EECS 571 Principles of Real-Time Embedded Systems
Lecture Note #16:

Real-Time Communications

Kang Shin (kgshin@eecs.umich.edu)

Real-Time Communication

- Between sensors & control panel and processors
  Between processors
- Non-RT protocols: throughput
  RT protocols: (absolute/prob.) delay guarantees
- Message delay = formatting and/or packetization + queueing + xmission + depacketization
- How to derive message deadlines?
- Message drop preference in case of congestion (pkt marking and classification).
- RT traffic sources:
  - Constant rate
  - Variable rate
- Traffic characteristics may change inside the network

Network Topologies

- Diameter, node degree, fault-tolerance

- Types:

  Shared:
  Point-to-point:

- Switching

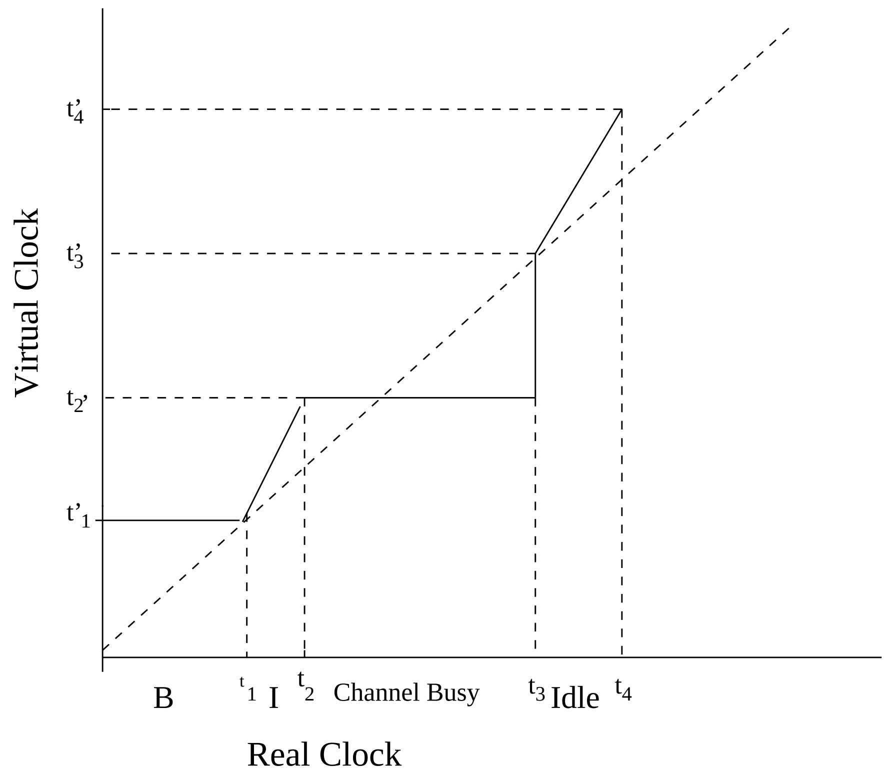
  Packet SW: routing, flow control, & scheduling
  Circuit SW:
  Cut-thru SW: Wormhole, virtual cut-thru

- Interconnections: system and node levels, I/O connectivity

Virtual-time CSMA

- 2 synchronized clocks, "real" and "virtual," for each station.

  RC: to record message arrival times
  VC: to count up to a virtual time to start xmission, VSX(M). *freezes* when channel is busy and *runs faster* than RC when channel is idle.



- When VC$\geq$VSX(M), packet M becomes eligible for xmission.
  If collision, modify VSX value.

- Question: How to initialize VC when channel becomes free, and how to modify VSX(M) when $\exists$ a collision involving M?

More on VTCSMA

τ:	Signal propagation time
A_M:	Arrival time of message M
T_M:	Time required to xmit M
D_M:	Deadline of M
L_M:	Latest time M to be sent to meet D_M
$\Lambda_M(t)$:	$D_M - T_M - \tau - t$

$$VSX(M) = \begin{cases} A_M & \text{for VTCSMA-A} \\ T_M & \text{for VTCSMA-T} \\ L_M & \text{for VTCSMA-L} \\ D_M & \text{for VTCSMA-D} \end{cases}$$

In case of collision, M is rexmitted immediately with prob p; and VSX(M) is modified to be a random # in

$$I = \begin{cases} (\text{current VC}, L_M) & \text{for VTCSMA-A} \\ (0, T_M) & \text{for VTCSMA-T} \\ (\text{current RC}, L_M) & \text{for VTCSMA-L} \\ (\text{current RC}, D_M) & \text{for VTCSMA-D} \end{cases}$$

More on VTCSMA

- When channel changes from busy to idle

$$\text{VC} = \begin{cases} \text{no change} & \text{for VTCSMA-A} \\ 0 & \text{for VTCSMA-T} \\ \text{RC} & \text{for VTCSMA-L and -D} \end{cases}$$

- Effects of clock skew

Node	M	Actual RC at M's arrival	RC at node	D_M	L_M
1	1	8	actual RC-1	32	16
2	2	9	actual RC+1	36	20

VC runs twice faster than RC.
N_1 and N_2 xmit M_1 and M_2 at their VC values 16 and 20, corresponding to their RC values 8 and 10 which are identical.

- Lower loss rate than CSMA but wastes net bandwidth by idling the net.

Window Protocol

- Use events on the bus to synchronize node actions

- Each node maintains a time "window"

- When L_M of a packet falls in this window and the channel is idle, the packet is *eligible* for xmission.

- Each node maintains a stack of window history, (upper bound of window, ID of collided message or 0).

- Shrink or enlarge window based on the events on the bus.

- What does a collision even when $w = 1$ mean and what to do?

- What's a slot?

