Real-Time (Paradigms) (51)

5. Real-Time Communication

Data flow (communication) in embedded systems :

Sensor --> Controller

Controller --> Actor

Controller --> Display

Controller <--> Controller

Major challenge in real-time communication:

The communication delay adds to the computer response time.

---> the delay has to be bounded and to be known, even in the presence of disturbing factors such as other (RT) traffic, variable load, or faults

Different goals ---> different key performance measure

In non-real-time systems: Throughput

In real-time systems: Guarantee (high probability) of delivery of a single message within a certain deadline

Real-Time (Paradigms) (51a)

Classes of real-time network traffic, each characterized by the following parameters:

by deadline:

- Hard real-time: deadline of message related to deadline of tasks
- Multimedia: deadline of messages application-dependant

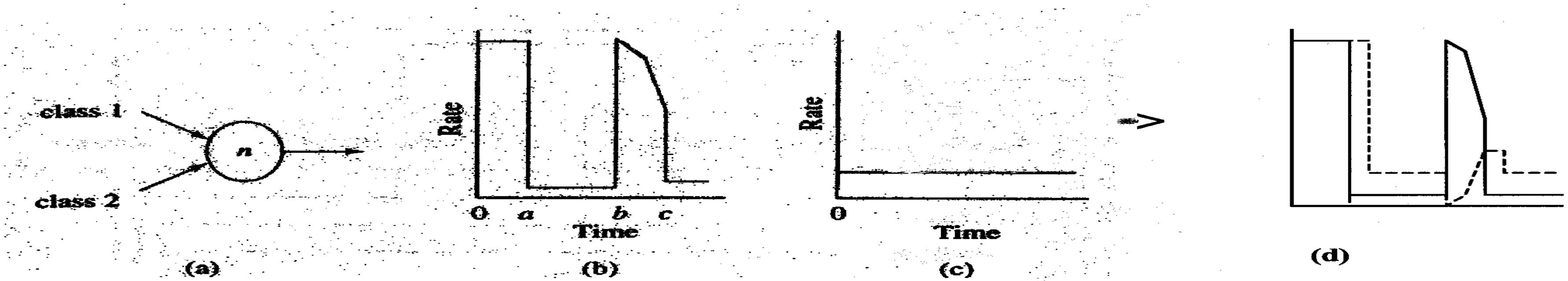
by priority:

- Defines which messages are transferred first
- In overload situations low priority messages are dropped
- Priority not necessarily related to the importance of a message (see Rate Monotonic Scheduling)

by arrival pattern:

- Constant rate: fixed size packets at periodic interval (Example: sensor data)
- Variable rate: variable size packets and/or irregular interval (Example: voice or video data)

Character of traffic might change while traveling through multiple hops in the network:



a) two classes competing at a node b) High priority, bursty c) Low priority, constant d) Resulting traffic

Real-Time (Paradigms) (51b)

Architectural issue

Hard RT systems are often designed with dedicated Network Processor (NP).

Main functions of a NP:

- Execute operations necessary to deliver a message from a source to its destinations.
- establish connections between source and destination nodes
- handle end-to-end error detection and message retransmission
- routing (select primary and alternative routes)
- allocate bandwidths necessary to guarantee timely delivery
- packetize information into data blocks and reassembling at the destination
- implement buffer management policies
- monitor the state of the network

Pros:

- Reduced overhead by dedicated processor running the protocols.
- Minimized interference between CPU-scheduling of the application and network traffic (in most cases network traffic is asynchronous to the local CPU-scheduling introducing unpredictability by spontaneous interrupts)
- Fault-tolerance mechanisms (Redundancy: message replication, multiple independent networks links, site-fault detection) can be implemented orthogonal to the application

Real-Time (Paradigms) (51c)

Description of a real-time communication protocol system

7 Layer ISO OSI reference model:

1) Physical Layer

- Network medium (copper, fiber, wireless)
- Network topology (bus, ring, star, mesh, point-to-point)
- Bit-encoding

2) Data-Link Layer

- Medium access control
- Logical Link Layer (Frame-encoding, Acknowledgements, Error and Flow Control)

3) Network Layer

4) Transport Layer

5) Session Layer

6) Presentation Layer

7) Application Layer (Semantics of Data : application specific “profiles” (sensor data, actor parameter))

MAC is the decisive factor when designing RT communication protocols with Ethernet as de facto standard for LAN’s

Is Ethernet suitable for Real-Time?

