ES623 Networked Embedded Systems

Real-Time Communication

6th April 2013

Requirements

- Protocol Latency
- Support for Composability
- Flexibility
- Error Detection
- Physical Structure

Protocol Latency

- Time interval between the start of transmission of a message at the communication network interface (CNI) of the sending node, and the delivery of this message across the CNI of the receiving node
- Latency Jitter -communication protocol should have a predictable and small maximum protocol latency and a minimal jitter
  - application programs in host rely on this *a priori* known predictable latency
  - Any variation in the protocol latency, affects the operation of the application programs adversely.

Protocol Latency

- Simultaneous Delivery in Multicast: standard communication topology in distributed real-time systems is multicast, not point-to-point.
- Same image of an RT entity is needed at a number of different nodes
  - e.g., at the man-machine interface, at a process-model node, and at an alarm-monitoring node.
- A message should be delivered at all receiver CNIs within a short and known time interval.

Support for Composability

- Establishing composability by following two means:
- Temporal Encapsulation of the Nodes: The communication system should erect a temporal firewall around the operation of a host, forbidding the exchange of control signals across the CNI.
- Thus, the communication system becomes autonomous and can be implemented and validated independently of the application software in the host.
- The timing properties of the application software in a temporally encapsulated host can also be validated in isolation.

Support for Composability

- Fulfilling the Obligations of the Client: A server that is implemented in the host of a node can only guarantee its deadlines if the clients fulfill their obligation
  - not to overload the server with too many, or uncoordinated, service requests.
- The communication system should exercise flow control over the requests from the clients, and assist in fulfilling the temporal obligations of the client.

Flexibility

- Communication systems must support different system configurations that change over time
- real-time protocol should be flexible to accommodate these changes without requiring a software modification and retesting of operational nodes that are not affected by the change
  - communication system within a car

Error Detection

- Errors occurring during message transmission must be detected, and should be corrected without increasing the jitter of protocol latency
- If an error cannot be corrected, the sender and all the receivers, must be informed about the occurrence of the error with a low latency
- In real-time system, the detection of message loss by the *receiver* of a message is *of* particular concern
- *Eg: node at* a control valve, receiving output commands from another node.

Error Detection

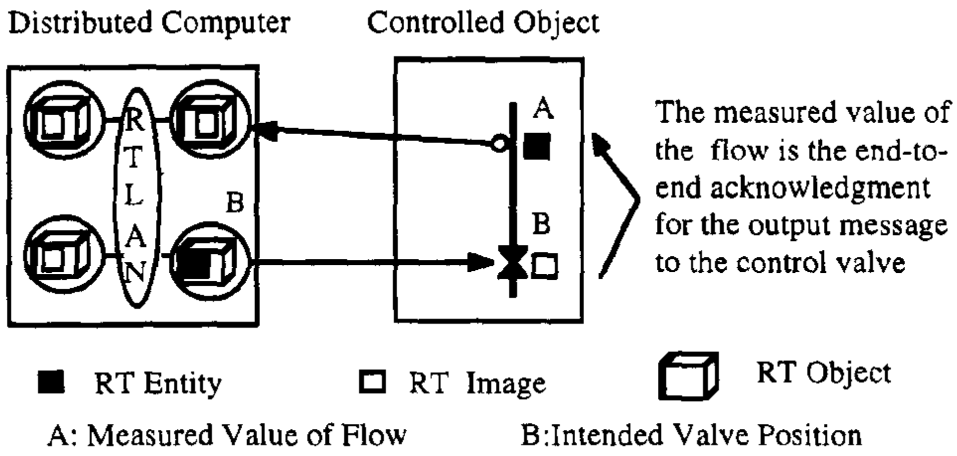
- In distributed real-time system, external electromagnetic interference (EMI), e.g., a flash of lightning, causes the correlated mutilation of all messages on the communication system - *blackout.*
- Blackouts normally last only for a few milliseconds, or for even a shorter period of time.
- The communication system should detect such a blackout, and continue with its operation as soon as the blackout disappears – *blackout management.*
- The failure of a node must be detected by the communication protocol, and must be reported consistently to all the remaining nodes.

End-to-End Acknowledgment

- In a real-time system, the end-to-end acknowledgment about the success or failure of a communication action can arise from a node that is different from the receiver of an output message.
- An output message to an actuator in the environment should cause some effect in the environment.
- This effect is monitored by an independent sensor.
- The results observed by this sensor ensure that the desired action of the message has actually been achieved.
- This is an example of an *end-to-end protocol such as that required at the* interface between the computer system and the controlled object.

End-to-End Acknowledgment

- Avoids false implication
- A proper end-to-end protocol that mechanically sense the actuator position is required to avoid catastrophic false information



Physical Structure

- Multicast communication requirement suggests a communication structure that supports multicasting at the physical level, e.g., a bus or a ring network.
- A fully connected point-to-point communication architecture that provides single-hop broadcasting requires N-1 communication ports at each node, in an ensemble of N nodes.
- For many applications, the high cost of several communication ports at each node, the physical drivers at each node, and the cabling are prohibitive in a point-to-point network.