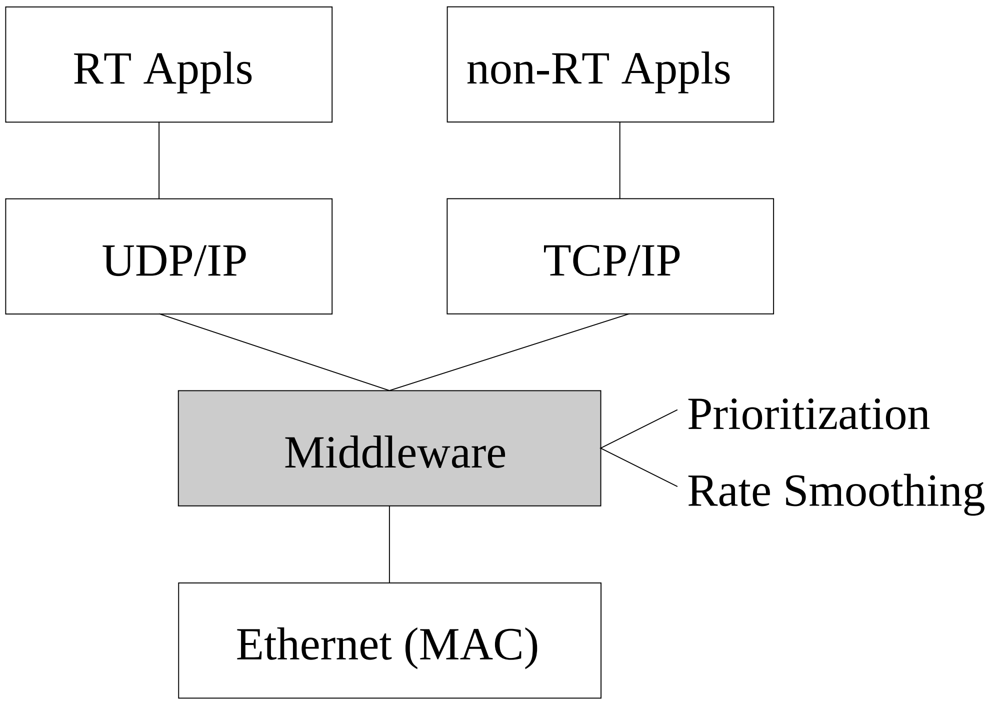
Ethernet-based RT Communication

- Advantages
  - low price & simple management
  - mature technology
- Difficult to provide RT guarantees — CSMA/CD
  - packet collision
  - multiple collisions due to bursty non-RT packets

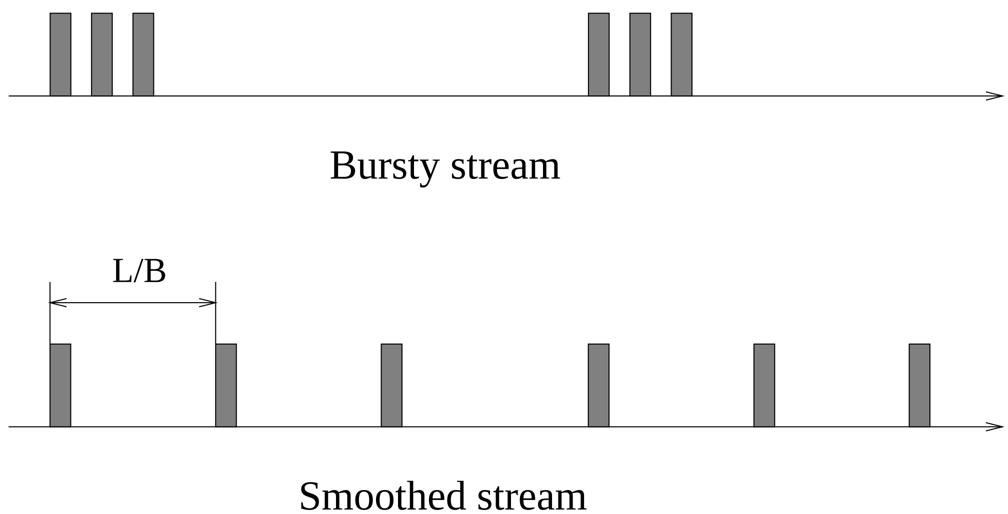
Ethernet MAC protocol

- CSMA/CD protocol
  - No exclusive medium control
  - Listen before transmit
  - Stop transmission upon sensing collision
- Binary Exponential Backoff (BEB): backoff range increases with collisions.

A New Solution



Fixed-Rate Traffic Smoothing



Kang Shin (kgshin@eecs.umich.edu)