No, because CSMA/CD (Carrier Sense Multiple Access / Collision Detection)
is inherently not time-bounded. Why?

Real-Time (Paradigms) (52)

Principal service of the Medium Access Control Sublayer:

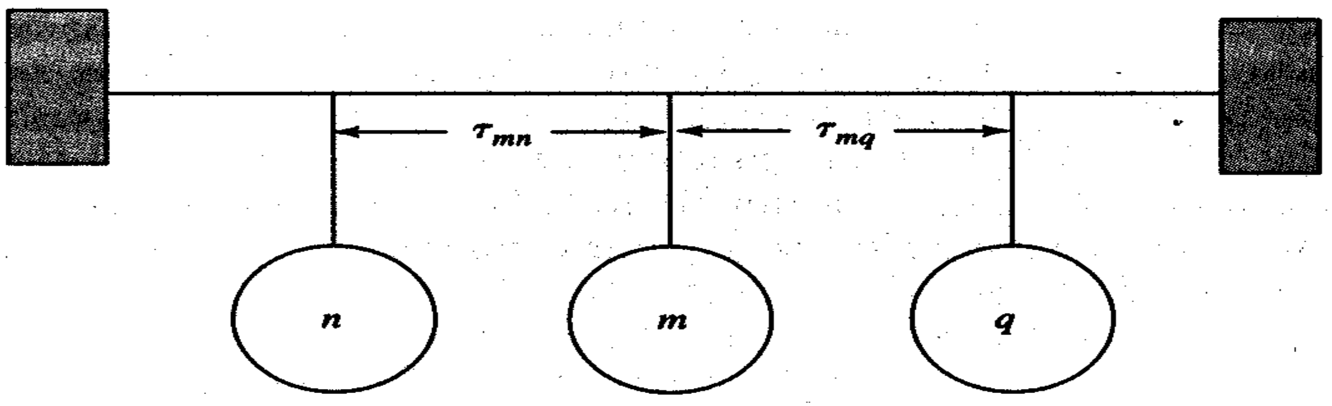
Allocating a single broadcast channel (like Ethernet) among competing users

CSMA protocols

All the nodes can monitor the communication channel and can check whether it is busy or idle (This means carrier sense). If a node observes that the channel is busy, it refrains from sending in order not to interfere (collide).

CSMA is a truly distributed algorithm, with each node deciding when to transmit ---> collisions are still possible, must be detected, but should be avoided ---> CSMA/CD protocols like Ethernet.

Example case of a collision in a bus network



There are various types of CSMA, differing in how to compute the time until trying a send again if the medium is busy.

Real-Time (Paradigms) (53)

Types of CSMA protocols

1-persistent

Behavior:

When a station has data to send, it transmits with a probability of 1 whenever it finds the channel idle. If the channel is busy, the station waits until it becomes idle (greedy approach).

nonpersistent

Behavior:

It differs from 1-persistent w.r.t. the case where the channel is busy.Then, the station deliberately waits a random period of time before sensing the channel again.

