

Chapter 7

Real-Time Communication

Feb-08	1
--------	---

UCI
DREAM Lab



Content

- Real-Time Communication Requirements
- Flow Control
- OSI Protocols for Real-Time?
- Fundamental Conflicts in Protocol Design
- Media-Access Protocols
- Performance Comparison: ET .vs. TT
- The Physical Layer

Feb-08	2
--------	---

UCI
DREAM Lab



RT Comm. Requirement

- **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**

– A RT communication protocol should have a predicable and small maximum protocol latency and a minimal jitter

- **Simultaneous Delivery in Multicast**

– The standard communication topology in distributed real-time systems is multicast, not point-to-point

– A message should be delivered at all receiver CNIs within a short and known time interval

For better, at the user components, e.g. Official Release Time (ORT)

Feb-08

3

UCI
DREAM Lab



RT Comm. Requirement (cont)

- **Support for Composability**

– **Temporal encapsulation** of the nodes

- **Elect a temporal firewall around** the operation of a host, forbidding the exchange of control signal across the CNI

- The communication systems become **autonomous** and can be implemented and validated independently of the application software in the host

– **Fulfilling the obligation of the client**

- A server cannot meet its deadlines if the clients overload the server with too many, or uncoordinated, service requests

- The communication system should exercise **flow control** over the requests from the clients, and **assists in fulfilling the temporal obligations of the client** (funny expression ?)



Feb-08

4

UCI
DREAM Lab



RT Comm. Requirement (cont)

- Flexibility
 - The communication system must support different system configurations that change over time
 - E.g., Cars with all different optional functionalities
 - Flexibility is also required to service important sporadic messages, such as an “emergency shutdown” message, with minimal delay

Feb-08

5

UCI
DREAM Lab



RT Comm. Requirement (cont)

- Error Detection
 - The communication system must provide predictable and dependable service
 - Errors occurring during the message transmission must be detected and should be corrected, if possible, without increasing the jitter of the protocol latency
 - Detection of node errors
 - The failure of the node must (too strong ?) be detected by the communication protocol, and must be reported consistently to all the remaining nodes of the ensemble (membership service)

Feb-08

6

UCI
DREAM Lab



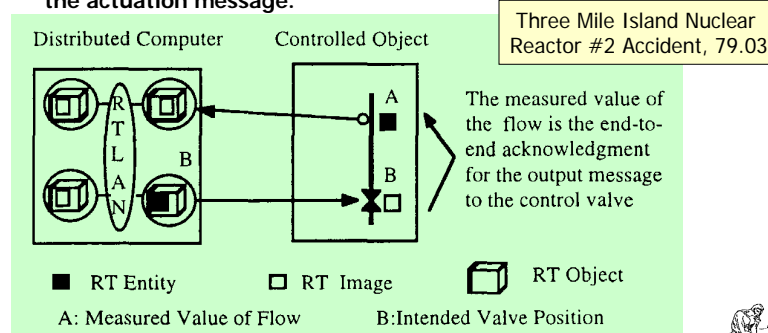
RT Comm. Requirement (cont)

- Error Detection (cont)

- End-to-end acknowledgement

- The end-to-end acknowledgement about the success or failure of a communication action can arise from a node that is different from the receiver of an output message

E.G., a node which senses the impacts in the environment may ack the actuation message.



Feb-08

7

UCI
DREAM Lab



RT Comm. Requirement (cont)

- Physical Structure

- The multicast communication requirement suggests ideally a communication structure that supports multicasting at the physical level, e.g., a bus or a ring network
 - Bus .vs. Ring: Not so clear which one to choose
 - A bus structure is more attractive because of its simple interface and better resilience to with respect to fail-silent node failures. Also, the simultaneous arrival of a message at all nodes is another advantage.
 - A ring structure is advantageous for the communication platforms based on optical fibers (not easy to construct a fiber-based bus)

Feb-08

8

UCI
DREAM Lab



Flow Control

- **Definition**
 - Control of the speed of information flow between a sender and a receiver in such a manner that the receiver can keep up with the sender
 - The receiver determines the maximum speed of the communication
- **Explicit Flow Control**
 - The receiver sends an explicit acknowledgement message to the sender, informing the sender that the sender's previous message arrived correctly
 - The sender is within the sphere of control (SOC) of the receiver, i.e., the receiver can exert back pressure on the sender to control the rate of the transmission

Feb-08

9

UCI
DREAM Lab



Explicit Flow Control (cont)