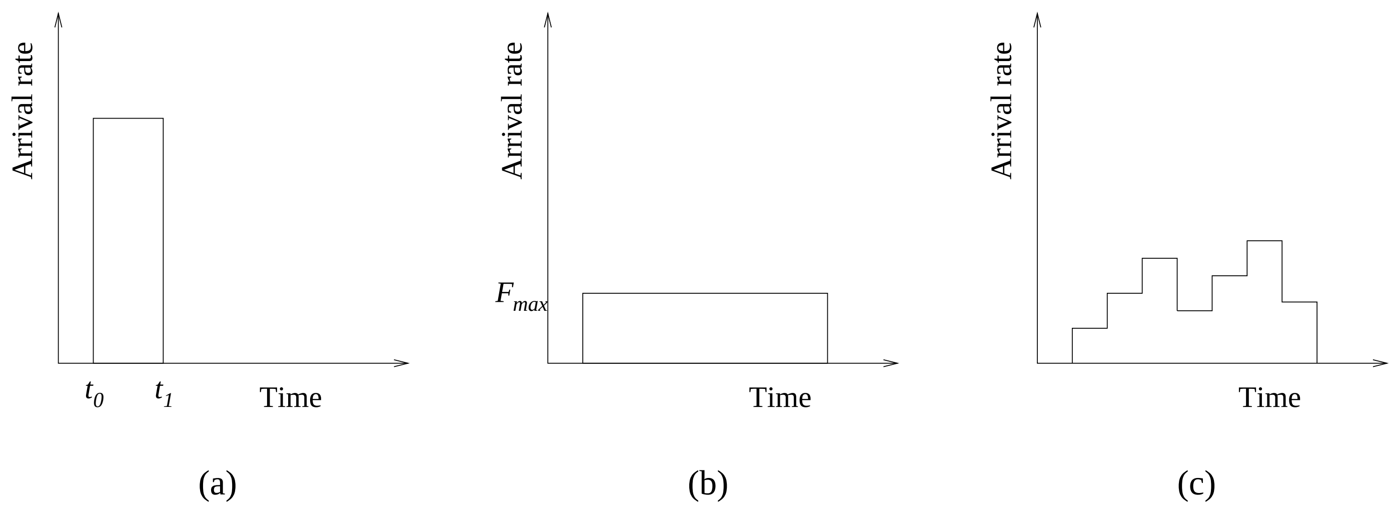
Disadvantages of Fixed-Rate Smoothing

- Poor scalability due to network-wide input limit distributed to stations

- Low network utilization by non-RT traffic
  - o constant station input limit

- Solution – adaptive-rate traffic smoothing

Fixed-Rate vs. Adaptive-Rate Smoothing



Kang Shin (kgshin@eecs.umich.edu)

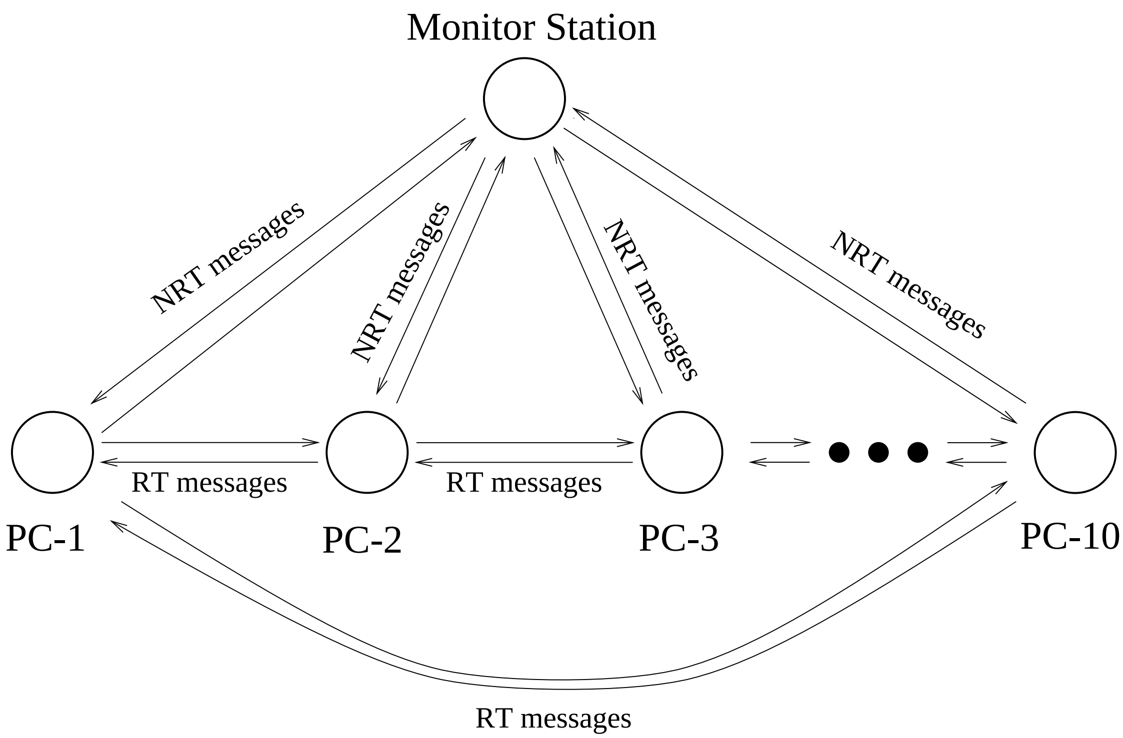
Harmonic-Increase and Multiplicative-Decrease Adaptation

- Parameter: minimum packet inter-arrival time or refreshing period (RP)
- Uses NIC-collected collision statistics
- Adaptation
  - No collision: RP decreased by Δ periodically
  - Upon collision: RP doubled