Bus Versus Ring

- In applications where the physical interconnection is realized by twisted pair wires, a bus structure is more attractive than a ring structure
- Advantage of a bus :
  - simultaneous arrival of a message at all nodes
  - simpler interface
  - better resilience with respect to fail-silent node failures
- If optical fibers form the physical medium, a ring structure is advantageous because the point-to-point connection of fibers is simpler than the construction of a fiber-based bus

Flow Control

- *Control of the speed of information flow between* a sender and a receiver

- In any communication scenario, it is the receiver determines the *maximum speed of communication*

- *explicit flow control and implicit flow control*

Explicit Flow Control

- Receiver sends an explicit acknowledgment message to the sender, informing the sender that the sender's previous message arrived correctly, and that the receiver is now ready to accept the next message
- Receiver can exert *back pressure on the* sender to control the rate of transmission
- The most important protocol with explicit flow control is the well-known **Positive-Acknowledgment-or-Retransmission** (PAR) protocol.

PAR Protocol

- Given a sender, a receiver, a communication medium, a time-out value, and a retry counter
- whenever a sender is asked by its client to send a new message,
  - the sender initializes a retry counter to zero,
  - starts a local time-out interval, and
  - sends the message to the receiver by way of the communication medium.
- When the sender
  - receives an acknowledgment message from the receiver within the specified time-out interval,
  - it informs its client of the successful transmission, and
  - duly terminates.

PAR Protocol

- If the sender does not receive a positive acknowledgment message from the receiver,
  - sender increments the retry counter by one,
  - resends the message, starts the local time-out again, and
  - waits for an acknowledgment message of the receiver.

- If sender does not receive a positive acknowledgment message within the specified time-out interval,
  - sender checks the retry counter to determine whether the given maximum number of retries has already been exhausted.
  - If so, the sender aborts the communication, and
  - informs its client about the failure.

PAR Protocol

- If a new message arrives at the receiver,
  - receiver checks whether this message has already been received.
  - If not, sends an acknowledgment message to the sender, and
  - delivers the message to its client.

  - If the receiver has already received the message,
  - it just sends another acknowledgment message back to the sender.

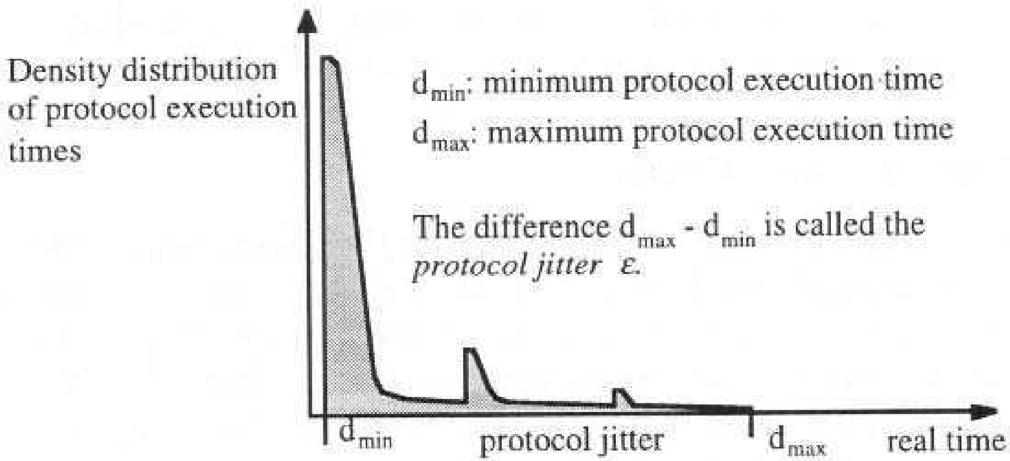
- point in time at which the sender's client is informed about the successful transmission, can be significantly different from the point in time at which the receiver's client accepts the delivery of the message.

PAR Protocol

- Basic principles:
  - Client at the sender's site initiates the communication.
  - Receiver has the authority to delay the sender via the bi-directional communication channel.
  - Communication error is detected by the sender, and not by the receiver.
  - Receiver is not informed when a communication error has been detected.
  - Time redundancy is used to correct a communication error, thereby increasing the protocol latency in case of errors.

PAR Protocol

- typical PAR protocol with a retry counter of 2 is shown.
- In most cases, the first message transmission is successful.
- Therefore, immediately after the minimum protocol latency d_{min}, *there is a peak in* the density distribution of the protocol execution times
- After the second attempt, another increase in the probability of success can be seen, and similarly for the third attempt before the transmission efforts are finally abandoned at d_{max}



Implicit Flow Control

- *sender and receiver agree a priori*
- Requires the availability of a global time-base
- Sender commits itself to send a message only at the agreed points in time, and the receiver commits itself to accept all messages sent by the sender, as long as the sender fulfills its obligation.
- No acknowledgment messages are exchanged during run time.
- Error detection is the responsibility of the receiver, which knows (by looking at its global clock) when an expected message fails to arrive.

Implicit Flow Control

- Fault tolerance can be implemented by active redundancy i.e., sending *k physical copies of every message (if possible by way of different* channels).
- As long as at least one of the *k copies arrives, the communication is* successful.
- Number of messages that must be delivered by the communication system is always constant.
- Communication is unidirectional because there is no need for a return channel from the receiver to the sender.
- Well-suited to multicast communication

Thrashing

- Throughput of a system decreasing abruptly with increasing load
- Example of a traffic system in a large city