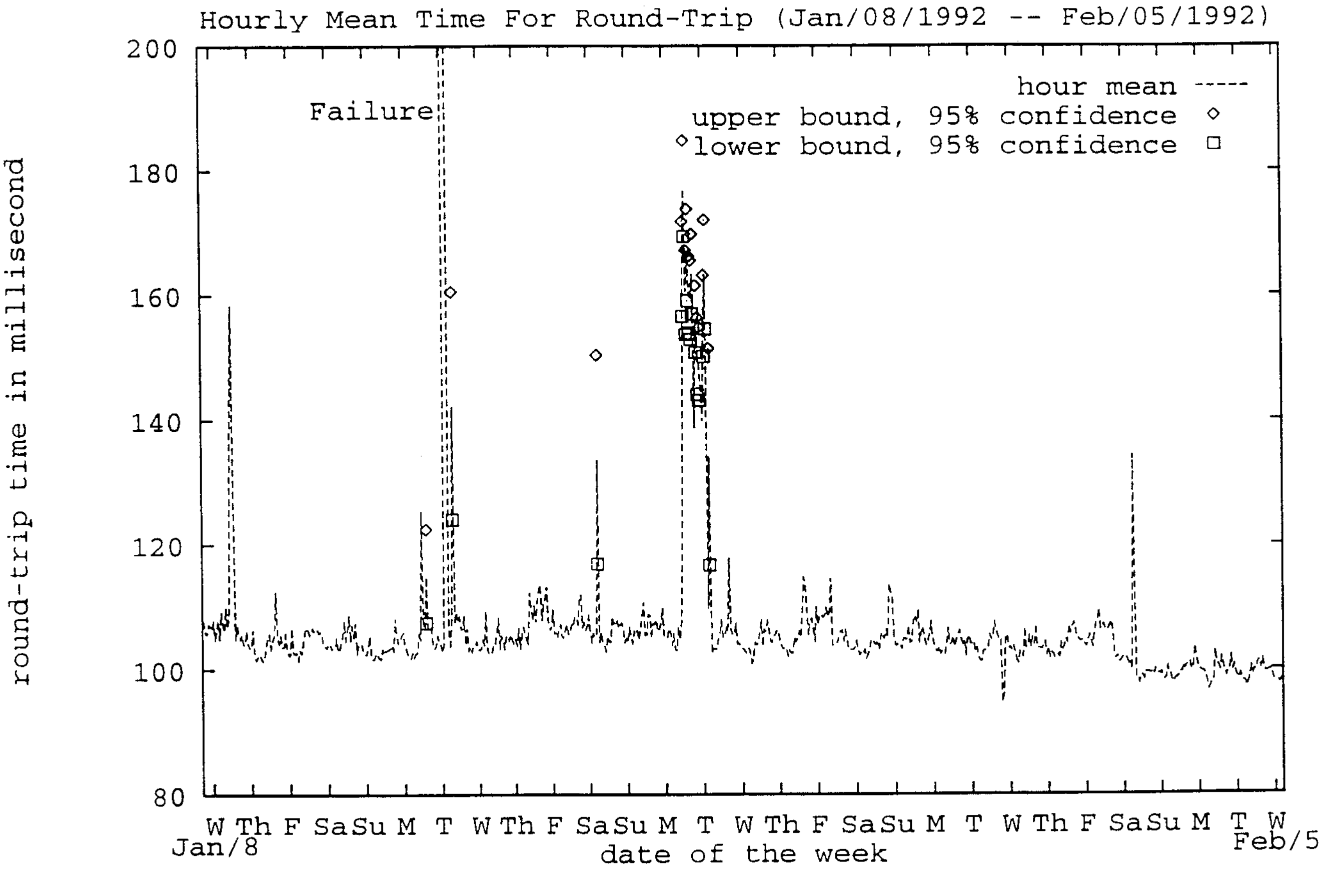
Experiment I: Performance of WAN communication with time as variant

- Procedure:
  - from 1/08/92 to 2/05/92
  - between `raid8.cs.purdue.edu` and `airmics.gatech.edu`
  - one batch of 22 messages in every 5 minutes