Experimental Evaluation of Adaptive-Rate Smoothing

- 11 PCs
- 10BASE-T Ethernet LAN
- Collision domain diameter = 10 m

Topology

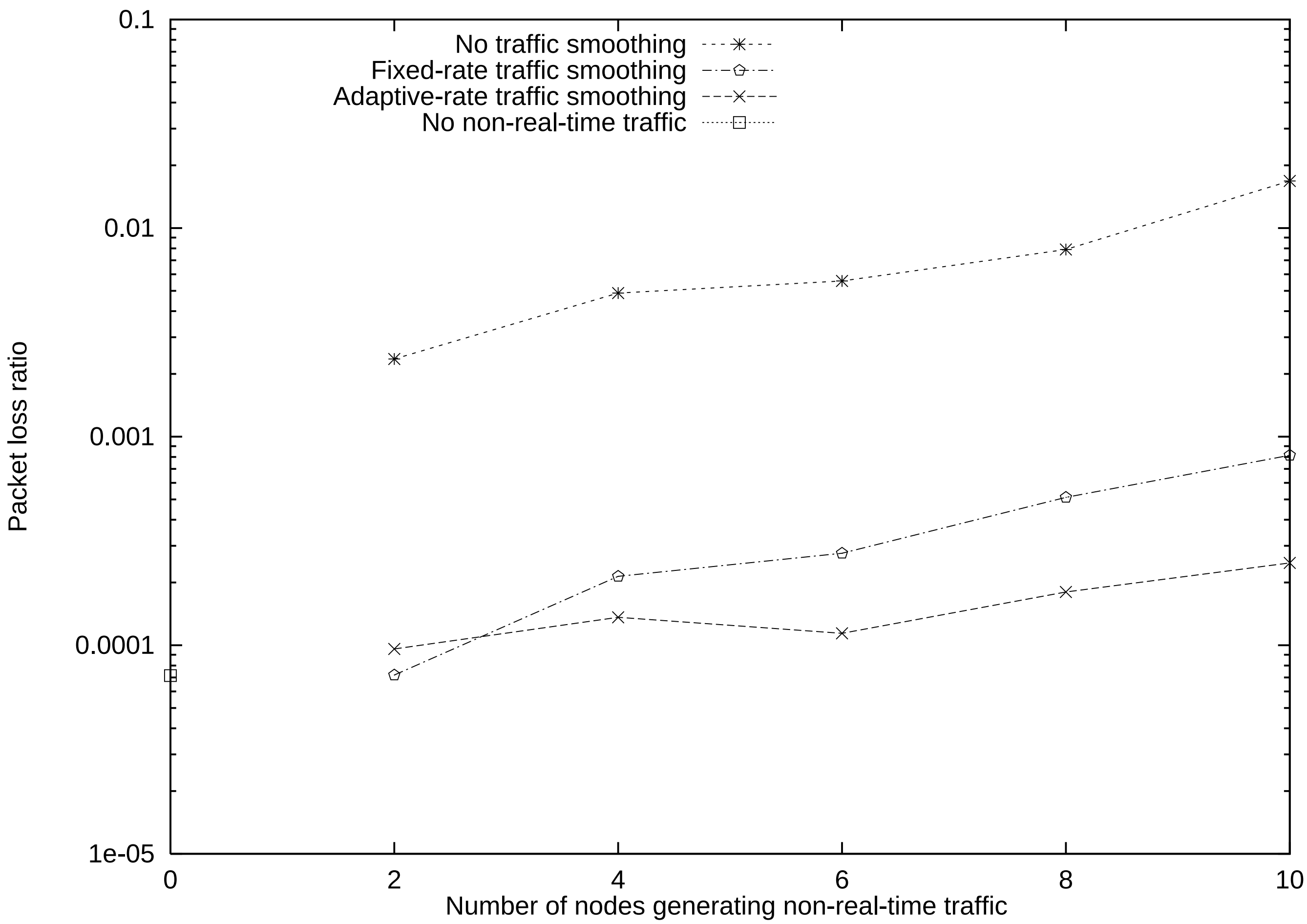

Monitor Station
NRT messages
NRT messages
NRT messages
NRT messages
RT messages
RT messages
PC-1
PC-2
PC-3
PC-10
RT messages

Traffic Generation Statistics

- RT traffic
  - a 100 byte long message every 0.3 sec from PC-n
  - total RT traffic generation rate – 53.3 kbps
  - roundtrip deadline – 129.6 msec

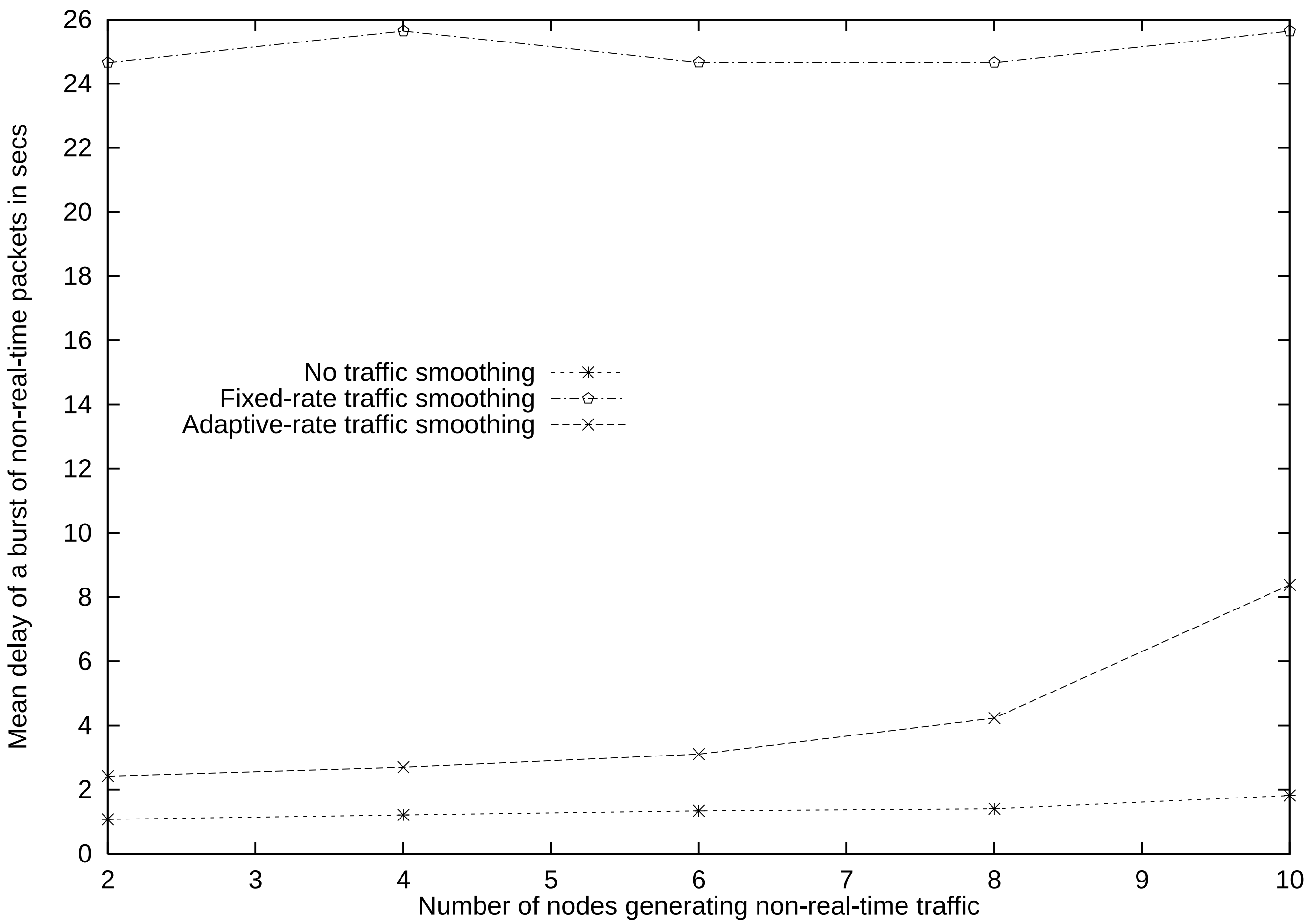
- non-RT traffic
  - non-greedy mode
    - a burst (1 MB) every 25 sec from an activated node
    - 320 kbps from a single node
  - greedy mode
    - zero burst inter-arrival time
    - variable throughput

