Optimistic Parallel Simulation of TCP/IP over ATM networks

M.S. Oral Examination

November 1, 2000

Ming Chong

mchang@ittc.ukans.edu

Agenda

- Introduction
  - parallel simulation
  - ProTEuS
- Georgia Tech. Time Warp (GTW)
- Implementation
- Evaluation
- Conclusion

Introduction

- DARPA's Next Generation Internet Implementation Plan call for simulations of multiprotocol networks with 10,000,000 nodes in year of 2005.
- Conventional sequential simulators such as *BONeS* and *OPNET* lack capabilities.
- Parallel simulation and new modeling framework
  - *GTW*, Georgia Tech Time Warp
  - *Telesim* project, University of Calgary
  - UCLA's *ParSec*, Purdue's *ParaSol*, etc.

Parallel Discrete Event Simulation

- A simulation is partitioned into Logical Processes (LPs).
- LPs are distributed on a shared-memory multiprocessor machine.
- LPs communicate by timestamped message (i.e. event scheduling).
- Synchronization technique is required to ensure that events are processed in the same order as in a single processor simulation.
- Causality error -- LP receives a message with a timestamp earlier than the LP's local clock.

Synchronization
Conservative vs. Optimistic

Conservative approach

- LP advances its local clock ONLY if it could ensure no causality errors
- Parallelism depends on how much an LP can lookahead
- Network simulation -- lookahead available is often too little to exploit parallelism
- Deadlock possible

Optimistic approach: Time Warp

- Causality errors are allowed (I.e. each LP advances without regard to the states of other LPs).
- Mechanism is required to detect and correct causality errors.
- Rollback: Restore simulation state from a previously saved state.
- State-saving to permit Rollback.

Motivation

- Compare the performance of *GTW* to *ProTEuS* on large-scale ATM and TCP/IP networks simulation.
- Focus on
  - Parallelism (i.e. speedup)
  - Scalability with network size
  - Impacts of network characteristics

ProTEuS

- A rack of PCs costs less than a shared-memory multiprocessors machine.
- ProTEuS performs network simulation on a network of PCs and ATM switches.
- Simulation involves real TCP and ATM protocol stack.
- Proportional time distributed system to synchronize distributed simulations.

Georgia Tech Time Warp (GTW)

- Optimistic discrete event simulator developed by PADS group of Georgia Institute of Technology.
- Support small granularity simulation
  - Cell level simulation of ATM network
- GTW runs on shared-memory multiprocessor machines
  - *Sun Enterprise, SGI Origin, KSR*

Logical Process (LP)

- GTW simulation consists of a collection of LPs.
- Mapping of LPs to processors is static.
- Execution of LP is message driven.
- Behavior of LP is governed by 3 functions
  - ***Initialize()***
    - Bind LP to processor, allocate memory
    - initialize state variables, send initial message to trigger simulation at time 0.
  - ***Process-event()***
    - Invoke event handlers upon arrival of an event
    - modify state variables (state-saving), schedule new events
  - ***Wrapup()***
    - Output statistics

State and Checkpointing

- Each LP defines a state vector
- A state vector may include 3 types of state variables distinguished by checkpointing schemes.
  - **Read-only**
    - No checkpointing
  - **Full-copy**
    - Perform state-saving prior to each event processing
  - **Incremental**
    - Perform state-saving only when variables are modified.
- Different checkpointing schemes are designed to reduce state-saving overhead.

Data structures

Each processor maintains 3 important queues

- Message Queue (MsgQ)
  - Hold incoming positive messages.
- Event Queue (EvQ)
  - Hold unprocessed and processed messages.
- Message cancellation queue (CanQ)
  - Hold messages that have been cancelled (I.e. anti-messages, negative messages).

Event queue data structure

The event queue (EvQ) consists of

- Processed event queue
  - Each LP maintains a processed event queue sorted by receive timestamp.
  - Each processed event contains pointers to state vector history, pointers to messages scheduled by this event.
- Unprocessed event queue
  - Each processor maintained a single priority queue of unprocessed events for all LPs mapped to the processor.
  - Eliminate the need to enumerate the next executable LP.

The main scheduler loop

After initialized, each processor enters a loop:

- Messages in MsgQ file into EvQ, one at a time
  - Timestamp(msg) < LP local time ==> Rollback
    - Cancel msg sent by rolled back event
    - Enqueue cancelled msg into CanQ of the processor holding the msg
- Process anti-message in CanQ
  - Anti-messages annihilate their complementary positive messages
  - If positive messages have been processed ==> secondary rollback
- Dequeue an unprocessed event (smallest timestamp) from EvQ, process the event.

Computing GVT

- Global virtual time (GVT)
  - timestamp lower bound of all unprocessed or partially processed messages, and anti-messages.
  - Ensure simulation progress, perform fossil collection.
- Any processor can initiate a GVT computation
- All processors report their local minimum
- Last processor to report computes new GVT
- Fossil collection is performed to reclaim memory

Implementation

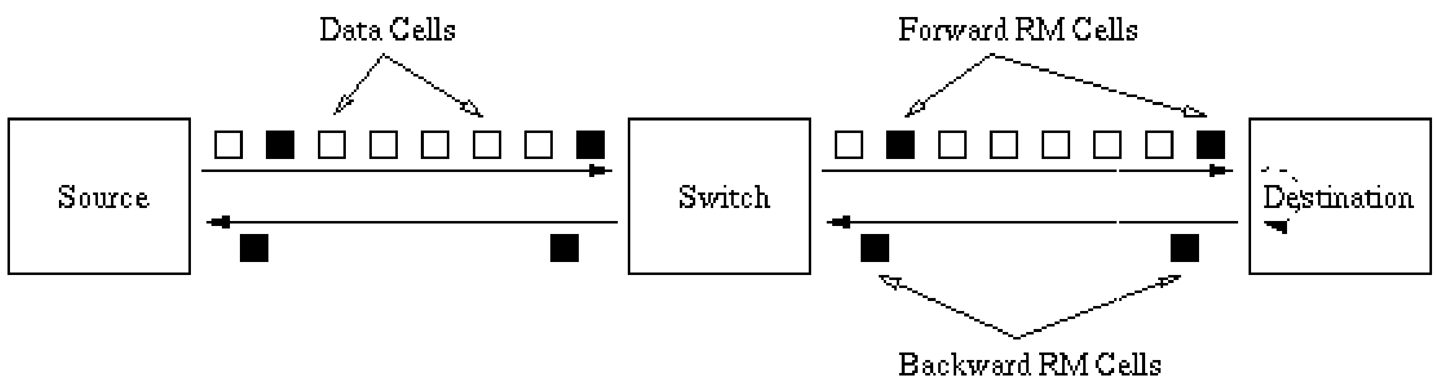
- Simulation models are modularized based on protocol layers:
  - ABR, VBR, TCP sources
  - TCP
  - ATM AAL5
  - ATM network
  - link
- Based on *NIST ATM simulator*
- Consistent with ProTEuS

Implementation: Protocol layers

- TCP source, ABR source
  - greedy
- VBR source
  - cell trace from MPEG clip
- TCP
  - Derived from BSD 4.3 (Reno)
- ATM AAL5
  - segmentation and reassembly
- ATM network layer
  - ATM Forum Traffic Management 4.0

ABR traffic management

- Network provides information on available bandwidth through a feedback system (EPRCA) via *resource management* (RM) cell.