- Data:
  MTFRT – Mean time for round trip

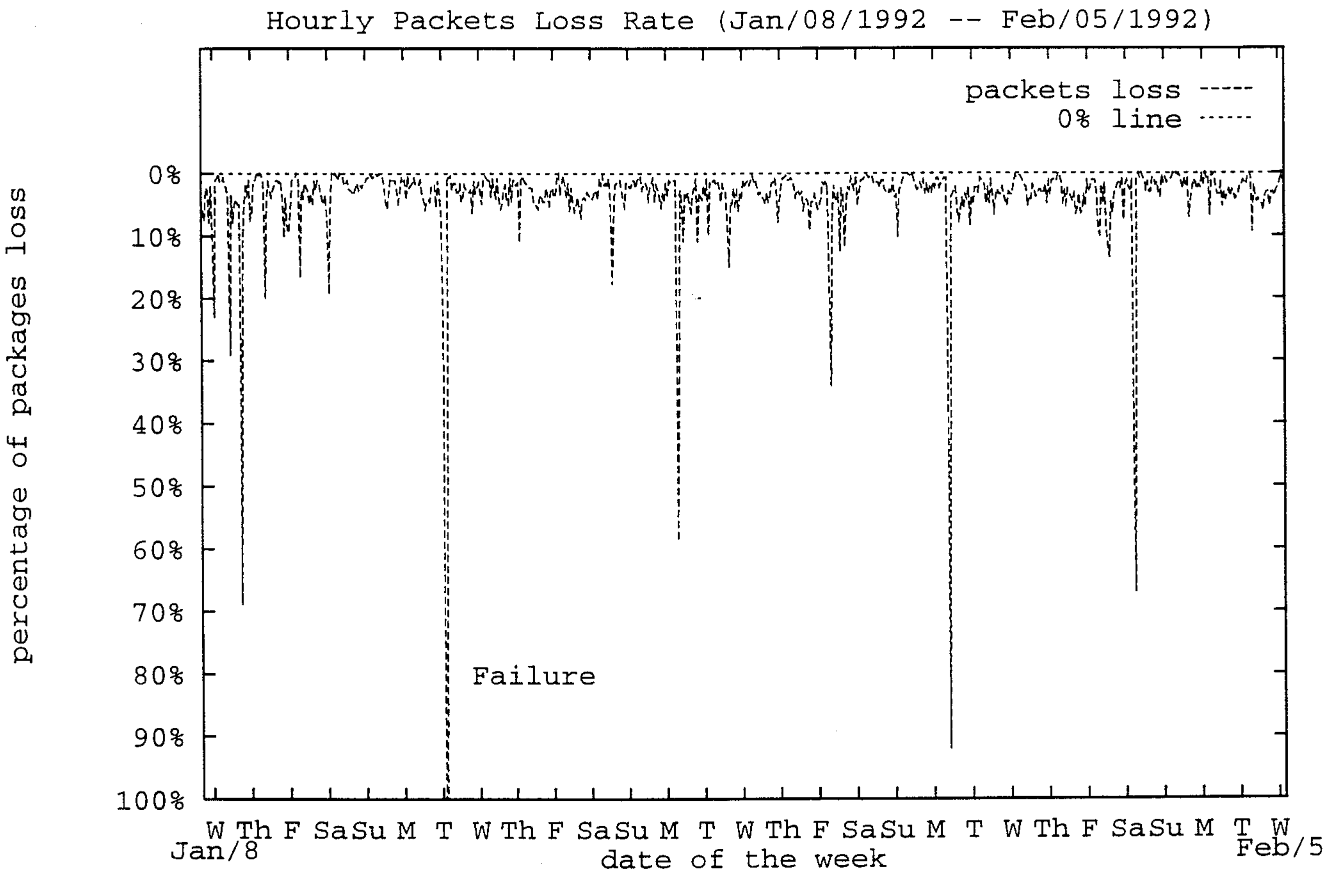
Time Variant

Hourly mean-time for round trip messages



Time Variant *(cont'd)*

Hourly loss rate for messages



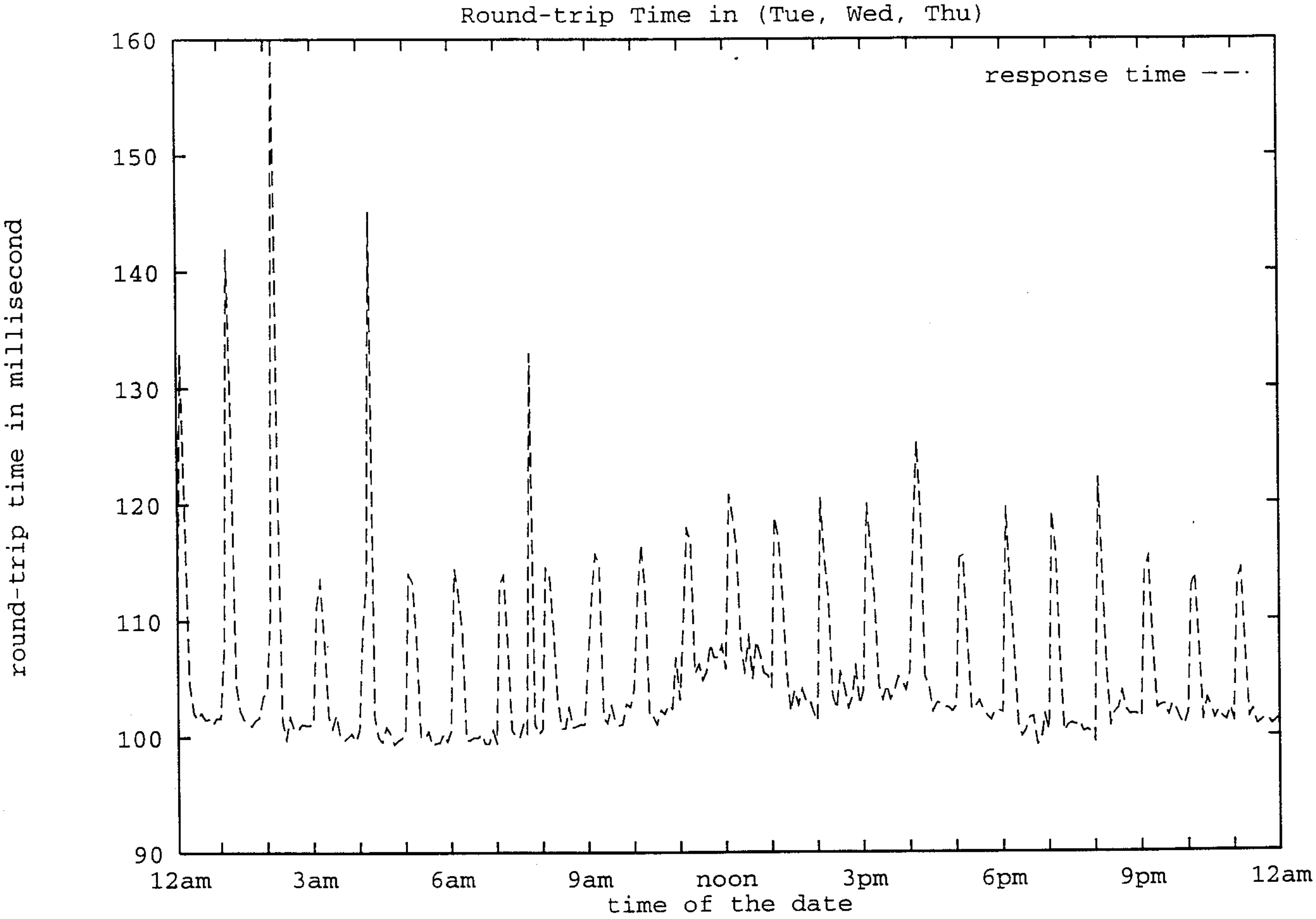
Time Variant *(cont'd)*

Distribution of the hourly data

mean-time	number	pct	average	average
for RT	of hours		variance	loss rate
(ms)			under such	MTFRT
90 – 95	1	(0.1%)	2.08	2.3%
95 – 100	66	(9.8%)	0.74	2.7%
100 – 102	57	(8.5%)	0.94	2.1%
102 – 104	163	(24.3%)	1.02	3.6%
