4. Real-Time Communication

Roadmap for Section 4

- Characteristics of RT Communication
- Communication Media
- Network Topologies
- Protocols

Introduction

- Effective communication among various devices is vital for functioning of a real-time system
  - Sensors, control panels, processors, actuators, output displays
- Hard RT:
  - Communication overhead must be bounded
- Soft RT
  - occasional failure to meet deadlines is not fatal
  - multimedia, teleconferencing
- Goal of RT communication:
  - Maximize probability of in-time delivery
  - Lost message: infinite delivery time

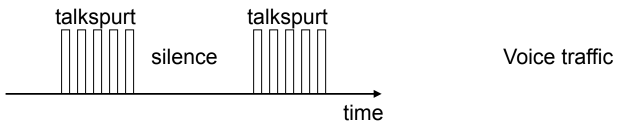
Characteristics of RT Comm.

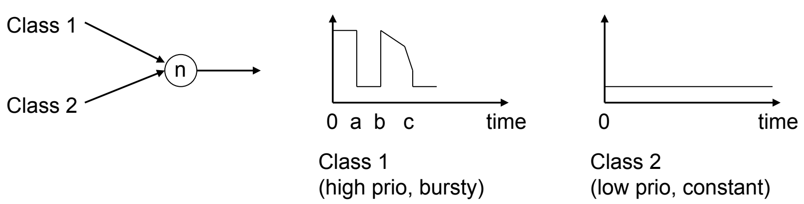
- Overheads:
  - Formatting/packetizing the message
  - Queueing the message; waiting for access to comm. medium
  - Sending the message from source to destination
  - Deformatting the message
- RT traffic uses multiple message classes
- Characterized by:
  - Deadline
  - Arrival pattern
  - priority

Categories of RT communication

- Constant rate traffic:
  - Fixed sized packets periodically created
  - Not bursty; easy to handle
- Variable rate traffic:
  - Fixed sized packets irregularly created
  - Variable sized packets
  - Voice + video
- Traffic characteristics can change as packets flow through multiple hops of a network
  - Traffic classes compete for bandwidth at intermediate nodes

Example: multi hop network

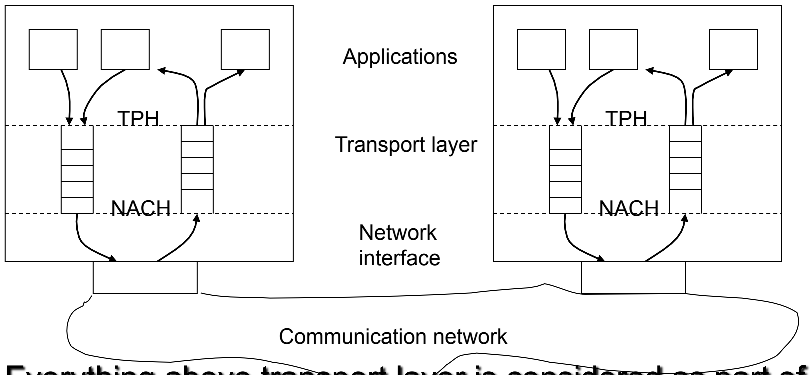




No class 2 output in [0,a] and [b,c] at node n

Class 2 output becomes bursty at node n

Model for Real-Time Communication



- Everything above transport layer is considered as part of the application

Model for RT Comm. (contd.)

- Queues for incoming and outgoing messages
- TPH - Transport Protocol Handler
  - Interface to local applications
  - Implements transport service
- NACH - Network Access Control Handler
  - Interface to network
  - Implements network access and transport service for TPH
- Packet
  - Messages are divided into fixed-sized packets
  - Transmission of a packet cannot be interrupted
  - Frames, segments, cells

RT Traffic Model

- Messages typically are part of a typed data stream
- Periodic messages:
  - Transmission of periodic messages is a periodic task
  - Characterized by period p, duration of transmission e, and deadline D
- Aperiodic messages:
  - No knowledge about deadlines and inter-arrival rates
  - Best effort delivery
- Sporadic messages:
  - Aperiodic messages with known deadlines and inter-arrival rates
  - Characterized by average inter-arrival rate, average period, and minimal inter-arrival rate

Performance Metrics for RT Comm.