Information and Telecommunication Technology Center

EPRCA

Switch

- Determine load by monitoring queue length
- Compute *fairshare* of the bandwidth for each ABR VC
- Modify CI, NI bits in BRM cells to indicate network congestion, advertise *fairshare* to source via ER.
- Explicit rate (ER) is the max rate allowed to source

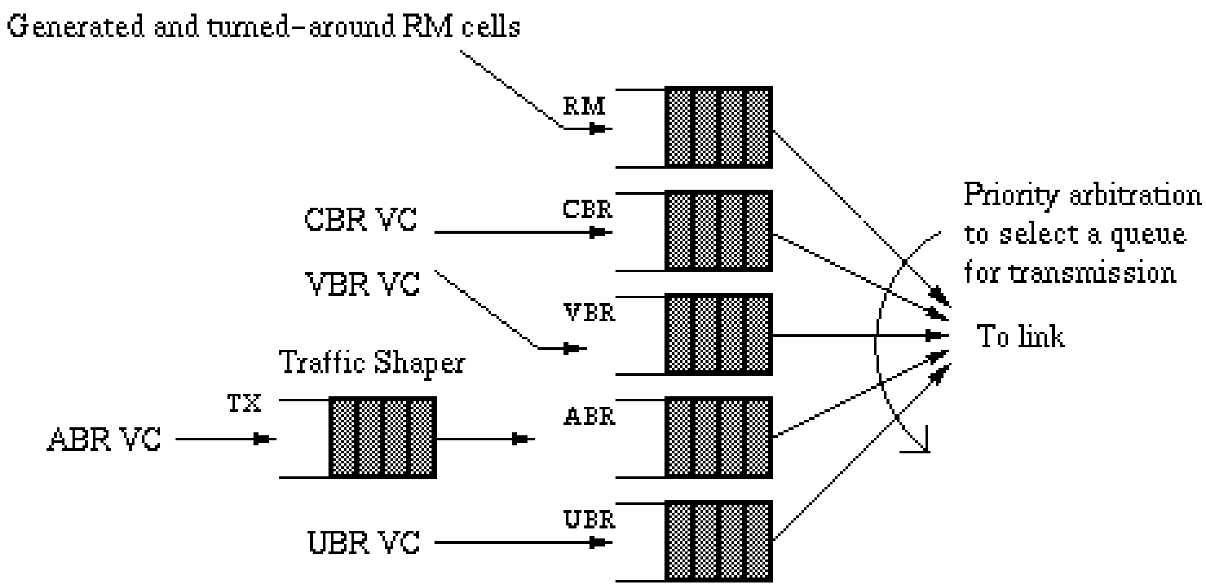
Host

- Compute Allowed cell rate (ACR) based on CI, NI, ER

CI	NI	New ACR
0	0	$\mathrm{MIN}(ER, ACR + PCR \times RIF, PCR)$
0	1	$\mathrm{MIN}(ER, ACR)$
1	X	$\mathrm{MIN}(PCR, ACR - ACR \times RDF)$

Queuing Discipline

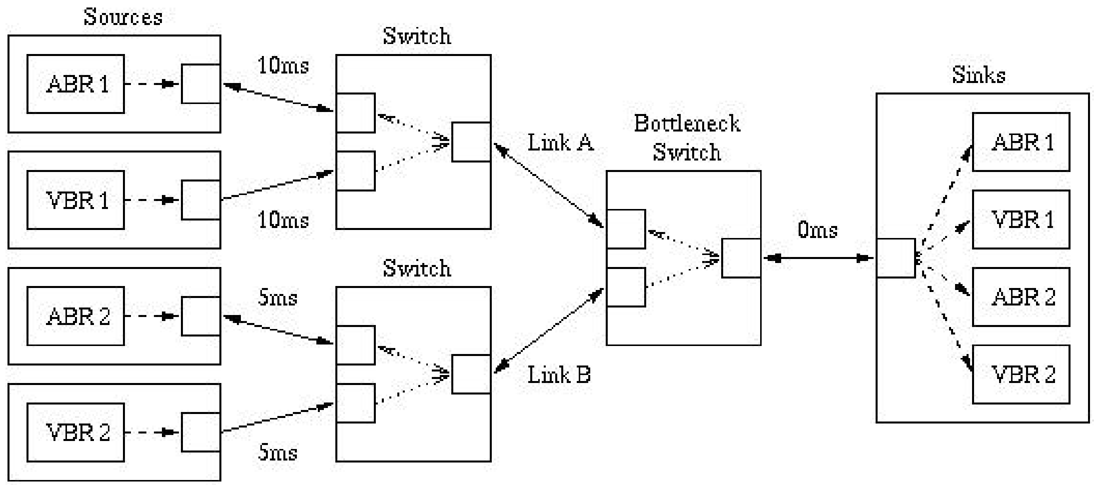
- Per-Class queuing
- Priority order on traffic classes: RM, CBR, VBR, ABR, UBR
- Cell-level traffic shaping on ABR VCs.