- **PAR (Positive-Acknowledgement or Retransmission) protocol**
 - **The Sender**
 - Set `retry-counter = 0`, start a local time-out interval, send out the message to the receiver
 - If a positive ACK is received within the time-out, the last transmission is regarded as a successful one
 - If no ACK within the time-out
 - (1) If `retry-counter <= max-num-retry`, increase the `retry-counter` by 1, resend the message, reset the time-out value
 - (2) If `retry-counter > max-num-retry`, declare the last transmission fails
 - **The Receiver**
 - Send an ACK to the sender no matter whether the message has been received before or not

Feb-08

10

UCI
DREAM Lab



Explicit Flow Control (cont)

- PAR (Positive-Acknowledgement or Retransmission) protocol

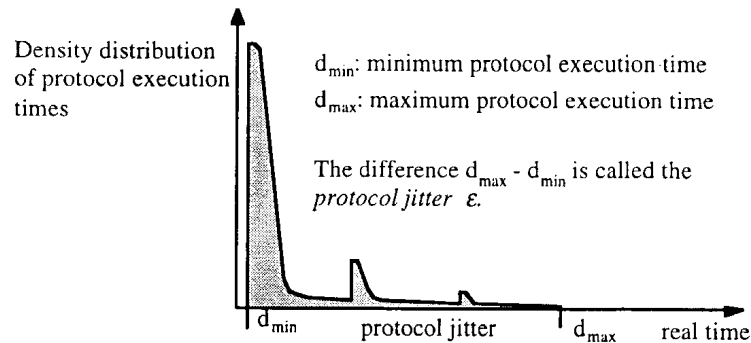


Figure 7.2: Distribution of latencies of a typical PAR protocol.

Feb-08

11

UCI
DREAM Lab



Implicit Flow Control

- The sender and the receiver **agree a priori**, i.e., at system start up,
on the points in time when messages are sent
 - Requires a **global time-base**
- The sender commits itself to sending a message only at the agreed points in time, and the receiver commits itself to accepting all messages sent by the sender, as long as the sender fulfills its obligations
- **Error detection is the responsibility of the receiver**, which knows when an expected message fails to arrive

Feb-08

12

UCI
DREAM Lab



Flow Control - Thrashing

- Real-time systems must be free of the thrashing phenomena

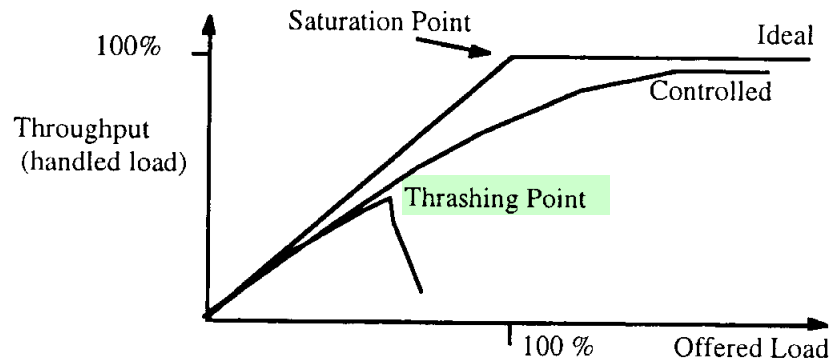


Figure 7.3: Throughput-load characteristic.

Feb-08

13

UCI
DREAM Lab



Flow Control - Thrashing

- Mechanisms causing thrashing
 - Mechanisms that require a more than proportional increase in resources as the local increases, are prone to cause thrashing
 - Examples
 - Retry mechanism in the PAR protocol:
When a comm system slows down due to the heavy load, a high-level PAR protocol generates additional load.
 - Operating system: scheduling & queue management:
When the scheduling overhead becomes high due to the heavy load, the overhead becomes additional load.
 - Technique to avoid thrashing in explicit flow-control
 - Monitor the resource requirements of the system continuously and to exercise a **stringent back-pressure flow control** as soon as a decrease in the throughput is observed

Feb-08

14

UCI
DREAM Lab



Flow Control in Real-Time Systems

I do not see the point listing this here

- The following table suggests implicit flow control is better suited for real-time systems

Characteristic	Explicit flow control	Implicit flow control	Hard real-time system
Control signal	Receiver must be able to control the send event of the sender	Control signals are generated by the progression of real-time at a constant rate	Receiver cannot fully control the events in the sphere of control of the sender
Error detection	At the sender	At the receiver	At the receiver
Prone to thrashing	Yes	No	Thrashing must be avoided
Multicast	Difficult	Yes	Multicast required

- When a number of lightning strikes hit the power lines, a monitoring & control system for the electric power grid cannot exercise explicit flow control over the alarms to avoid or contain the thrashing.