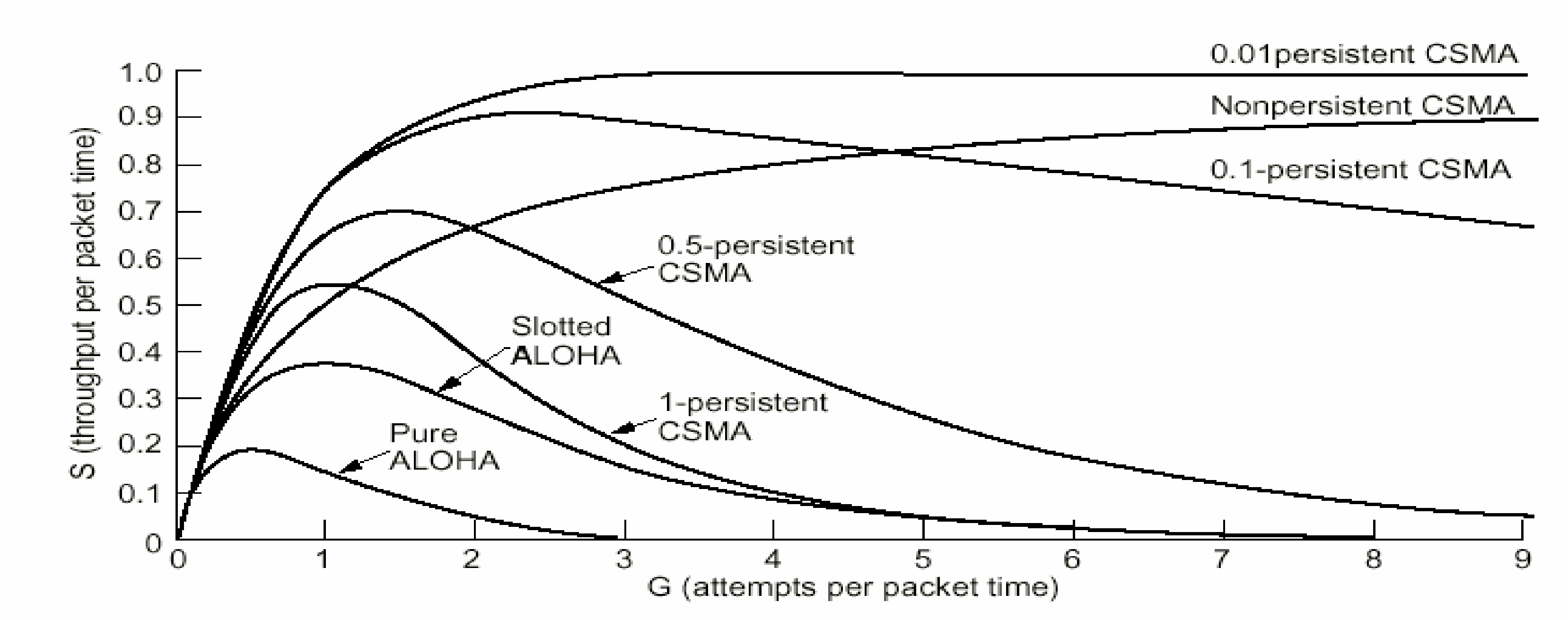
p-persistent (applies to slotted channels)

Behavior:

When a station has data to send, it transmits with a probability of p whenever it finds the channel idle. With a probability of q = 1- p it defers until the next slot and the same procedure iterates. If the channel has become busy meanwhile, the station waits a random time and starts again.If the channel is busy when first sensing it, the station waits until the next slot and repeats the procedure.

Real-Time (Paradigms) (54)

Comparison of channel utilization versus load :



---> CSMA/CD protocols are not suitable for Real-Time because they cannot be time-bounded!

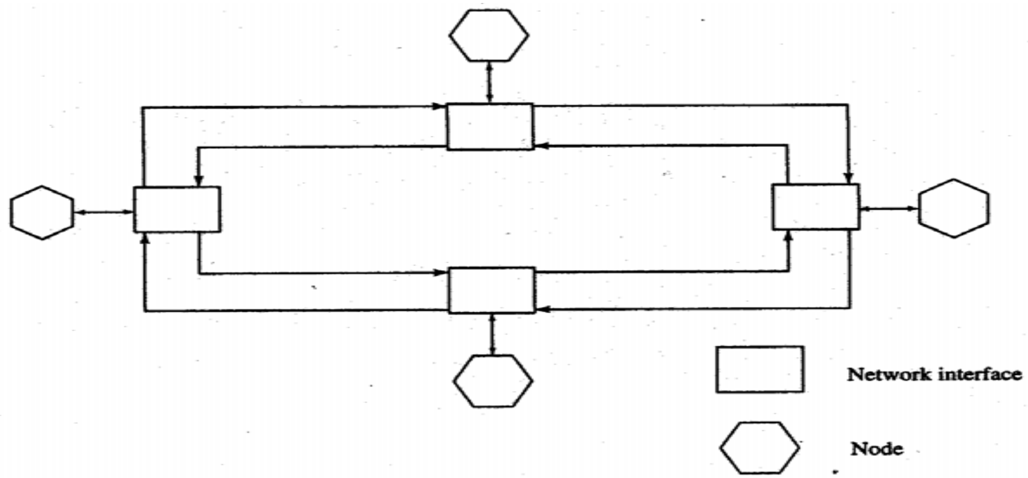
Real-Time (Paradigms) (55)

Examples of real-time communication systems

Timed-Token Protocol

Physical medium: Bus or ring on any medium

Example of a typical ring structure



Basic algorithm for accessing the medium:

- A *token* is a grant of permission to a node to transmit its packets on the network.
- When the token-holding node completes its transmission, it surrenders the token to another node.
- A node is only permitted to transmit on the network if it currently holds the token.
- The token is passed along a logical ring of all participating nodes.