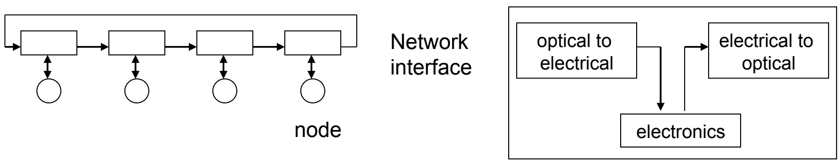
- Rates:
  - lost messages, missed deadlines, corrupted messages
  - Certain minimum values need to be guaranteed
- Throughput:
  - Number of messages per time unit
  - Scheduling and flow control algorithms are based on a assumed minimal throughput value
- End-to-end delay:
  - Duration of a transmission from sending to receiving task
  - Essential for control systems; tolerable for multimedia systems
- Delay jitter:
  - Variation in end-to-end delay
  - Can be minimized through buffering - thus increasing the delay

Communications media

- Electrical medium:
  - Twisted pair - several kHz frequency
  - Coaxial cable - up to 450 MHz frequency
  - T-connections, vampire tap
- Optical fibers:
  - Laser diodes, light pulses - 10 GHz frequency
  - Pulse amplitude decreases (attenuation) and pulse width increases (dispersion) when light pulse travels
  - Required power at receiver determines size of network
- Advantages of optical fibers:
  - High bandwidth
  - Optical signals are immune to electromagnetic interference

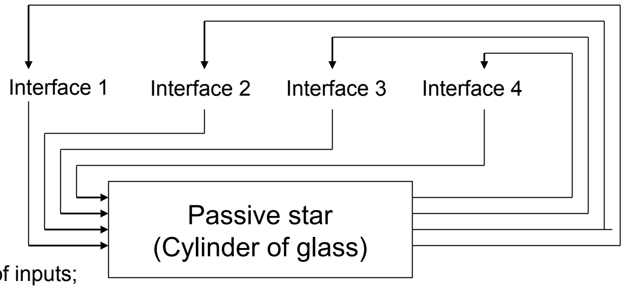
Optical fibers (contd.)

- Disadvantages:
  - Difficult to tap
  - Only two topologies work well: point-to-point, passive star



Optical signal is re-created at every non-receiver node

Passive star topology

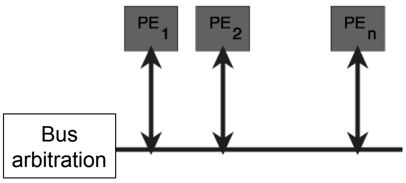


Output = sum of inputs;
Problem: signal is divided among receivers (power)

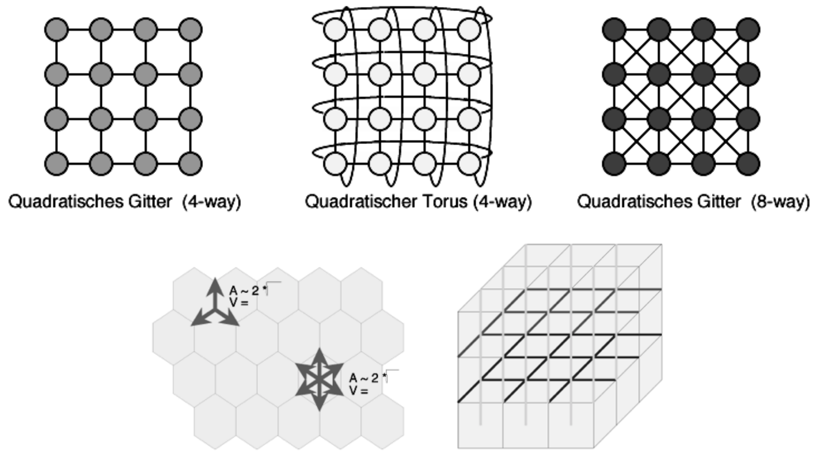
- Mismatch:
  - frequency (bandwidth) of network interface <-> bandwidth of media
  - Use WDM: wavelength-division multiplexing
  - Tunable lasers implement different channels