RT Message-Loss Ratio (non-greedy mode)



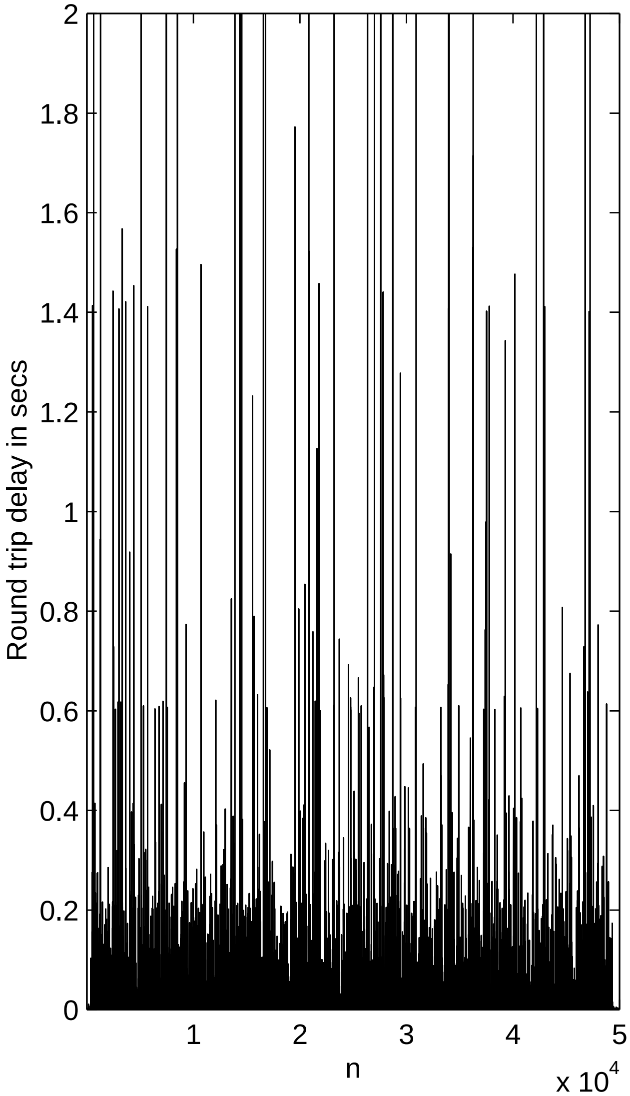
Kang Shin (kgshin@eecs.umich.edu)

Avg Xmit Time of non-RT Bursts (non-greedy mode)

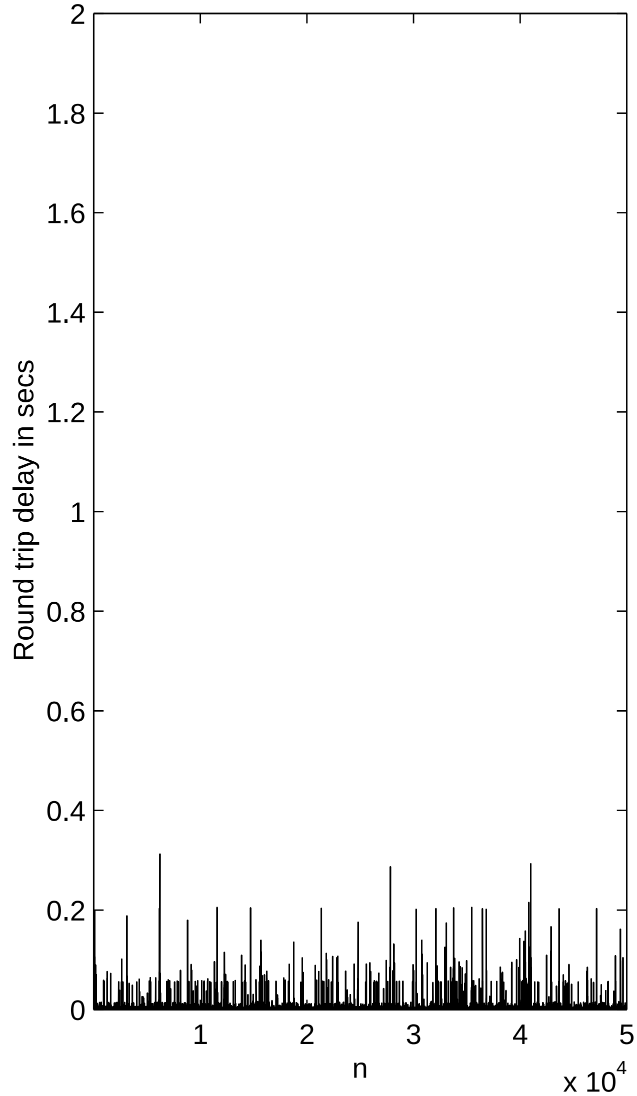


Kang Shin (kgshin@eecs.umich.edu)