Evaluation

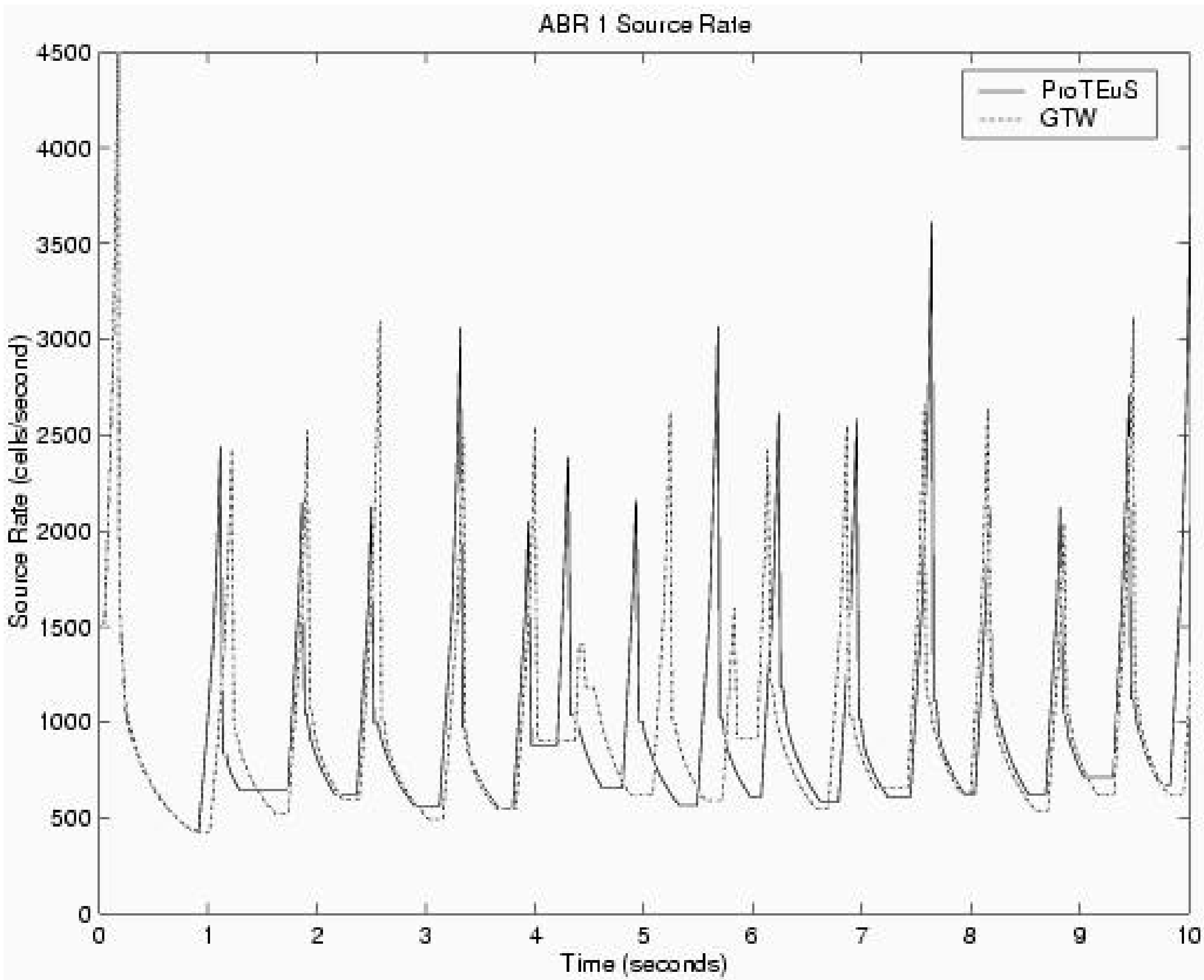
- Evaluate performance of GTW, compare to ProTEuS
  - Speedup
  - Scalability
  - Network characteristics, simulation parameters
- Hardware -- *Clipper* located at LBNL
  - Sun Enterprise server
  - 8 CPU (168 MHz)
  - 1 GBytes physical memory

Validation of GTW models

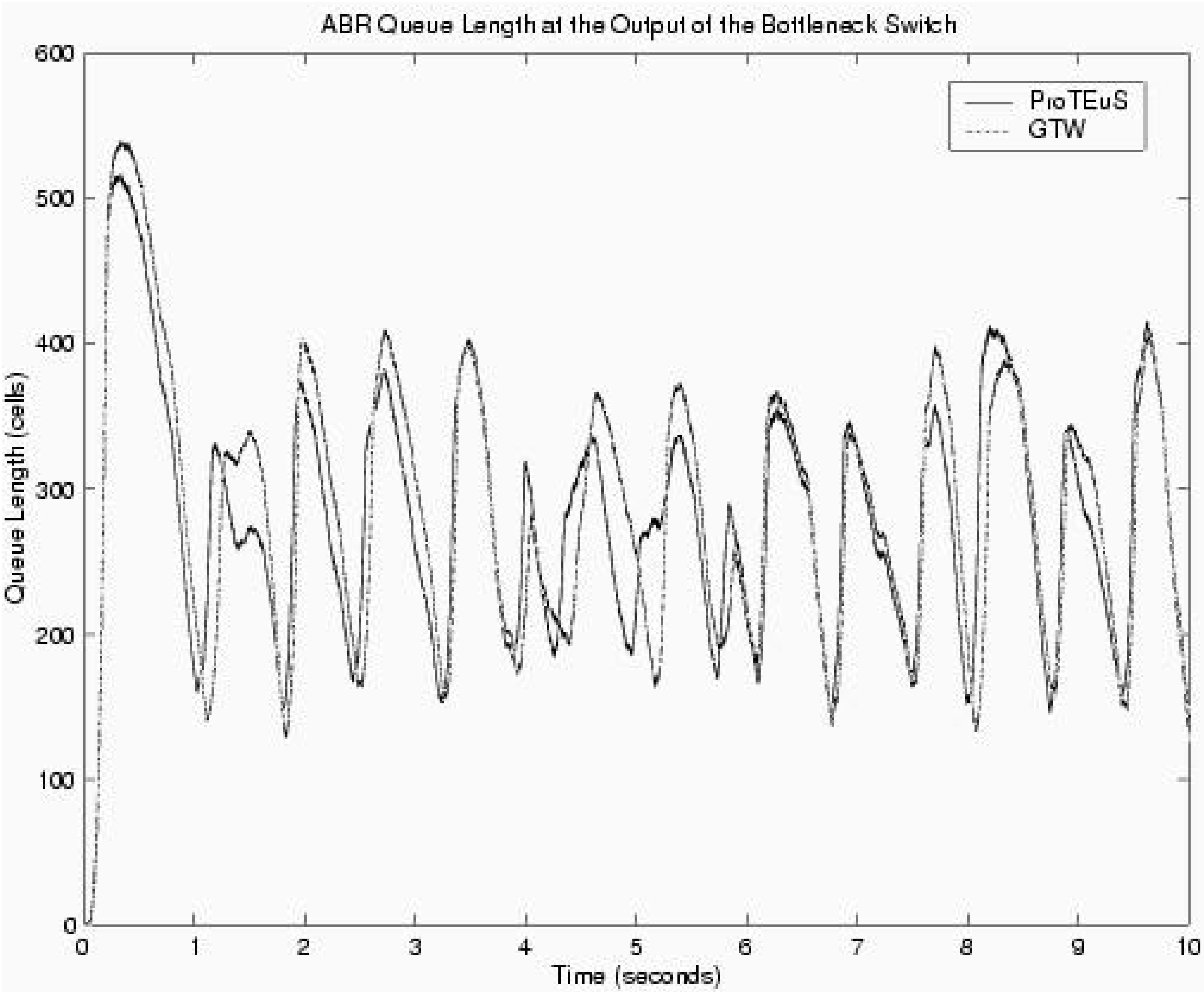


• Line rate	8000 cps
• ABR sources	Greedy (PCR=8000 cps, ICR=1000 cps)
• VBR sources	Bursty (MPEG clip, avg rate = 3000 cps)
• EPRCA threshold	(Low, High) = (200, 300) cells
• Simulated time	50 seconds