Key parameter TTRT (target token rotation time):

Upper bound for the cycle time of the token which is defined as :

The time it takes for the token to make a complete circuit around the nodes of the ring.

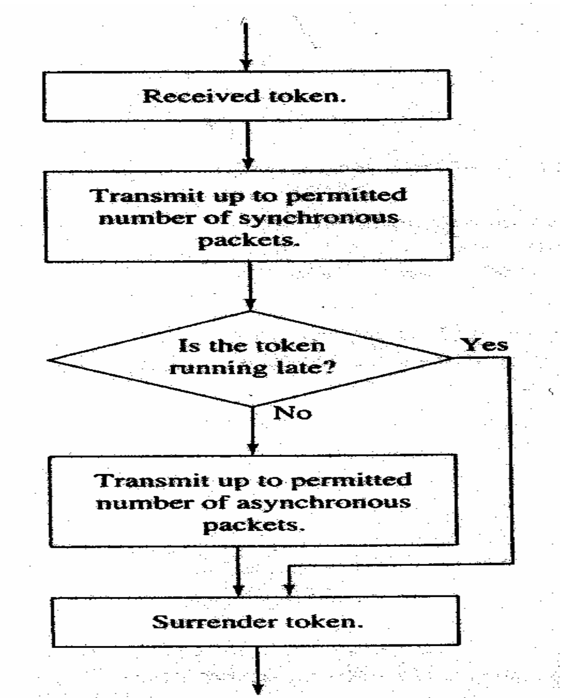
Real-Time (Paradigms) (56)

Basic idea of the protocol:

If each node i (of overall n nodes) holds the token for a fraction f_i of TTRT, with

$$\sum_{i=i}^{n} f_i TTRT \leq TTRT$$

it is guaranteed that each node can send real-time messages up to a certain amount of time (f_i *TTRT)* each TTRT time units.
This is a simple protocol, that fulfils hard real-time requirements and is efficient as long as each node has to send only real-time traffic (constant or bounded rate), i..e. the fraction defined above represents only the real-time traffic.



Extension of the protocol to allow for mixed real-time and non-real-time traffic (“synchronous” and “asynchronous packets”)

- The token is called “late”, if the time since its previous arrival time at the same node (i.e. the actual cycle time) is greater than TTRT.
- The token may become late if the transmission of an asynchronous packet delays the token rotation.

Real-Time (Paradigms) (57)