RT Traffic Roundtrip Delays

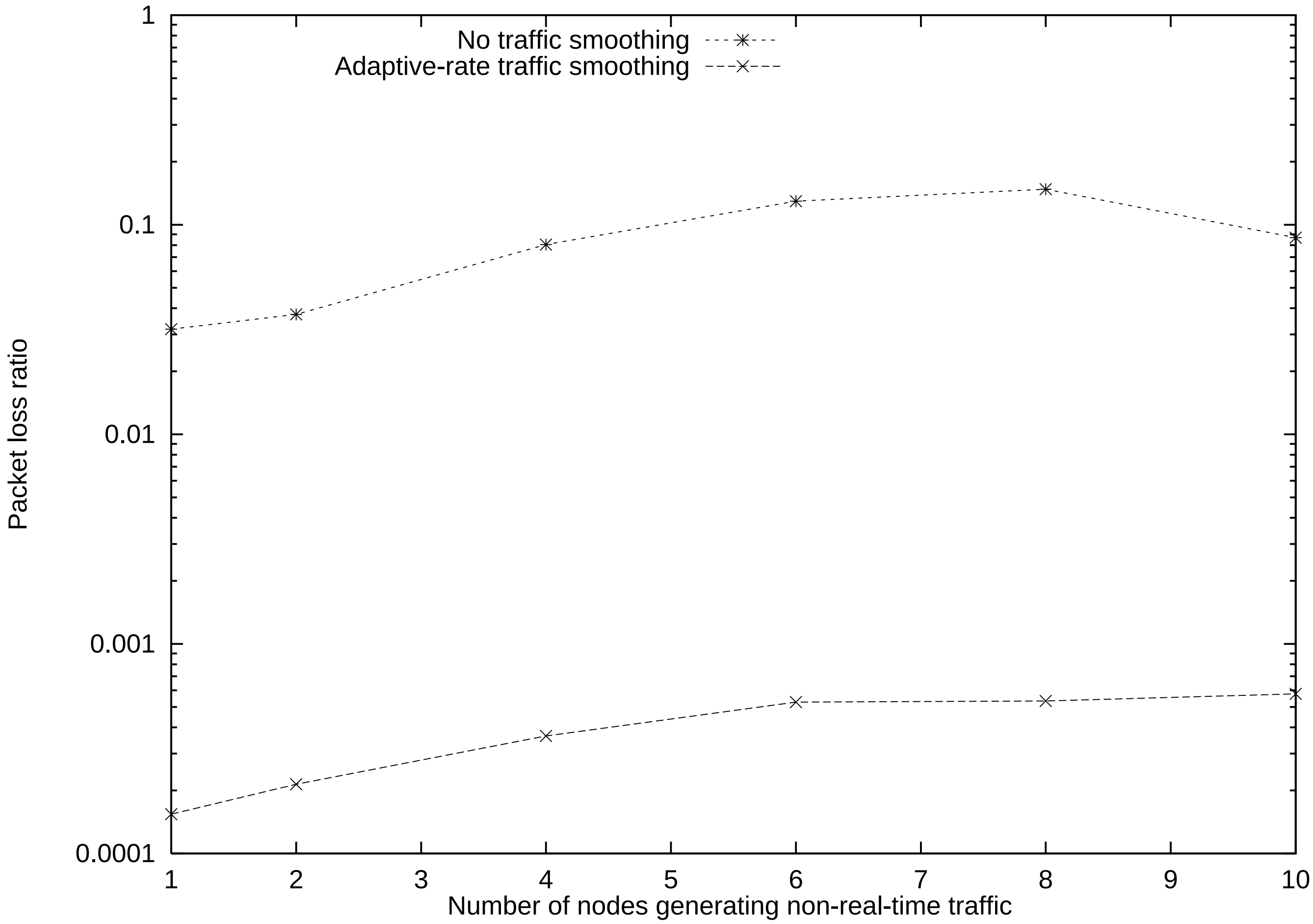


No smoothing

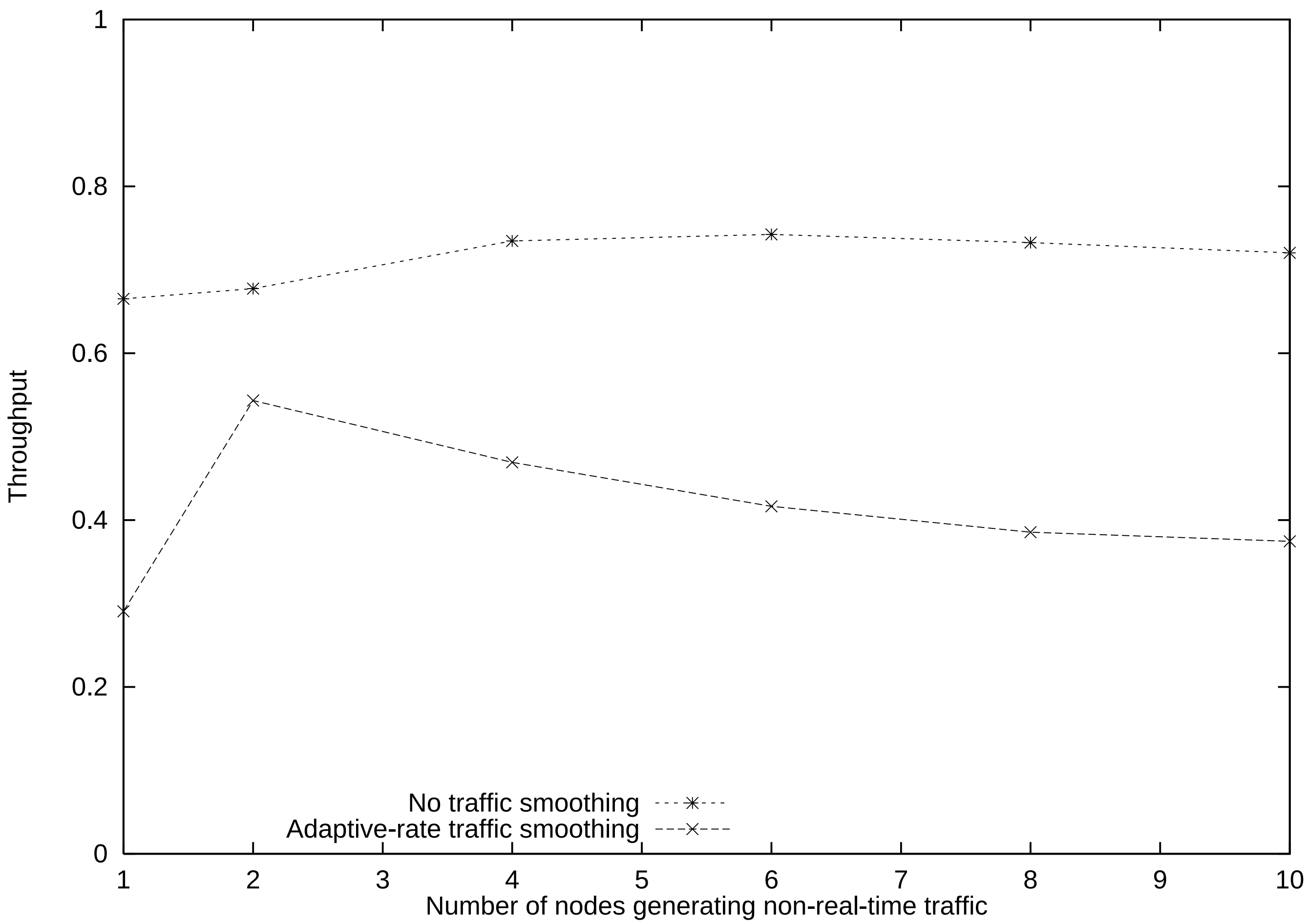


adaptive smoothing

RT Message-Loss Ratio (greedy mode)



Throughput of non-RT Traffic

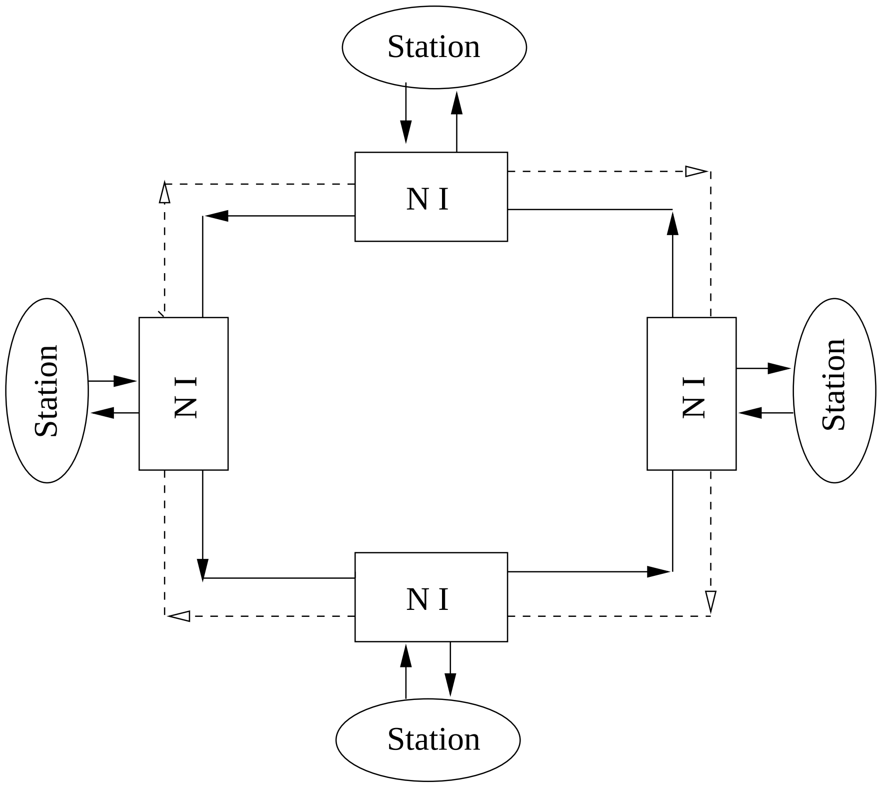


Kang Shin (kgshin@eecs.umich.edu)

Summary on Real-Time Ethernet

- Ethernet augmented with middleware for soft RT guarantees
  - compatible with TCP/IP, UDP/IP and Ethernet standard
  - implemented on Linux and Windows NT
- Prioritization and traffic smoothing
- Fixed-rate traffic smoothing
- Adaptive-rate traffic smoothing

Token-Passing Protocols



Overheads: med. propagation time, token xmission time and capture delay, and NI latency.