104 – 106	184	(27.4%)	1.27	3.4%
106 – 108	119	(17.7%)	1.60	4.2%
108 – 110	34	(5.1%)	2.40	4.3%
110 – 130	25	(3.7%)	3.97	5.9%
130 – 180	21	(3.1%)	14.52	7.8%
failure	2	(0.3%)	N/A	100.0%
total	672	(100.0%)	1.76	4.0%

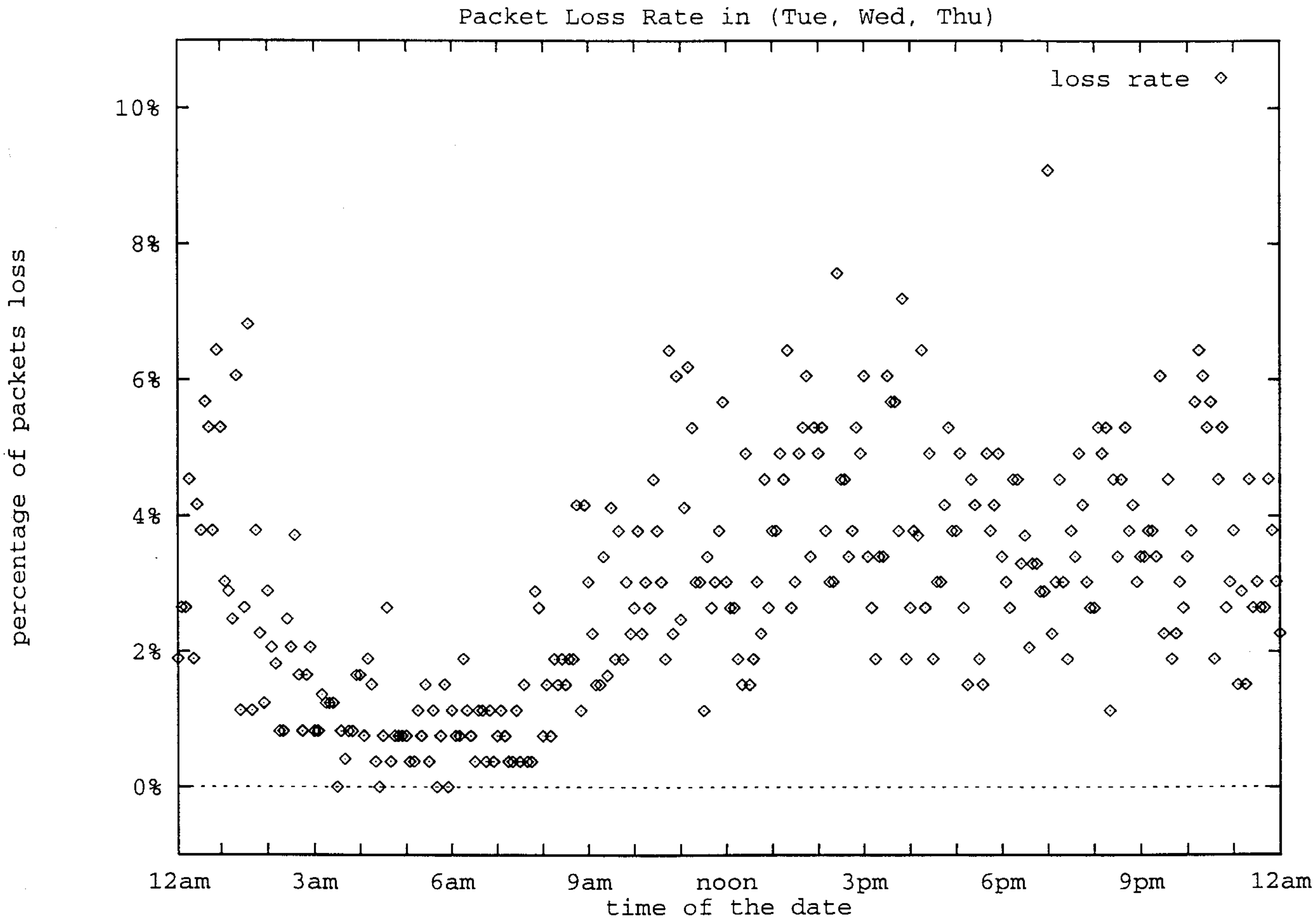
Time Variant *(cont'd)*

Average round-trip time in normal work days



Time Variant *(cont'd)*

Average loss rate in normal work days



Observation for Time Variant

- There are large variations in parameters such as communication delay and messages loss.