ABR source rate



ABR queue length



Link utilization

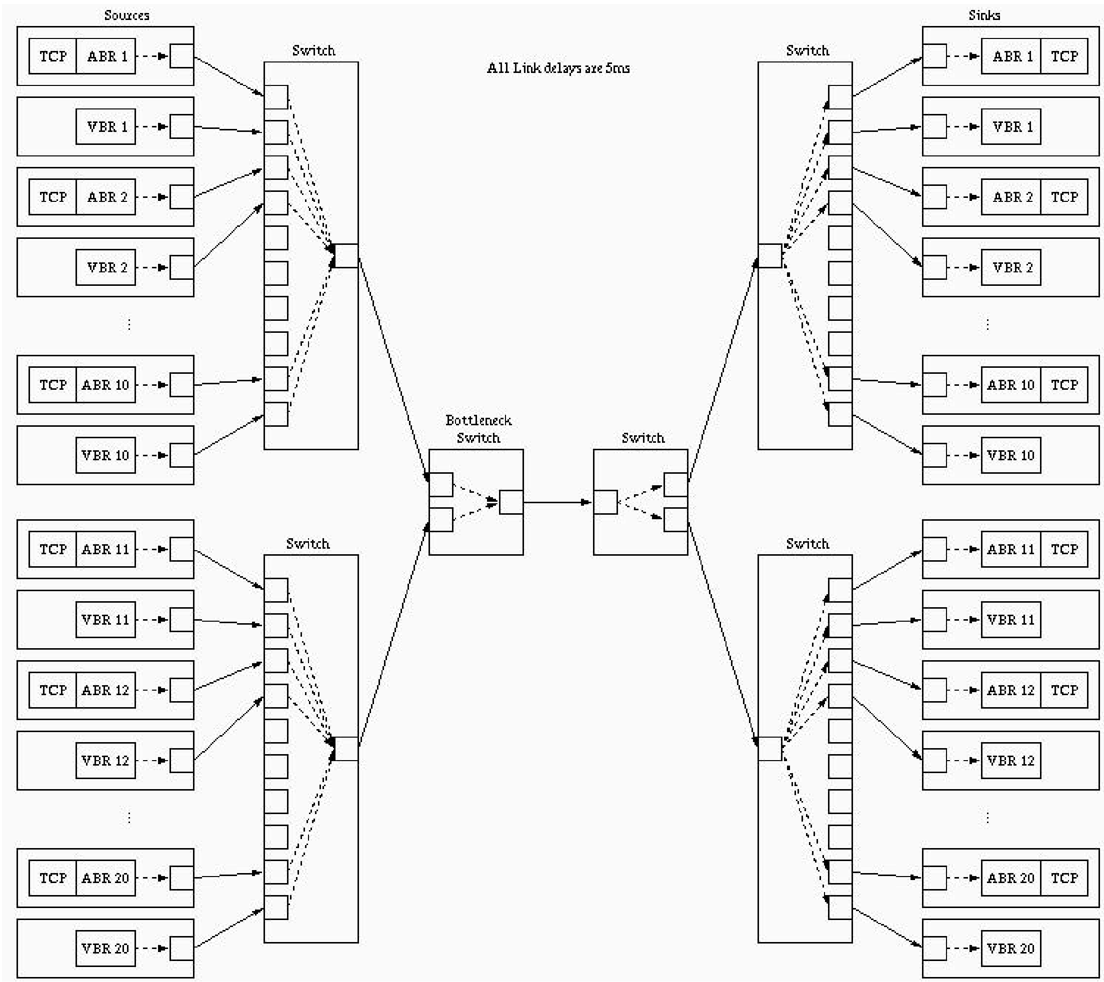
Experiment	Link A		Link B	
	GTW	ProTEuS	GTW	ProTEuS
A:5ms B:20ms	0.502	0.503	0.498	0.497
A:15ms B:15ms	0.498	0.499	0.502	0.501
A:20ms B:5ms	0.498	0.499	0.502	0.501

Mean queuing delay

Experiment	ABR 1 queuing delay (sec)		ABR 2 queuing delay (sec)	
	GTW	ProTEuS	GTW	ProTEuS
A:5ms B:20ms	0.159	0.156	0.164	0.163
A:15ms B:15ms	0.165	0.163	0.161	0.160
A:20ms B:5ms	0.167	0.165	0.159	0.157