Network topologies

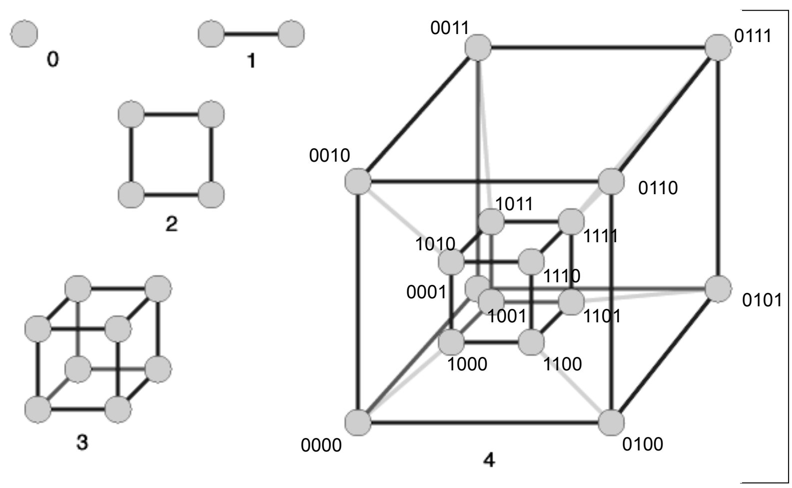
- Characteristics:
  - Diameter: maximum distance between any two nodes
  - Node degree: number of edges adjacent to each node
  - Fault tolerance: measures to what extend link/node failures can be tolerated
- Classes:
  - Point-to-point (rings)
  - Broadcast/shared (buses)
  - Mesh
  - hypercube



Meshes and toruses



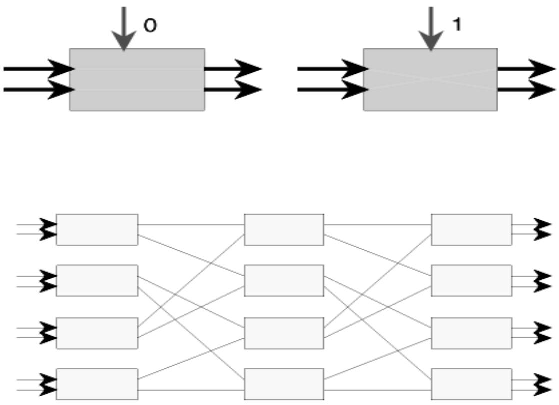
Hypercubes



Embedding a hypercube in a passive star - wavelength allocation

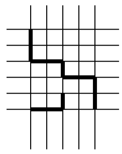
From	To	Wavelength	From	To	Wavelength
000	001	λ_1	100	000	λ_{13}
000	010	λ_2	100	101	λ_{14}
000	100	λ_3	100	110	λ_{15}
001	000	λ_4	101	001	λ_{16}
001	011	λ_5	101	100	λ_{17}
001	101	λ_6	101	111	λ_{18}
010	000	λ_7	110	010	λ_{19}
010	011	λ_8	110	100	λ_{20}
010	110	λ_9	110	111	λ_{21}
011	001	λ_{10}	111	011	λ_{22}
011	010	λ_{11}	111	101	λ_{23}
011	111	λ_{12}	111	110	λ_{24}

Multistage network



Sending Messages

- Packet switching:
  - Message is broken down into packets
  - Standard or variable length
  - Packet headers specify source and destination information
- Circuit switching:
  - Circuit between source and destination is set up for whole transmission time
- Wormhole routing:
  - Pipelining packet transmission in a multihop network
  - Each packet is broken down into train of flits (flow control digits)
  - Only header flit has destination address
  - Problem: danger of deadlocks



Embedded Systems Programming

19

Media access

- Protocol required for access of a broadcast medium
- CSMA/CD
  - Carrier Sense Multiple Access / Collision Detection
  - No priorities
  - Possibly unbounded communication delay
  - Not a good choice for real-time systems