- The variations exists in two dimensions: along the time axis and across the networks.

- The time of day has strong influence on the message delivery.
  - The message delay is slightly longer around noon.
  - Hourly peek phenomenon.

Experiment II: Performance of WAN communication with site as variant

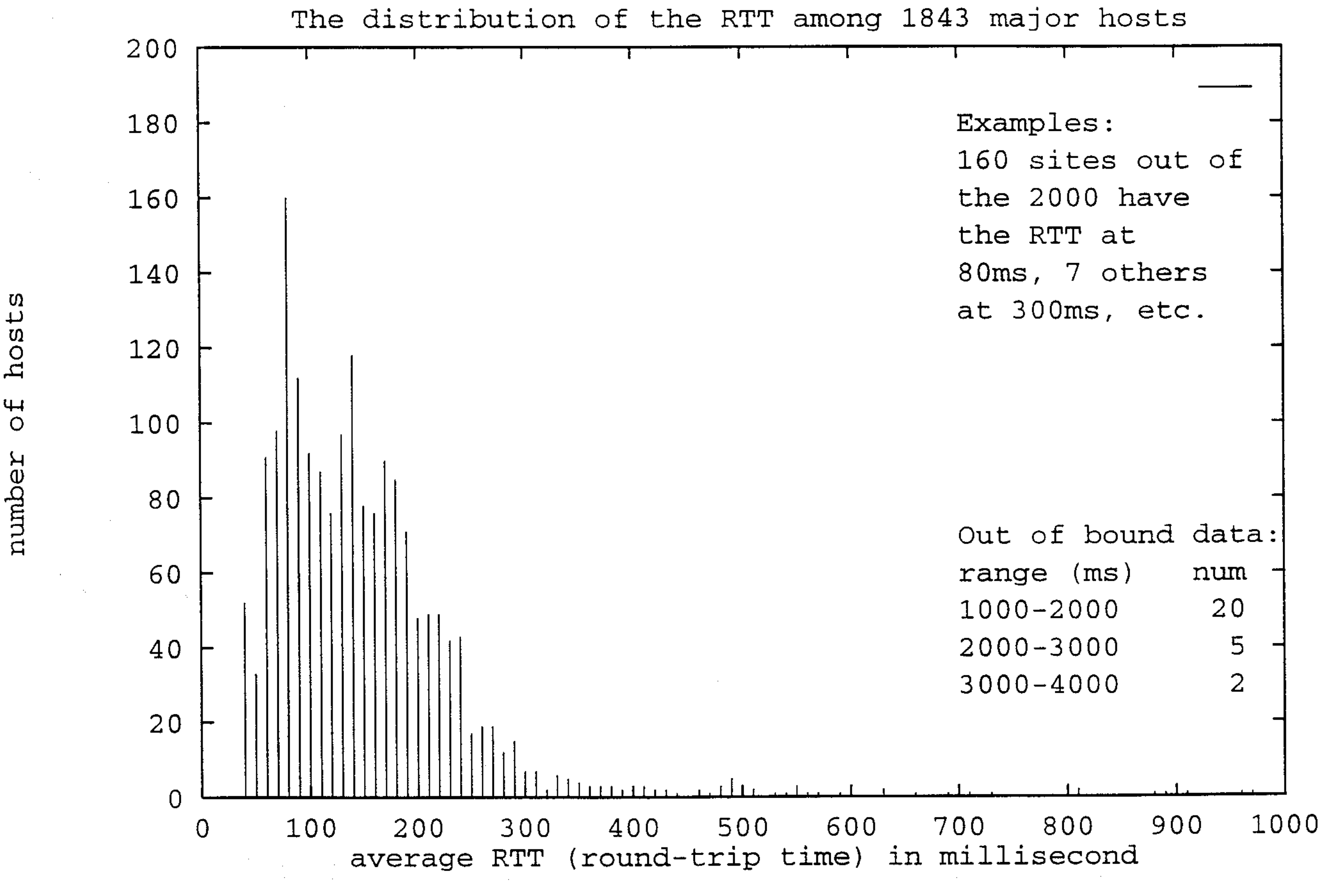
- Procedure:
  - selected over 2000 major hosts from American univeristies and colledges.
  - collected data from 1843 hosts and 505 local networks

- Major hosts used:
  Mostly are backbone host such as mail homes. They are good representative for measurements.