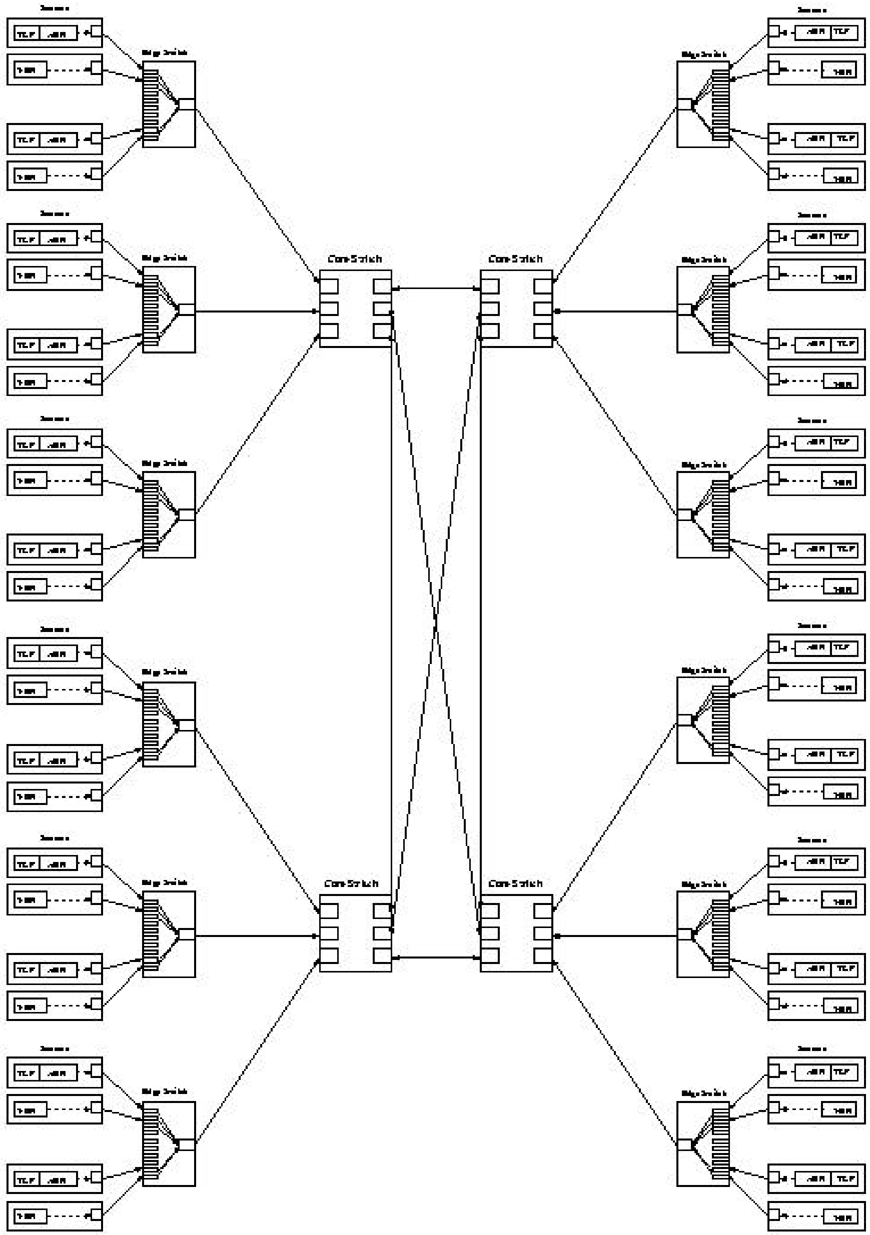
GTW performance evaluation

Scenario A: 6 ATM switches, 40 hosts



- Link: OC-3
- link delay: 5 ms

ABR sources	Greedy PCR = 21000 cps ICR = 25% PCR MCR = 0 cps
VBR sources	50% square wave period = 100 ms MAX = 15000 cps MIN = 10000 cps
TCP source	Greedy
TCP layer	Window size = 512 KBytes TCP Processing time = 1 ms



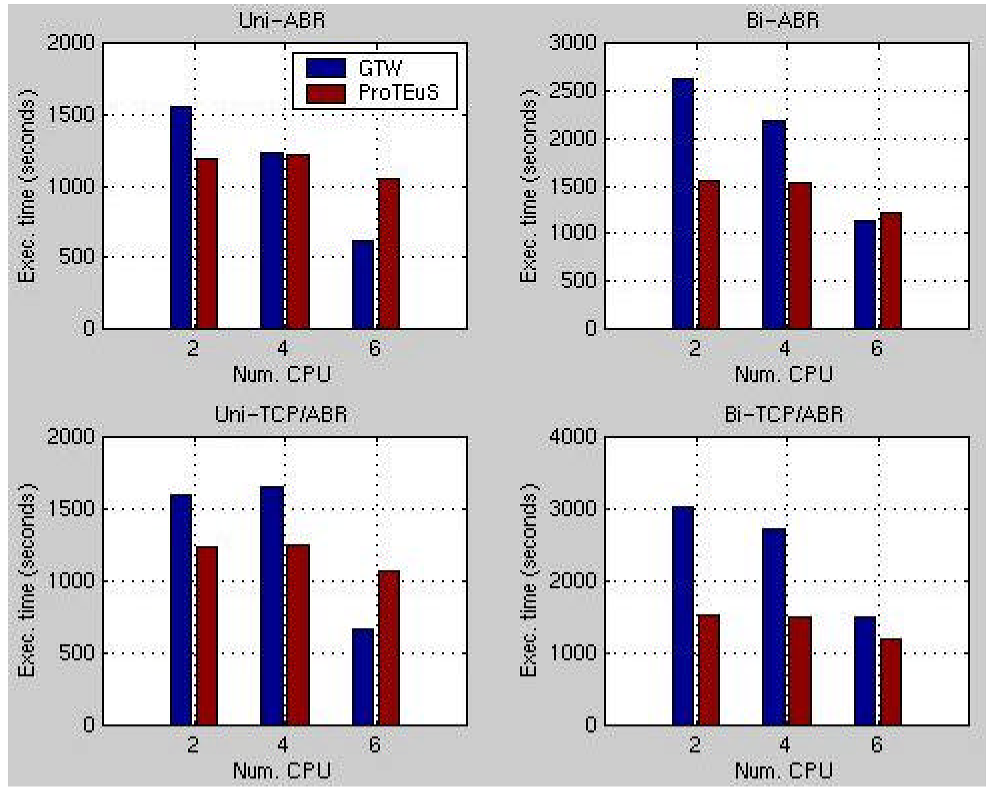
Scenario B

- 16 ATM switches, 120 Hosts
- OC-3 link
- 5 ms link delay

ABR sources	Greedy PCR = 36000 cps ICR = 25% PCR MCR = 0 cps
VBR sources	50% square wave period = 100ms MAX = 36000 cps MIN = 10000 cps
TCP source	Greedy
TCP layer	Window size = 128 KBytes TCP Processing time = 1 ms

Results: Scenario A

Experiment	# Processors	Execution Time (seconds)	
		GTW	ProTEuS
Uni-directional	2	1551.75	1191.32
Traffic	4	1228.38	1213.28
ABR only	6	610.33	1055.88
Bi-directional	2	2622.97	1548.12
Traffic	4	2177.81	1540.79
ABR only	6	1134.29	1221.99
Uni-directional	2	1600.48	1234.22
Traffic	4	1649.88	1243.12
TCP over ABR	6	663.93	1070.77
Bi-directional	2	3016.11	1540.10
Traffic	4	2730.70	1502.08
TCP over ABR	6	1488.28	1200.08