Embedded Systems Programming

20

Media Access - physical problems

- Physical transmission has limited speed
- “Happens simultaneously” depends on
  - physical distance between sender and receiver
  - clock resolution
- This problem can be ignored if latency << transmission time for a single bit
- Propagation delay might be important
  - 100 MBit/s, 200m distance
  - Transmission of a single bit: 10 ns
  - Propagation delay: 1000ns (transmission speed 2/3 c)
  - Channel “stores” 100 bits
- Collisions can be detected only after propagation delay

Protocols

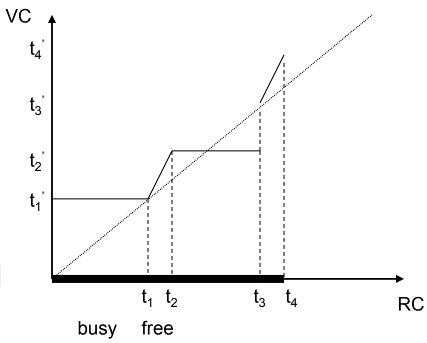
	Deadline guarantee	Network topology
VTCSMA	no	broadcast
Window	no	broadcast
Timed token	yes	ring
IEEE 802.5	yes	ring
Stop and go	yes	Point-to-point
Polled bus	no	bus
Hierarchical round robin	yes	Point-to-point
Deadline-based	no	Point-to-point

VTCSMA

- Virtual-Time Carrier Sensed Multiple Access
  - Variation of standard CSMA (Ethernet)
  - All nodes monitor channel
  - With synchronized clocks, CSMA can be used to implement priority algorithm
- Suppose a node has ha set of packets to transmit
- Locally known:
  - State of channel
  - Priorities of waiting packets (local priorities)
  - Time according to synchronized clock
- VTCSMA uses time to establish global priorities

Operation of VTCSMA

- Real-time clock RC
- Virtual clock VC
  - Stopped when channel is busy
  - Reset when channel free
  - Runs faster than RC (rate η > 1)
- Virtual times are equal on all nodes
- VSX(M): start time for transmission of M
  - VC >= VSX(M) -> transmit M
  - Collision? -> modify VSX(M)

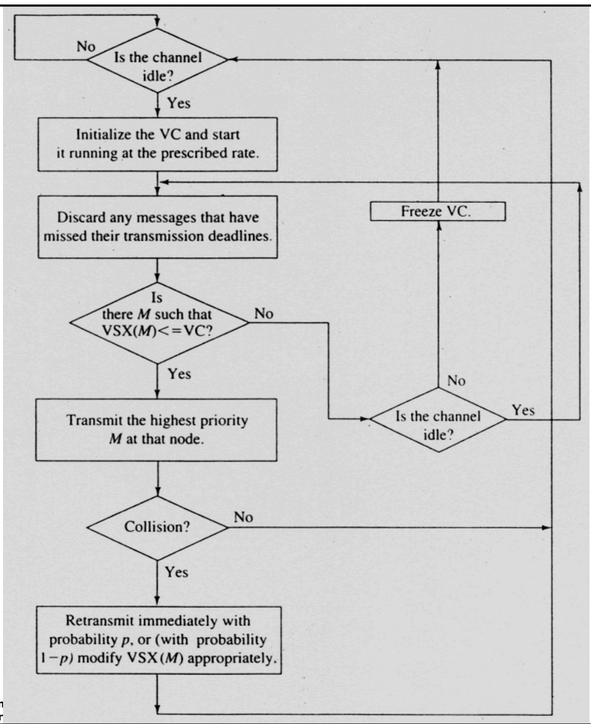


VC freezes when channel becomes busy and is reset to RC when channel becomes free

Variations of VTCSMA

- VSX(M) =
  - A_M: Arrival time of message (or packet) M - VTCSMA-A
  - T_M: Time required to transmit message M - VTCSMA-T
  - L_M: Latest time for send to meet deadline of M - VTCSMA-L
  - D_M: Deadline for delivery of M at destination - VTCSMA-D
- Collision: VSX(M) randomly chosen from interval I =
  - (current VC, L_M) - VTCSMA-A
  - (0, TM) - VTCSMA-T
  - (current RC, L_M) - VTCSMA-L
  - (current RC, D_M) - VTCSMA-D
- Channel changes from busy to idle; VC =
  - No change - VTCSMA-A
  - 0 - VTCSMA-T
  - RC - VTCSMA-L and VTCSMA-D