Timed-Token Protocol (TTP): 2 types of traffic

- Synchronous/RT: xmit up to h units of time (UOTs) every T UOTs.
- Asynchronous/non-RT: uses any bandwidth left unused by synchronous traffic.

Target Token Rotation Time (TTRT)

- Token cycle time $\leq$ 2 $\times$ TTRT
  How do you determine TTRT?

- Time available to xmit packets per TTRT
  $t_p =$ TTRT $- \Theta$

- Node i gets SBA, $B \times \frac{h_i}{t_p}$.

- If cycle time $>$ ($<$) TTRT token is *late* (*early*). Xmit only synch. traffic up to h_i (AND a certain amount of asynch. traffic).

- *Questions*
  - How to determine h_i?
  - Why token early or late?
  - Is avg token cycle time still $\leq$ TTRT?

Supporting Periodic Communication

- Node i needs to xmit c_i bits of RT traffic every P_i UOTs
  - o TTRT $\leq P_i/2$
  - o Synch. bandwidth is greater than $t_p B c_i / \lfloor \frac{P_i}{\text{TTRT}} - 1 \rfloor$
- How to deal with token loss?
  Claim-token (contains TTRT it requests)
  Beacon packet if every node doesn't receive its own claim-token or a normal packet within TTRT(i) after xmitting claim-token.
  Station immediately downstream from a break is the only node that will keep xmitting.

IEEE 802.5 Token Ring Protocol

Token

SD	AC	ED

SD	AC	ED	DA	SA	Message	ECC	ED	FS

SD: Starting Delimter
AC: Access Control
ED: Ending Delimiter
DA: Destination Address

SA: Source Address
FS: Frame Status
00 dest unavailable
10 frame uncopiable
11 frame copied

- FS field checked by the sender
- Sender removes data frame it sent
- Priority arbitration via AC field
  - o 3 bits for current and reserved priorities
  - o When data frame or a token goes by, a node checks the reserved priority: do nothing if higher, else write its priority into AC
  - o Upon completion of current xmission, the sender issues a token with priority in AC
  - o Node that increased priority is responsible for restoring it to prior priority value
- Schedulability Analysis of Token Ring
  $T_1, T_2, \ldots, T_n$ are schedulable iff $\forall i \;\; \exists t \leq d_i$ such that $\sum_{j+1}^{i} e_j \lceil \frac{t}{P_j} \rceil$ + system overhead + $b_i \leq t$.

Polled Bus Protocol

- Synchronous contention resolution
- Stations contend for the channel during a contention period
- The station with the highest priority message gets access to the medium
- Example: Controller Area Network (CAN)

Bus Acquisition Algorithm

- Calculate a unique ID of m bits.

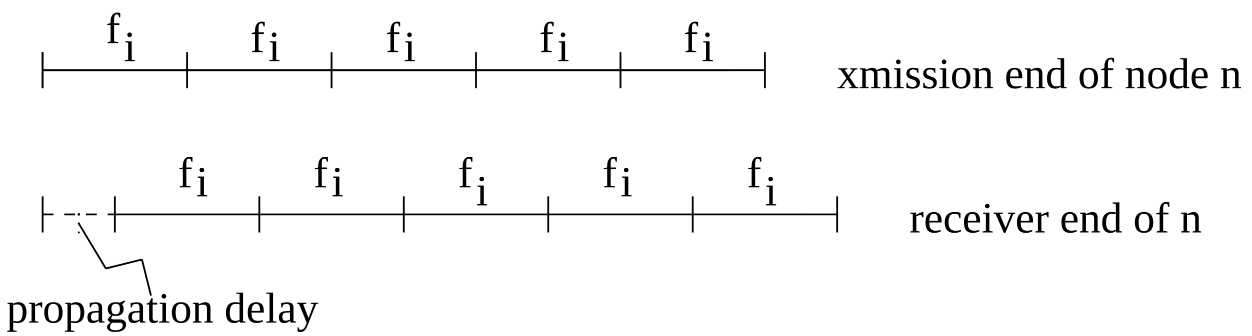
- If bus not busy then
  - o Write its ID to the bus, one bit at a time, starting with MSB
  - o Wait for a finite time and sample the bus
  - o If the value read by the processor is different from the value it wrote into the bus, it drops out
  - o After m rounds, the processor with the highest ID has sole control of the bus.

- The bus is assumed to wired-OR all impinging signals and all stations are synchronized.

- Key: how to design station IDs?

Stop-and-Go or Framing Protocol

- Frame = a time interval
  Each frame type represents a diff time interval and is associated with a *traffic class*.
- Upon arrival of a type-f_i pkt at an intermediate node n, it's held by n at least till the beginning of next instance of f_i.



- All nodes eligible for xmission are served in nonpreemptive priority order, with shorter-frame pkts having priority over longer-frame pkts
- If net load $\leq$ a certain limit, a type-m pkt will be xmitted within f_m time units of being eligible for xmission, i.e., bounded delay at each hop

Hierarchical Round-Robin Protocol

- Guarantees each traffic class i to xmit m_i pkts every T_i UOTs.

- Traffic is classified into n classes, where each class i is associated with (n_i, b_i, Φ_i)

  - o n_i: max # of class-i pkts xmitable during any given frame of which source j is allocated a certain max # $\alpha_i(j)$.

  - o If $\alpha_i(j)$ pkts are xmitted or no class-i pkts left for xmission, class-$(i + 1)$ pkts if any are xmitted, etc., for a max of b_i pkts during that class-i frame

  - o Φ_i = frame associated with class i
    $\Phi_1 < \Phi_2 < \ldots < \Phi_n$.