- Measurement: round-trip time

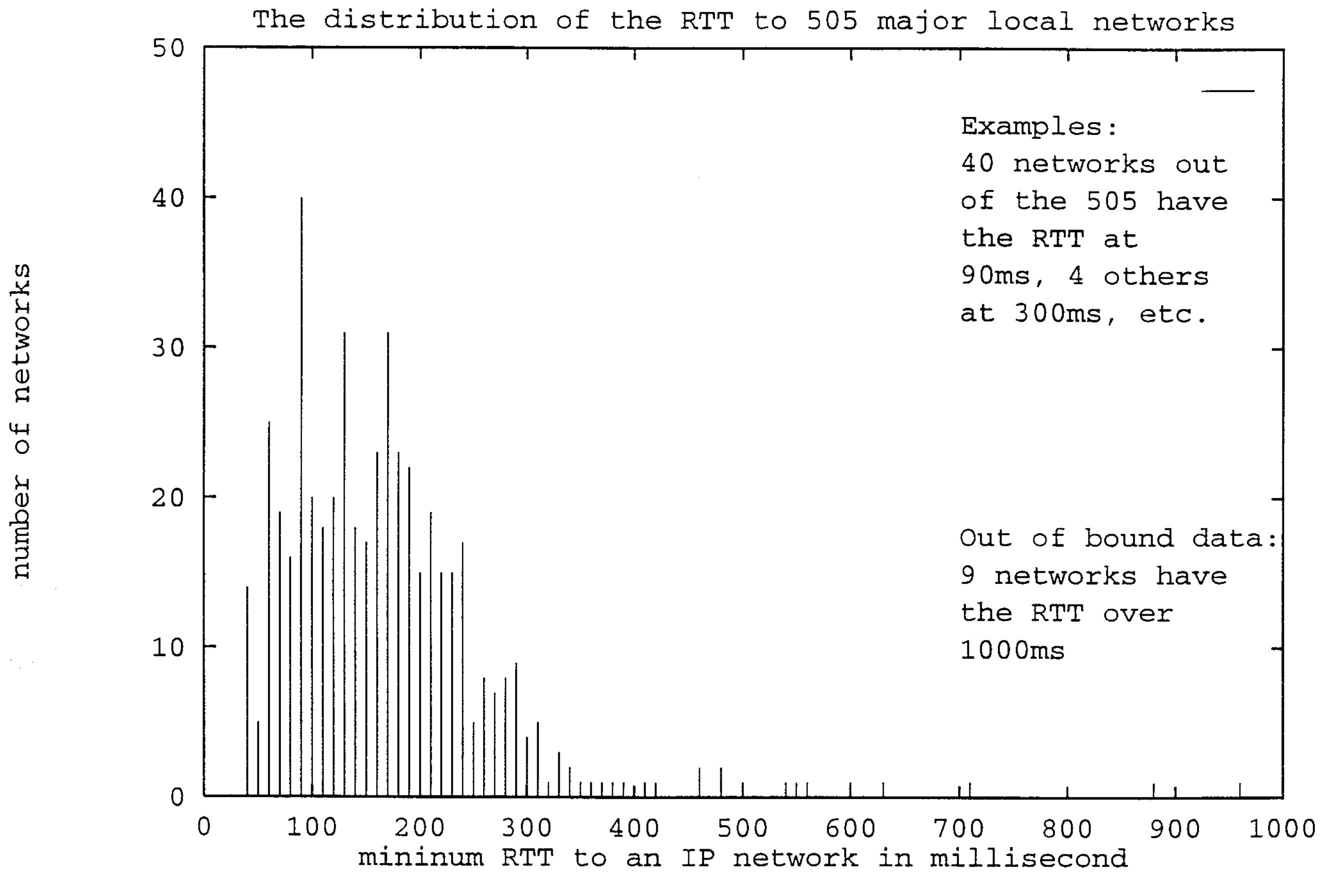
Site Variant

Distribution of round-trip time to a host



Site Variant *(cont'd)*

Distribution of round-trip time to a network



Observation for site variant

- The message delivery performance is unbalanced across the netowrks.

- Most of the major hosts can be reached within 400ms round-trip time.

- There is "clustering" effect in the network topology; different sites in a same local network have similar behaviour in the view point of an outside host.

Impact of WAN communication on DTP

- The time to delivery a transaction message is substantially longer. This implies that a transaction has to stay much longer in the system, which will increase the contention of a database system.

- The timeout mechanism is affected. The timeout value should be a function of the number of remote messages needed for the transaction as well as the messages' destination.

- Many of the local area network solution is not applicable or insignificant to wide area network.

  e.g. physical multicasting

Impact on DTP *(cont'd)*

- The focus on improving communication should be on reducing the number of messages generated by the system.

- More reliable transport service are needed.

- DTP algorithms need to take into account the highly variating in parameters such as message delay and loss.

  e.g. quorum consensus replication method

- Surveillance facilities that constantly monitor the network and collect performance data are helpful to improve the performance of DTP.

Communication Issues in DTP systems

- Reliability vs efficiency
  UDP is suitable for local area networks, but not reliable enough for wide area networks.

- Scalability
  Virtual ciruit type of connection mode protocol (e.g. TCP) has scalability problem.

- Two-level communication scheme
  - clustering sites by the communication performance
  - inter-cluster communication: UDP
  - intra-cluster communication: TCP

Surveillance technique for improving DTP

To export the knowledge about WAN performance to the DTP

knowledge that is relevant

Layers	Metrics of information	
	static	dynamic
Comm.	type of link	message delay throughput loss rate
OS	CPU speed	system load
DTP		response time transaction load

Surveillance *(cont'd)*

Applications That Benefit

- Replication Control
  quorum selection (to select the minimal subset of sites that has the best predicted performance)

- Atomic Control
  commitment (to early abort the transactions that cannot continue due to changes in network condition)

- Adaptable Communication Subsystem
  (to change error recovery strategic according to the network dynamics)

Experimenting DTP in WAN

- Experimental Facility for WAN
  - General aproaches
  - Our approach: emulation
- Performance of DTP in WAN

Experimental Facility for WAN

- General Approach 1: Real experiment
  - small scale, designated sites only
  - spans multiple administrative boundaries, had to control,
  - expensive and sometimes unrealistic

- General Approach 2: Simulation
  - highly scalable
  - need to simplify the model
  - need to justify the input parameters
  - need to collect a substantially large trace

Emulation Approach

- Any 2 hosts in a LAN emulate 2 hosts geographically separated over the world.

- Communication between these 2 hosts goes through the real path of the 2 emulated hosts.