VTCSMA



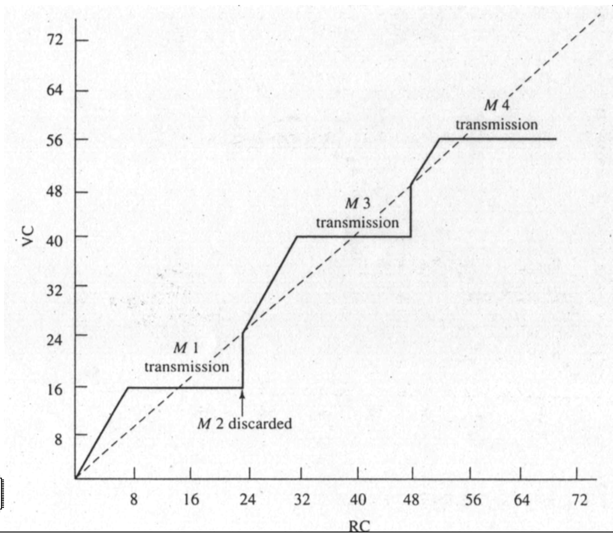
- Flowchart of the VTCSMA algorithm

VTCSMA Example

Example 6.9. As an example of how these algorithms work, consider the VTCSMA-L algorithm. Let $\eta = 2$ (i.e., the VC runs twice as fast as the RC when the channel is idle). Let us assume that the transmission time for each packet is $T_M = 15$, and that the propagation time is $\tau = 1$. Suppose the packets arrive according to the following table:

Node	M	RC at arrival	D_M	L_M
1	1	0	32	16
2	2	10	36	20
3	3	20	56	40
4	4	20	72	56

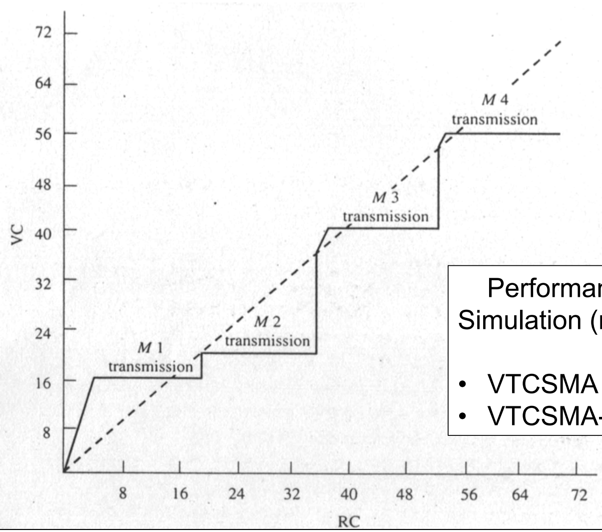
VTCSMA-L Example (contd.)



- VC-RC trajectory:

  $\eta = 2$

VTCSMA-L Example (contd.)



- **VC-RC trajectory:**

  $\eta = 4$

Performance evaluation:
Simulation (no analytical solution)

- VTCSMA is better than CSMA
- VTCSMA-D performs best

Window protocol

- Also based on collision sensing
- Window:
  - Time interval, identical at all nodes
  - If LTTT (latest time to transmit) falls into window
  - > packet becomes eligible for transmission
- Slot:
  - Time for end-to-end network propagation
  - > transmission only at beginning of a slot
- Window size is changed based on history on media
  - Collision: decrease window size
  - No traffic: increase window size
- (Window size == 1) && collision:
  - Random retransmission within next slot

Algorithm maintains a stack of two-tuples (u, i) in each node;

- u - upper bound of a window in which collision occurred
- i - zero, unless node has a message involved in collision

Window protocol - algorithm

```
Initialization:
   up := t + δ;
   empty the stack;
   If one or more messages have LTTT in the interval [t,up), transmit
        the one with minimum LTTT.

At the beginning of each slot:
   Find out if there is any message with (LTTT < t),
        and drop it, since it has missed its deadline.
   Discard any stack contents whose u field is less than t;
   If the channel is busy due to a collision, then
        abort any ongoing transmission by this node;
   else if the channel is idle following a collision
        contract_window_and_send(up, t);
   else if the channel is occupied by a message transmission
        continue the transmission, if any;
   else if the channel is idle after a successful transmission, then
        pop_and_send(up, t);
   else if the channel continues idle, then
        expand_window_and_send(up, t);
   end if;
end;
```