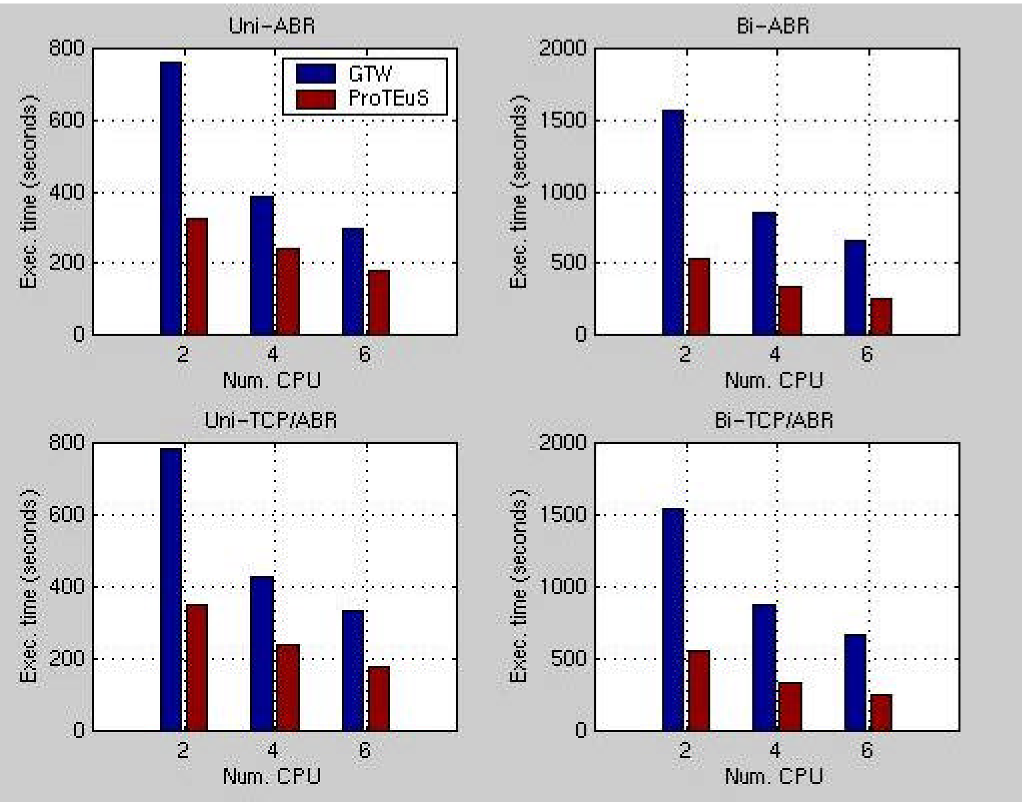


Observations

- ProTEuS scales better
- GTW exploits more parallelism

Results: Scenario B

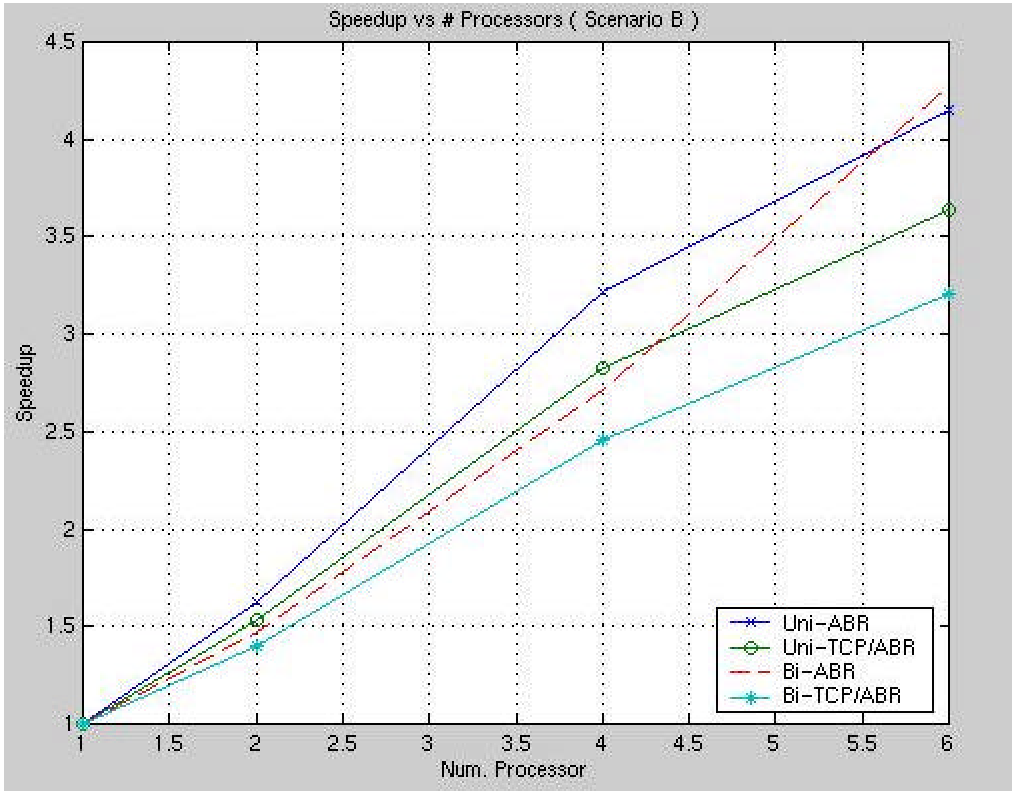
Experiment	# Processors	Execution Time (seconds)	
		GTW	ProTEuS
Uni-directional	2	762.88	327.14
Traffic	4	385.36	239.43
ABR only	6	298.47	178.87
Bi-directional	2	1569.48	527.29
Traffic	4	851.44	335.64
ABR only	6	662.42	257.88
Uni-directional	2	784.74	349.75
Traffic	4	425.72	241.39
TCP over ABR	6	331.21	178.66
Bi-directional	2	1535.20	549.22
Traffic	4	871.57	327.24
TCP over ABR	6	668.90	251.07