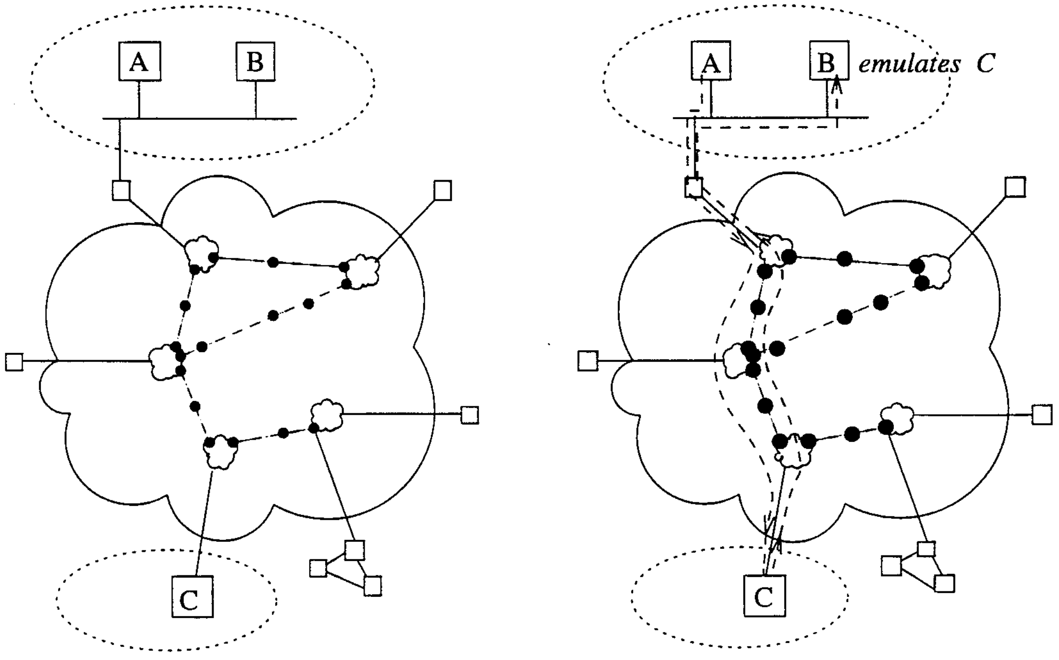
- Justification: what makes the difference between DTP for LAN and for WAN?

  - not the actual location of sites (as long as they are the comparable models),

  - but the communication path.

WANCE tool

Software systems that implement the emulation scheme and allow the emulation experiments of arbitrary sites in the wide area network.