Window protocol - discussion

- Performance:
  - Comparable to centralized media access control algorithms
  - Initial window size is relatively unimportant
  - Deadline anomaly: loose deadlines may result in worse transmission characteristics
- Discussion contention protocols:
  - Work best when traffic is light and end-to-end delay is small in comparison to bit transmission time
  - Heavy traffic: probability of collision increases; bandwidth wasted
  - If end-to-end delay is high -> collisions are detected late

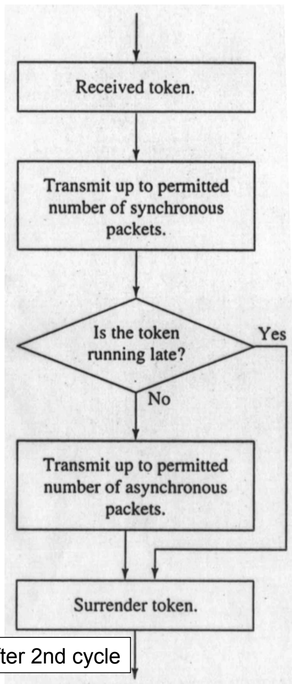
Token-based protocols

- Timed token protocol
  - Synchronous traffic: real-time traffic
  - Asynchronous traffic: non real-time traffic; takes up unused bandwidth
- Key: target token-rotation time TTRT
- Protocols attempts to keep rotation time < TTRT
  - > not always possible
  - Rotation time <= 2* TTRT is guaranteed
- Node may transmit pre-assigned quota at every token visit

Embedded Systems Programming 33

Operation of timed token protocol

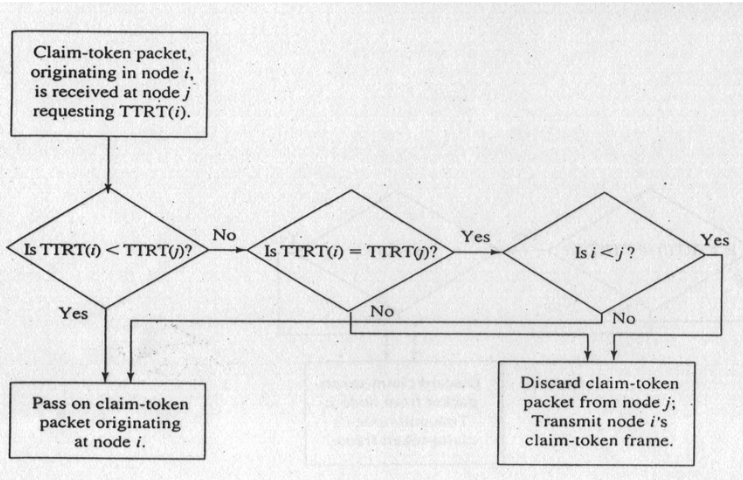
- First round:
  - Broadcast to determine TTRT
  - Smallest requested value is used
- How does a node decide how much asynchronous traffic to transmit per cycle
  - > problem: asynchronous overrun
- Token loss / fault:
  - Token late in two consecutive cycles: indication of failure
  - > start renegotiation of TTRT
  - Claim token protocol



Flowchart of timed token protocol after 2nd cycle

Embedded Systems Programming 4

Claim token protocol



Token ring protocol

SD	AC	ED

Token format

Data packet format

SD	AC	ED	DA	SA	message	error control code	ED	FS

SD = starting delimiter; AC = Access control; ED = Ending delimiter

FS = Frame status; SA = Source address; DA = Destination address

- Frame status (FS):
  - FS = 00: destination available (powered up)
  - FS = 10: frame could not be copied at destination
  - FS = 11: frame successfully received at destination
- Access Control (AC):
  - 3 bits for priority (current priority)
  - 3 bits reserved priority