Observation

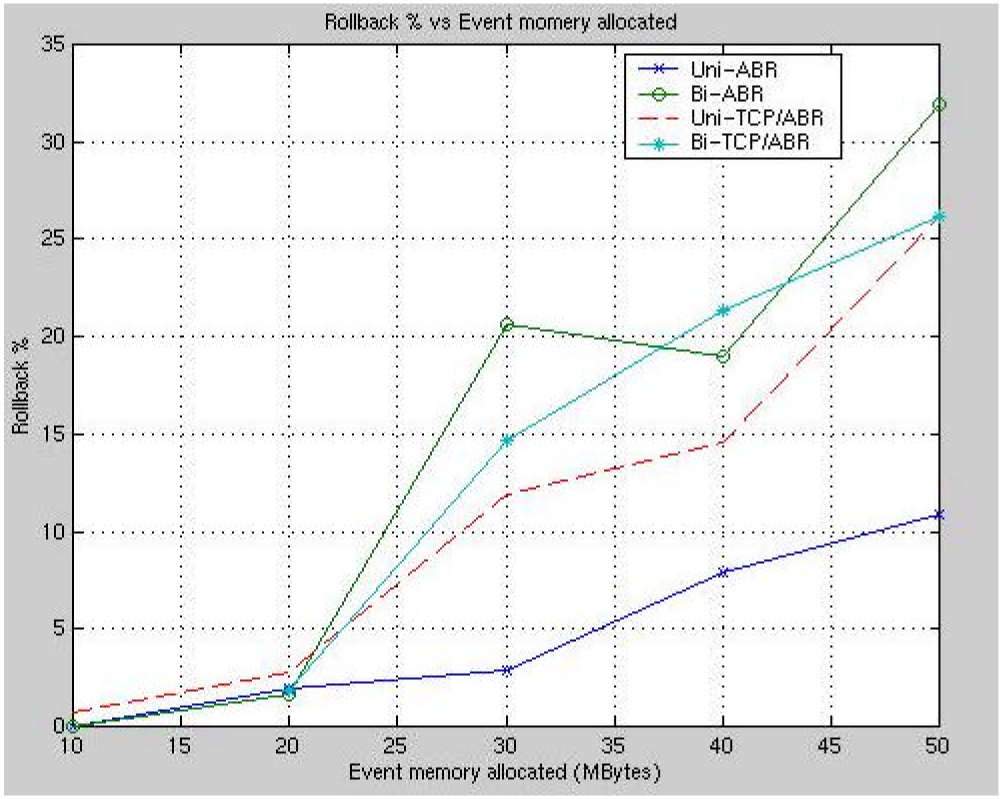
- ProTEuS outperformed GTW by a larger margin

GTW speedup: Scenario B

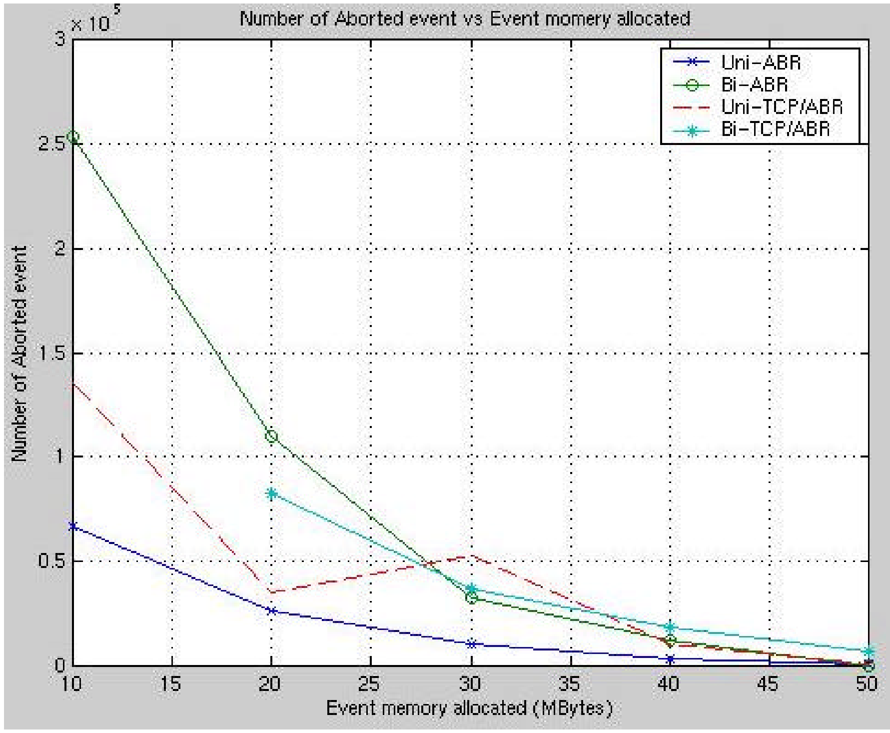


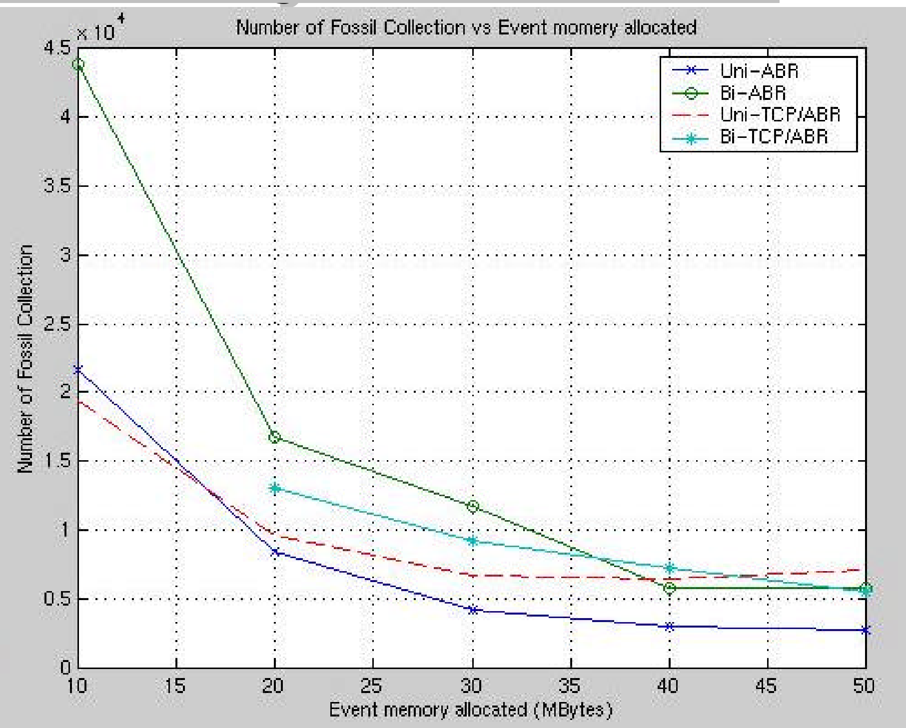
Effect of network characteristics

- Network with feedback loops
  - ABR & TCP
- Increased feedback traffic ==> more Rollbacks
- 6-switch model on 6 processors
- Rollback activity depends on event memory allocation