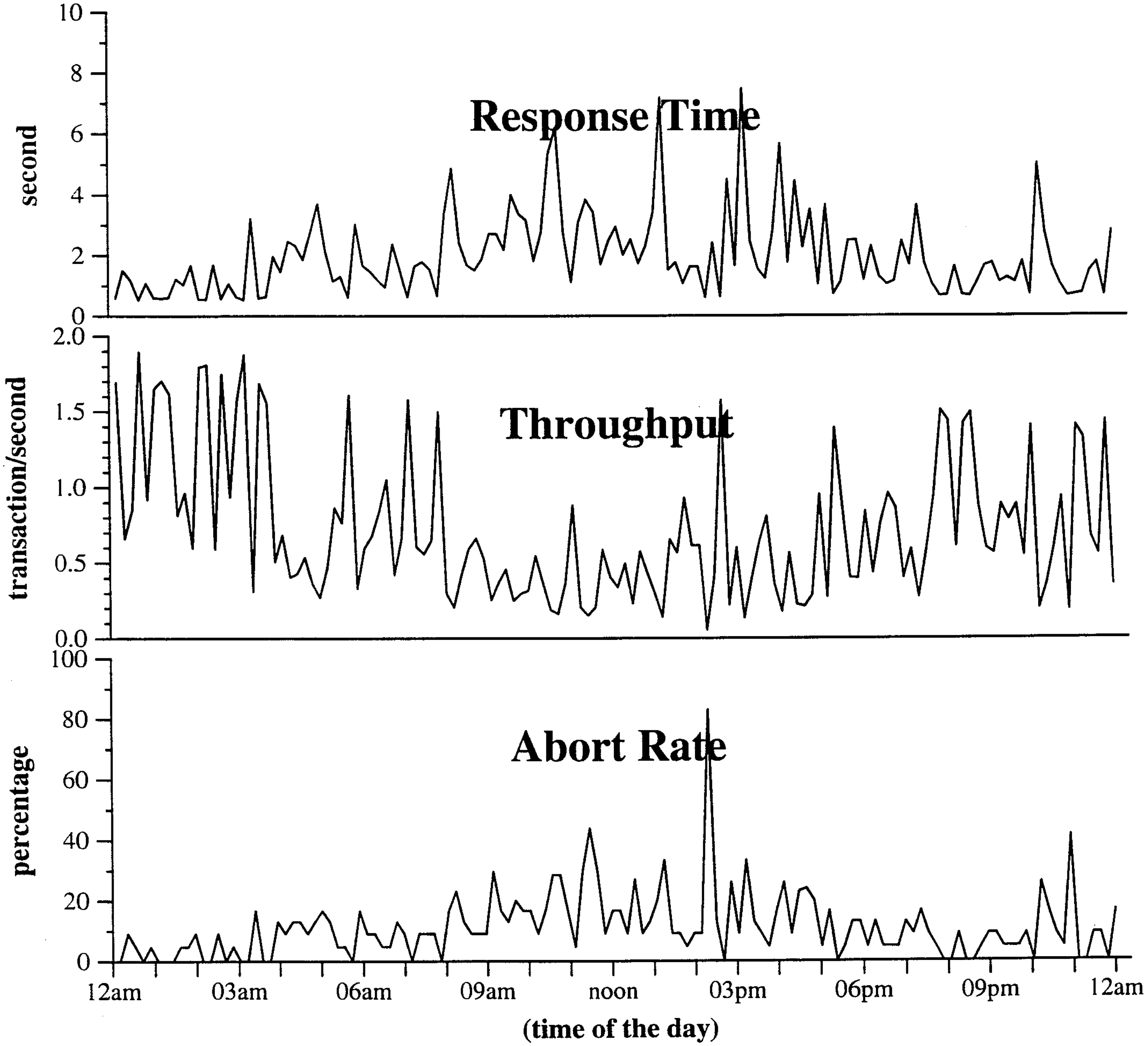


(a) A & B are in a LAN

(b) B emulates C

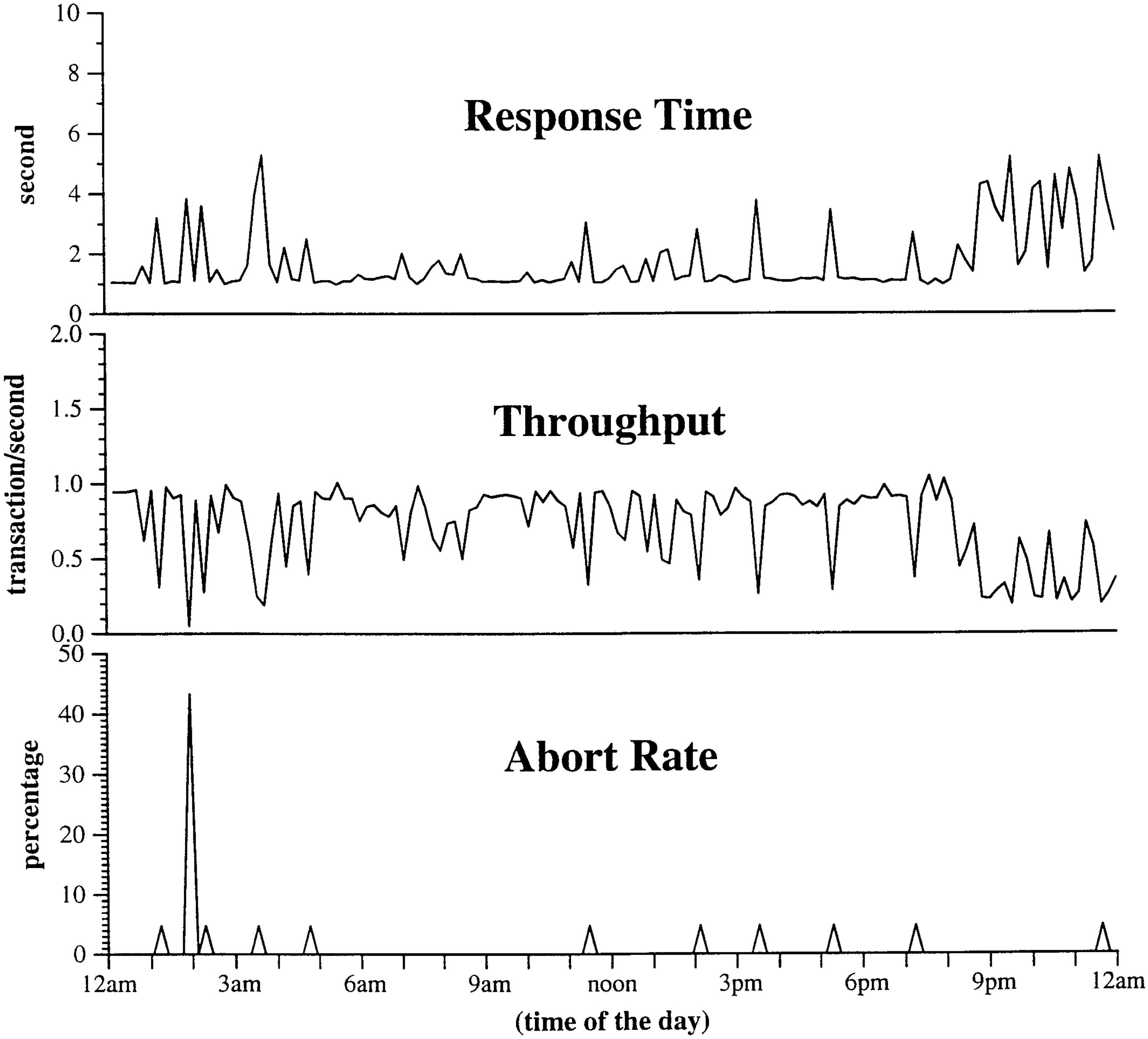
Performance of DTP in WAN

Purdue–Helsinki experiment



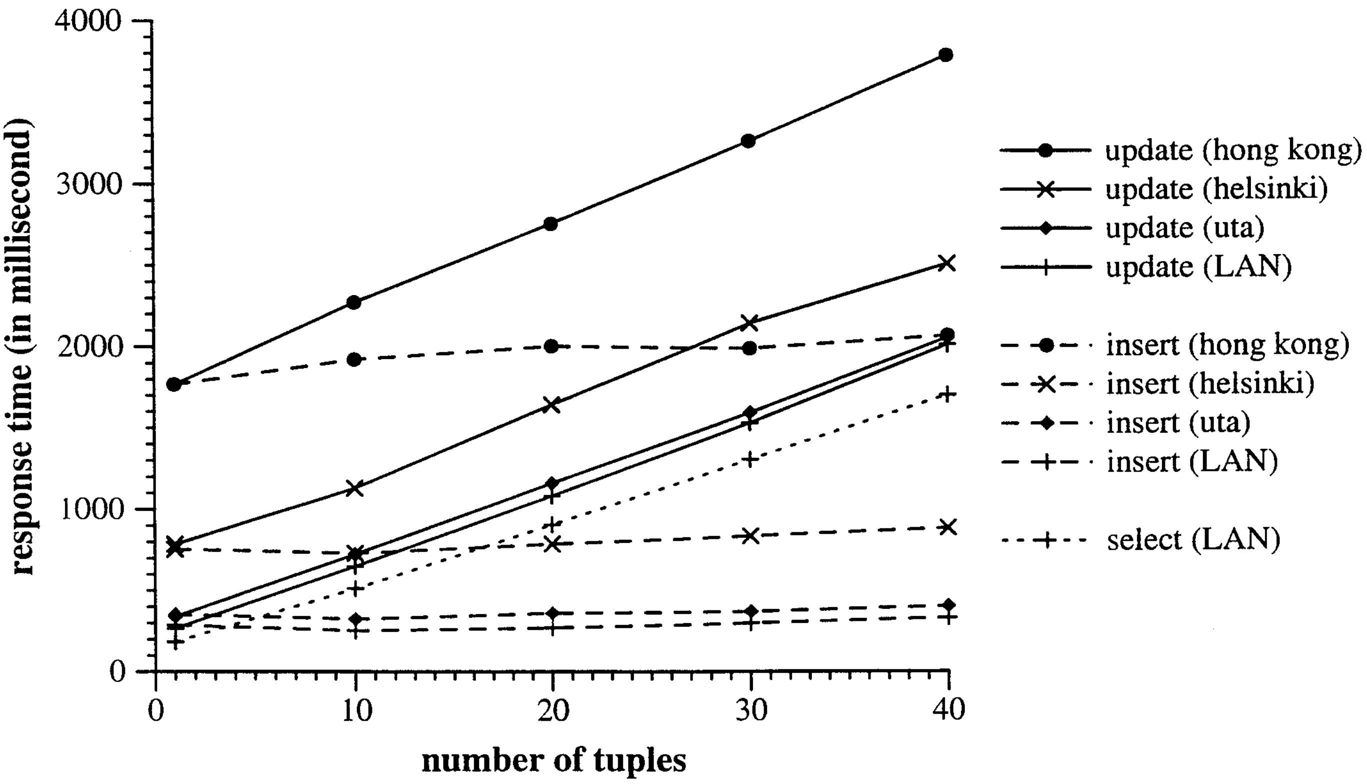
Performance of DTP in WAN

Purdue–Hong Kong experiment



Performance of DTP in WAN

Elapsed time for basic operations



Performance of DTP in WAN

2PC vs 3PC

time (in second)	response time		
configuration	2PC	3PC	overhead
within LAN	0.6	0.7	16%
Purdue–UTA	0.7	0.8	14%
Purdue–Helsinki	1.3	2.0	53%
Purdue–Hong Kong	3.5	5.5	57%

time (in second)	time spent in AC		
configuration	2PC	3PC	overhead
within LAN	0.3	0.4	33%
Purdue–UTA	0.4	0.5	25%
Purdue–Helsinki	0.9	1.6	77%
Purdue–Hong Kong	2.8	4.8	71%

Results and Conclusions

- The basic communication services required to support distributed transaction processing are
  - multicasting
  - remote procedure call (RPC)
  - inexpensive datagram services
  - local interprocess communication (IPC)
- Identified the problem in general purpose IPC
- Experiment and evaluate various communication model through the evolution of Raid communication software

Results of Our Research
(continued)

- Showed how several ideas fulfill the needs of transaction processing
  - lightweight protocol
  - simple naming scheme
  - memory mapping and shared memory
  - physical multicasting
  - explicit control passing
  - increasing adaptability

- Developed transaction-oriented communication facilities
  - Reduces kernel overhead during transaction processing in Raid by upto 70%
  - Reduces user level overhead during communication in Raid by upto 30%

References