Feb-08

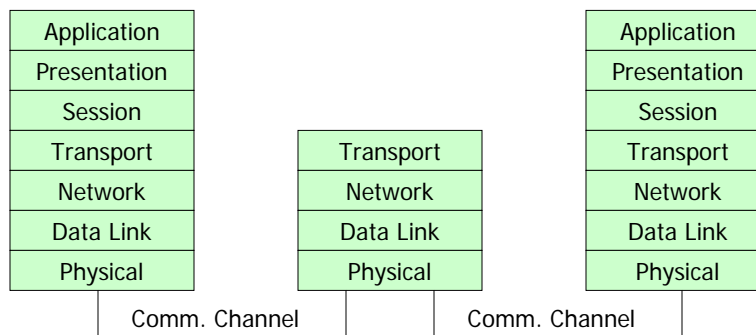
15

DREAM Lab



OSI Protocols for Real-Time ?

- The OSI Reference model



- The frequent assumptions in this model:
 - Comm protocols are of the PAR type.
 - Latency and latency jitter are not issues.

Feb-08

16

UCI
DREAM Lab



Asynchronous Transfer Mode (ATM)

- Developed to provide real-time communication with low jitter over broadband networks
- The information is packed into **ATM cells**, and the header contains control information for the identification and routing of the packet
- Well suited to provide basic communication services for wide area real-time systems

Feb-08 17

UCI
DREAM Lab



ATM (cont)

Cell (53 Bytes)					
Header (5 Bytes)					Information
Generic Flow Control	Channel Identifier	Payload Type Indicator	Cell Loss Priority	Header Checksum	Payload
4 bits	24 bits	2 bits	2 bits	8 bits	48 bytes

Feb-08 18

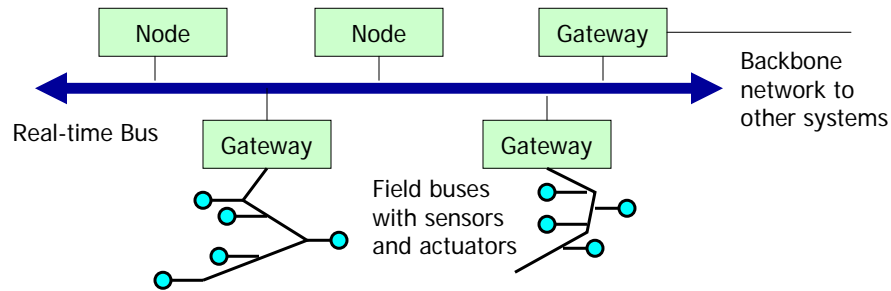
UCI
DREAM Lab



Real-Time Communication Architecture

- In many real-time applications, three different types of communication networks are distinguished
 - Field bus, Real-time network, and Backbone network

The field bus and the real-time network must provide guaranteed temporal performance



Feb-08 19

UCI
DREAM Lab



RT Communication Architecture (cont)

- Field Bus
 - To interconnect a node of the distributed computer systems to the sensors and actuators in the controlled object
 - Field bus messages have short data field, containing state data, typically two bytes in length, and are transmitted periodically with strict real-time requirements for latency and latency jitter
 - Precise clock synchronization should be provided at the field bus level
 - Fault tolerance is not a major issue.
 - The main concern at the field bus level is low cost

Feb-08 20

UCI
DREAM Lab



RT Communication Architecture (cont)

- **Real-time Network**
 - Reliable and temporally predictable message transmission with **low latency** and **minimal latency jitter**
 - Replicated communication channels
 - Support for fault tolerance to handle **replicated nodes** and **replicated communication channels**
 - **Clock synchronization** in the range of microseconds is desirable.
 - **Membership service** with **low latency for detecting node failures**
- **Backbone network**
 - Exchange of **non-time-critical information** between the real-time cluster and data-processing systems of an organization

Feb-08 21

UCI
DREAM Lab



RT Communication Architecture (cont)

Service characteristic	Field bus	Real-time network	Backbone network
Message semantics	State	State	Event
Latency/jitter control	Yes	Yes	No
Data field length	1-6 bytes	6-12 bytes	> 100 bytes
Clock sync.	Yes	Yes	Optional
Fault tolerance	Limited	Yes	Limited
Membership service	Maybe	Yes	Maybe
topology	Multicast	Multicast	Point-to-point
Comm. control	Multi-master	Distributed	Central/dist.
Flow control	Implicit	Implicit	Explicit
Low cost	Very important	Important	Not important