Operation of token ring protocol

- Node n_i checks reserved priority (reservation bits)
- If (res.prio) >= p_i:
  - Do nothing
  - Network is in use with same / higher priority
- If (res.prio) < p_i:
  - N_i writes res.prio = p_i
  - When current transmission has completed, sender issues token with priority indicated by reservation bits
- Implementation problem:
  - IEEE 802.5 allows only for 8 priority levels

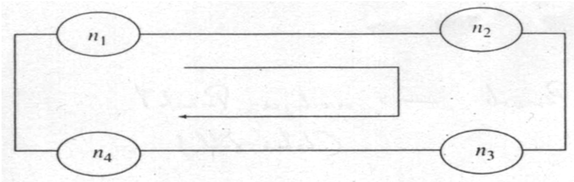
Schedulability Criterion for token ring protocol

Theorem 6.4. The task set $T_1, T_2, \ldots, T_n$ is schedulable iff for all $i = 1, \ldots, n$, there is some t, $0 < t \le d_i$ such that

$$\sum_{j=1}^{i} e_j \left\lceil \frac{t}{P_j} \right\rceil + \text{System overhead} + b_i \le t$$

- e_j: execution time associated with sending a message
  - Time to capture the token when node has high priority message
  - Time to transmit the message
  - Time to transmit the token when packet transmission is over
- b_j: maximum time for which a T_j packet can be blocked
- P_j: defined by application (perio) P_i >= d_i
- d_i: deadline associated with task T_i

Token ring protocol - example



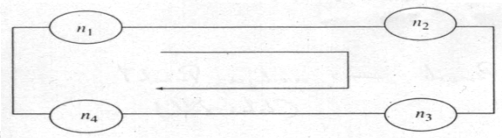
Example 6.17. Consider the token ring shown in Figure 6.35. Node n_1 has traffic generated at periods of 5, 6, and 10, respectively; n_2 periods of 5, 9, and 11, n_3 periods of 4 and 6, and n_4 of period 10. Table 6.3 shows the priorities of the packets[6] awaiting transmission on each node along with their arrival time.

Node	(Period, arrival time)
n_1	(5, 5), (6, 6), (10, 10)
n_2	(5, 5), (9, 9), (11, 11)
n_3	(5, 5), (4, 10)
n_4	(10, 10)

[6] The lower the priority number, the higher the priority.

Token ring - example (contd.)

Consider what happens when the token flows past the four nodes at times 6, 7, 9, and 10, respectively. At time 6, node n_1 writes the priority of its most important packet on the reservation field. At time 7, n_2's highest priority is the same, so it does not overwrite the reservation field. At time 9, when the token goes past it,



n_3's highest priority is also the same, and so again the token reservation bits are not changed. The reservation bits are also unchanged by n_4. Suppose the token returns to n_1 at time 12. The reservation bits allow transmission of a packet with period 5. Since n_1 has a packet of this period, it is allowed to transmit. At this time, though, it is not n_1 but n_3 that has the highest-priority packet (of period 4). However, n_3 will not have a chance to update the reservation bits until the next transit of the token, and so the decision to allow n_1 to transmit is based on outdated information.

Hierarchical round robin protocol

- Bound delay encountered by a packet at each intermediate node
  - > have bound for total network delay
- Traffic is classified into classes
- Each traffic class has own transmission frame (n_i, b_i, Φ_i)
  - n_i: maximum number of $class_i$ packets that may be transmitted during a frame;
    if exhausted: transmit $class_{(i+1)}$ packets
  - b_i: maximum overall number of packets during class-i frame Φ_i
- Messages that must be delivered quickly must have particularly short frames
- Non-mode conserving:
  - sender may be idle with high priority packets waiting

Hierarchical round robin - example

- Consider a system of three classes with the following allocations:

i	n_i	b_i	g_i
1.	3	3	6
2	3	1	10
3	1	0	20

Source	Class	Allocation
s_1	1	3
s_2	1	1
s_3	2	2
s_4	3	1

In each frame Φ_1 of duration 6, $class_1$ traffic takes up three slots;
the rest is reserved for classes 2 and 3
In each frame Φ_2 of duration 10, $class_2$ traffic takes up three slots;
with one being reserved for $class_3$