1. Bharat Bhargava, Enrique Mafla, John Riedl, and Bradly Sauder. *Implementation and Measurements of an Efficient Communication Facility for Distributed Database Systems.* Proc of the 5th IEEE Data Engineering Conference. Los Angeles, CA, February 1989.

2. Bharat Bhargava and John Riedl. *The Raid distributed database system.* IEEE Transactions on Software Engineering, June 1989.

3. Enrique Mafla. *Efficient Communication Support for Distributed Transaction Processing.* Purdue University PhD Thesis, December 1990

4. Bharat Bhargava, Enrique Mafla, and John Riedl. *Communication in the Raid distributed database system.* Computer Networks and ISDN Systems, April 1991.

5. Enrique Mafla and Bharat Bhargava. *Communication Facilities for Distributed Transaction Processing System.* IEEE Computer, August 1991.

6. Bharat Bhargava, Yongguang Zhang, and Enrique Mafla. *Evolution of Communication System for Distributed Transaction Processing in Raid.* USENIX Journal of Computing Systems, Summer, 1991.

References *(cont'd)*

1. Bharat Bhargava, Enrique Mafla, and John Riedl. *Experimental facility for implementing distributed database services in operating systems.* International Journal of System Integration, 1992 (to appear).

2. Bharat Bhargava and Yongguang Zhang. *A study of distributed transaction processing in wide area networks.* 1993 (submitted for publication)

3. Bharat Bhargava and Yongguang Zhang. *A scalable communication subsystem for wide area networks.* Purdue technical report, 1992.

4. Yongguang Zhang and Bharat Bhargava. *WANCE – a wdie area network communication emulation system.* Purdue technical report, 1993.