Feb-08 22

UCI
DREAM Lab



Fundamental Conflicts in Protocol Design

- **External Control vs. Composability**
 - Composability in the temporal domain requires
 - CNI of every node is fully specified in the temporal domain
 - The **integration** of a set of nodes into the complete system does **not lead to any change** of the temporal properties of the **individual CNIs**
 - The temporal properties of every host can be tested in isolation with respect to the CNI
 - In an event-triggered system, the temporal control signals originate external to the communication system, in the hosts of the nodes.
 - Not possible to achieve low-level temporal composability

Abstract

Feb-08

23

UCI
DREAM Lab



Conflicts in Protocol Design (cont)

- **Flexibility vs. Error Detection**
 - Flexibility implies that the behavior of a node is not restricted a priori
 - Error detection is only possible if the actual behavior of a node can be compared to some a priori knowledge of the expected behavior
- **Sporadic data vs. Periodic data**
 - A real-time protocol cannot be effective in both the transmission of the periodic data and the transmission of sporadic data !
 - Periodic data must be transmitted with **minimal latency jitter**
 - Sporadic data must be transmitted with **minimal delay**

Feb-08

24

UCI
DREAM Lab



Conflicts in Protocol Design (cont)

- **Single Locus of Control** vs. **Fault Tolerance**
 - Any protocol that relies on a single locus of control has a single point of failure but --- not always cost-effective to avoid it completely
- **Probabilistic Access** vs. **Replica Consistency**
 - Replica consistency (needed if active redundancy is to be implemented) cannot be easily realized with medium access based on probabilistic mechanisms !

Feb-08

25

UCI
DREAM Lab



Media Access Protocols

Characteristics of a communication channel

- **Bandwidth**
 - # of bits that can traverse a channel in unit time
 - E.g., 10 kbit/sec, 1Mbit/sec
- **Propagation delay:**
 - Time interval it takes for a bit to travel from one end of the channel to the other end
 - The term **bit length** of a channel is to denote the # of bits that can traverse the channel within one propagation delay
- **Limit to protocol efficiency**
 - In a bus system, the protocol efficiency is limited by the need to maintain a minimum time interval of one propagation delay between two successive messages
 - i.e., **data efficiency $< m / (m + bl)$** , where m is the message length, and bl is the **bit length**

Feb-08

26

UCI
DREAM Lab



Media Access Protocols (cont)

- CSMA/CD-LON
 - Example: Ethernet (collision detection)
 - LON:
 - An example of CSMA/CD targeted for real-time systems
 - Rely on a random number generator to choose a delay at the start of the transmission to reduce the possibility of collisions and during retransmissions as a result of collisions.
 - The size of the randomizing window is a function of the load on the channel and is designed to minimize the probability of a collision under high load.

Feb-08 27

UCI
DREAM Lab



Media Access Protocols (cont)

- CSMA/CA-CAN (collision avoidance)
 - Use bit arbitration to avoid collision
 - The **dominant state** can overwrite the **recessive state** on the channel.
 - The **propagation delay of the channel is smaller than the length of a bit cell.** -- Maybe a requirement to support the above.
 - The **msg priority is determined by the msg identifier.**
 - A node (msg identifier) with all winning-bits (e.g., all '0's) has the highest priority.

Arbitration	Control	Data field	CRC	A	EOF
11	6	0-64	16	2	7

Data format of a CAN message

Feb-08 28

UCI
DREAM Lab



Media Access Protocols (cont)

- **Token Bus**
 - Two time parameters determine the response of a token-bus system
 - **Token-hold time (THT)** and **Token-rotation time (TRT)**, which is the longest time for a full rotation of the token
- A serious error in any token system is the loss of the token
- Profibus is an example of a token bus protocol for real-time systems

Feb-08

29

UCI
DREAM Lab



Media Access Protocols (cont)

- **Minislotting-ARINC 629**
 - A time-controlled medium access strategy, where the time is partitioned into a sequence of mini-slots, each longer than the propagation delay of the channel
 - Every node is assigned a unique number of mini-slots that must elapse, with silence on the channel, before it is allowed to transmit
 - **Phase 1:** a set of processes wanting to transmit a msg is admitted into a “**distributed waiting room**”.
 - **Phase 2:** an epoch: all processes in the waiting room are allowed to transmit their msgs before any new process is allowed to enter the waiting room.