Effect of event memory allocation





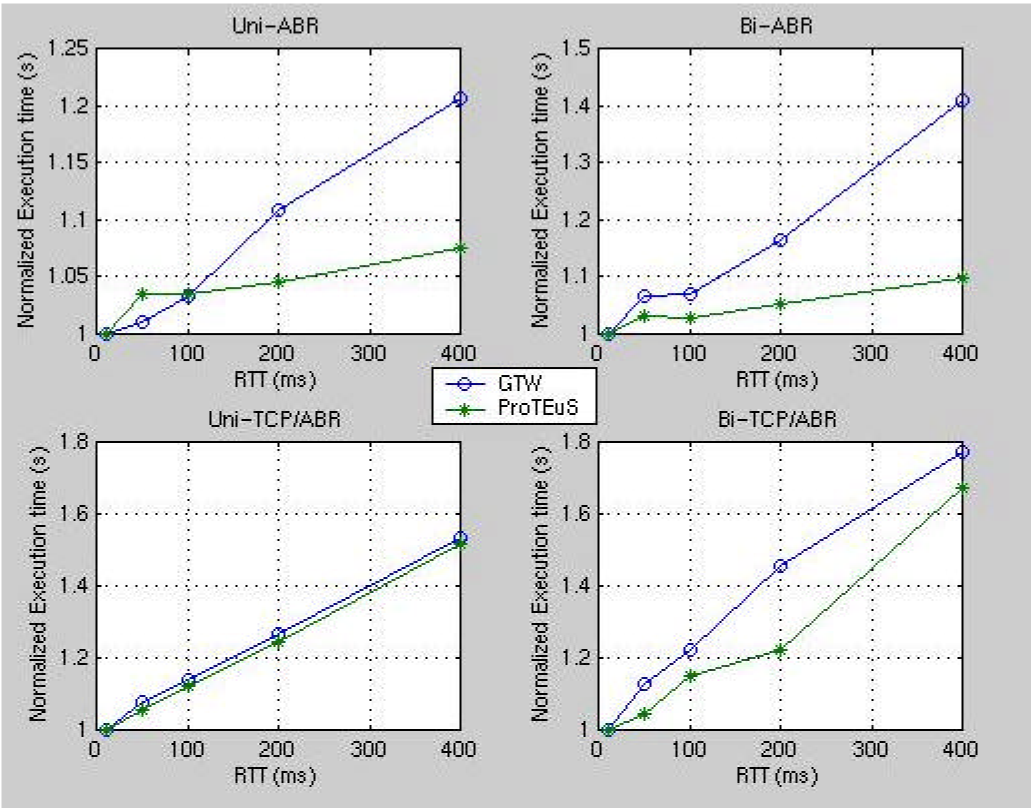
- less event memory ==> events are more likely aborted
- less event memory ==> more fossil collection to reclaim memory for new event
- Aborting event slowed down LP ==> reduce potential rollbacks

Effect of Round Trip Time (RTT)

- 6-switch scenario (6 CPUs used)
- RTT: 10, 50, 100, 200, 400 ms
- Fixed load

Observations

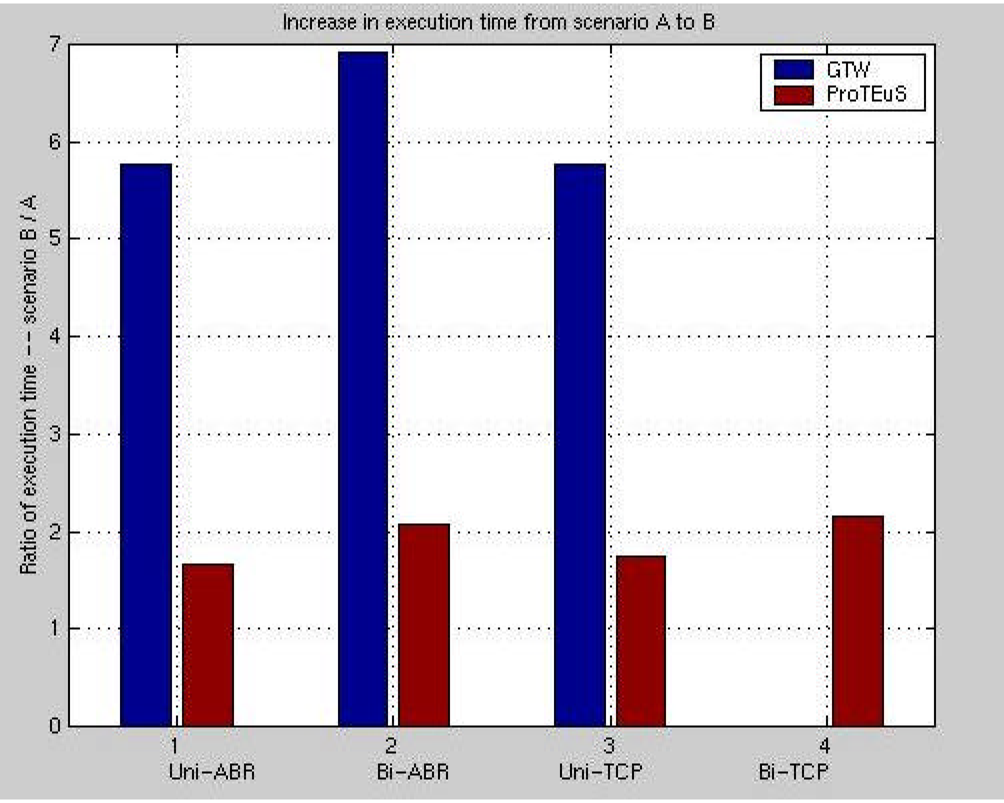
- longer RTT ==> poor performance
- Performance worsen with TCP
- Impact of RTT on ProTEuS is less



Effect of Network Size

- 6 processors used
- simulated time: 10 s
- Network size increases by factor of 3
- Load increases by factor of 5.3

Experiment	Network size / scenario	Execution Time (seconds) GTW	ProTEuS
Uni-ABR	A	610.33	1055.88
	B	3520.28	1754.40
Bi-ABR	A	1134.29	1221.99
	B	7845.38	2528.08
Uni-TCP	A	663.93	1070.77
	B	3834.70	1873.59
Bi-TCP	A	1488.28	1200.08
	B	*N/A*	2579.70



- ProTEuS scales better

Conclusions

- Require careful LP mapping to achieve load balancing
- Require tuning to optimize performance
- Network simulation can benefit from GTW
  - Great speedup on more CPU ==> exploit parallelism
- ProTEuS has better scalability in network size
- Network characteristics impact GTW's performance

Future Work

- Optimize models to reduce memory usage
  - memory consumption limits network size
- Simulate more realistic scenarios
  - Asymmetric topology
  - various kinds of traffics
- Experiment GTW on a NOW platform

Questions ?