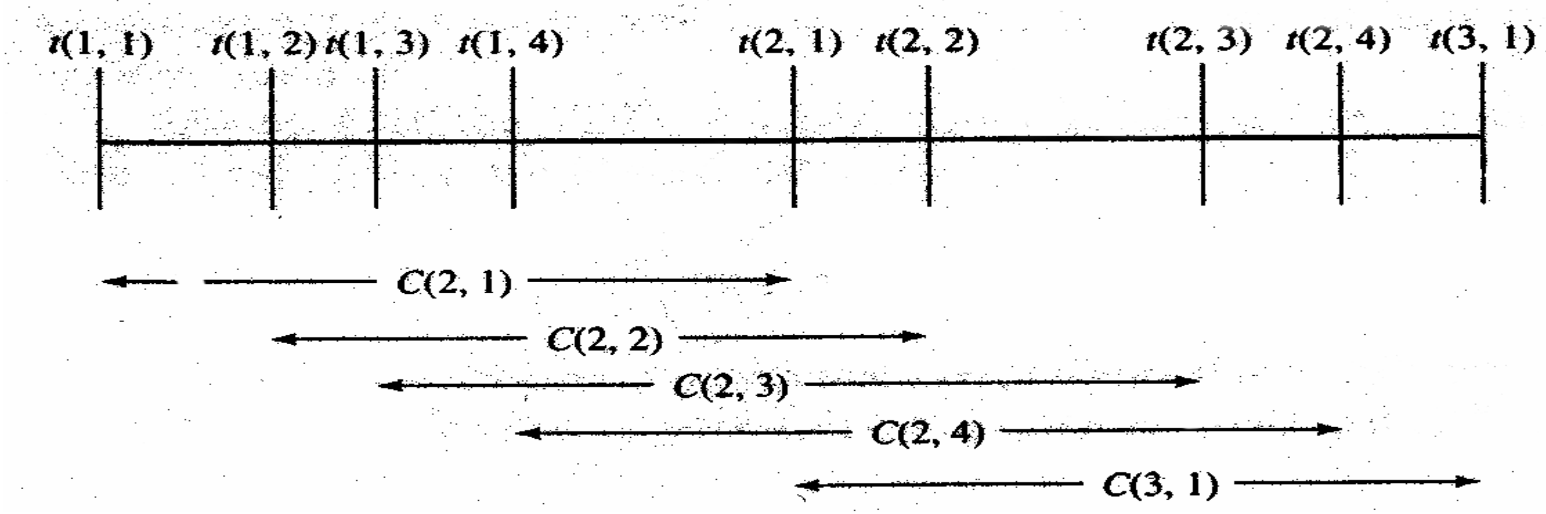
Analysis of the Timed-Token Protocol

Claim: Even with this extension the protocol still provides predictable timing behavior, the token-cycle time is bounded (in contrast to CSMA)

Theorem 1:
In the absence of failures, the maximum cycle time of the token is not greater than twice the TTRT.

Proof:

Let *(a, b)* the *a*th visit of the token to node *b*, *t(a,b)* the time when the token leaves at *(a,b)*, *C(a,b) = t(a,b) - t(a-1, b)* the *a*th token-cycle time as seen by *b*



Real-Time (Paradigms) (58)

S(a, b) the time for transmitting the synchronous packets at *(a, b)*
A(a, b) the time for transmitting the asynchronous packets at *(a, b)*
Case 1: The token was not late during the entire cycle preceding visit *(a, b)*

=> $C(a, b) < TTRT$ ✓

Case 2: The token was always late during the entire cycle preceding visit *(a, b)*

=> during *C(a, b)* only synchronous traffic is transmitted

$$C(a, b) = \sum_{x,y=a-1,b+1}^{a,b} S(x, y)$$

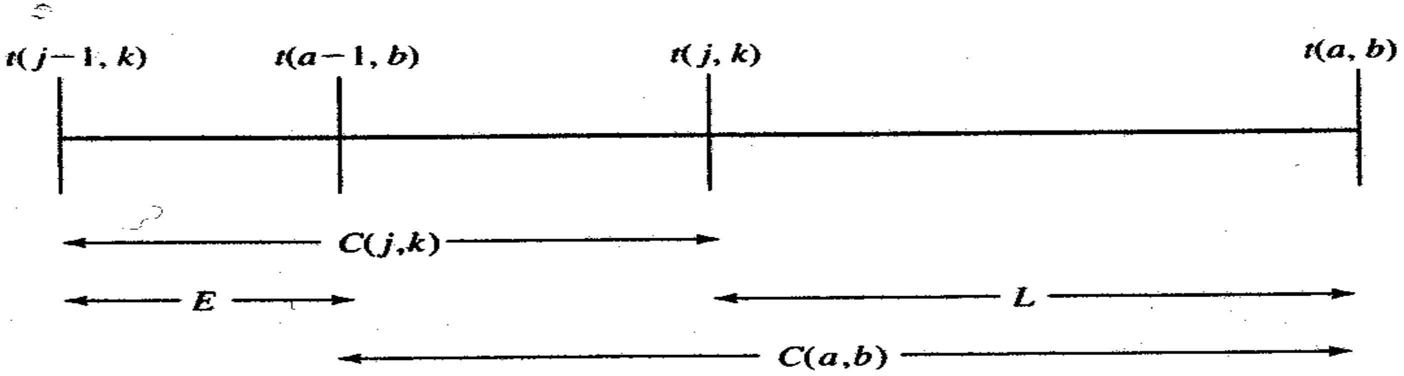
$$\leq \text{TTRT} \sum_{x,y=a-1,b+1}^{a,b} f_i$$

$$\leq \text{TTRT}$$

=> two consecutive cycles cannot have a time greater TTRT ✓

Real-Time (Paradigms) (58b)

Case 3: The token was not late for a part of the cycle preceding *(a, b)*. *(j, k)* was the visit preceding *(a, b)* were the token was not late the last time.



Since the token was early at t(j, k): *C(j, k) <= TTRT*

The token is late over the interval L (only synchronous traffic allowed):

$$C(a,b) = C(j,k) - E + L$$

$$\leq \mathbf{TTRT} - E + \sum_{x,y=j,k}^{a,b} S(x,y)$$

$$\leq \mathbf{TTRT} + \sum_{x,y=j,k}^{a,b} S(x,y)$$

$$\leq TTRT + TTRT \leq 2TTRT \quad \checkmark \qquad \blacksquare$$