Feb-08

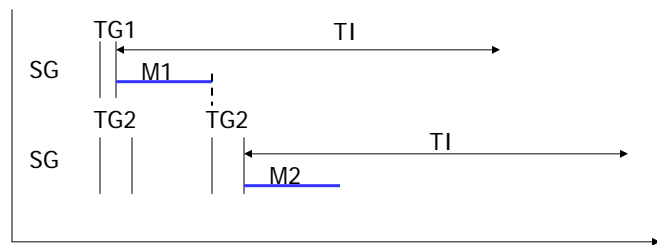
30

UCI
DREAM Lab



Media Access Protocols (cont)

- SG: **Synchronization gap** controlling the entrance to the waiting room.
 - Identical for all nodes.
- TG: **Terminal gap** controlling the access to the bus
- TI: **Transmit interval** disabling a host from monopolizing the channel.
 - Identical for all nodes.
- $SG > \text{Max}\{ TG_i \}$ and $TI > SG$



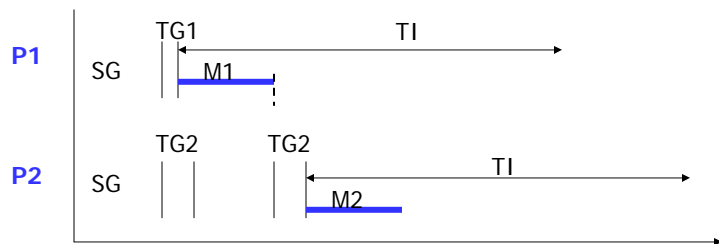
Feb-08 31

UCI
DREAM Lab



Media Access Protocols (cont)

- Both P1 and P2 are admitted into the waiting room after observing the silence of the channel for SG.
- Since P1 has a shorter TG, TG1, it starts transmitting after observing the silence of the channel for TG1.
- After P1 has completed the transmission, it is disabled from transmitting for the period of $TI - M1$.



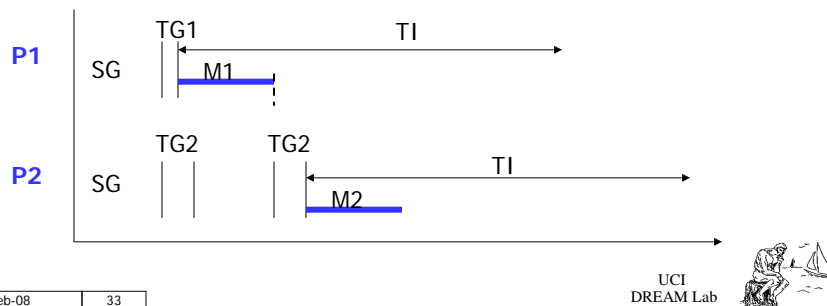
Feb-08 32

UCI
DREAM Lab



Media Access Protocols (cont)

- After P1 has completed the transmission, P2 must wait for TG2 again before starting to transmit.
- Other processes cannot enter the waiting room since $SG > \text{Max}\{ TG_i \}$
- Typical values for a 2 Mbps channel
 - TG: 4 – 128 microsec
 - TI: 0.5 – 64 millisecc



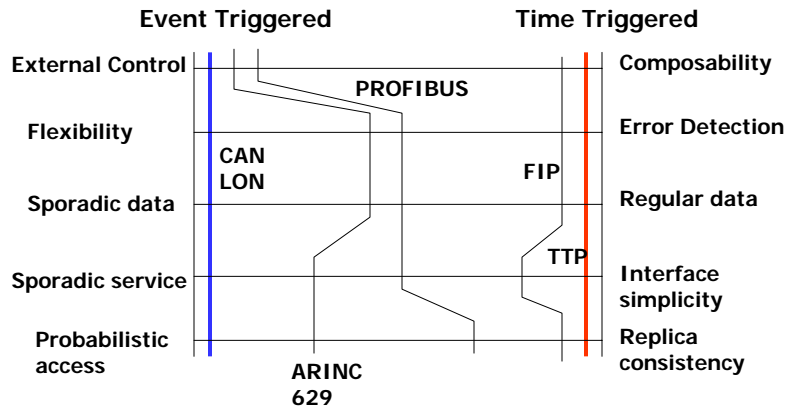
Media Access Protocols (cont)

- **Central Master - FIP**
 - Rely on a central master (called the bus arbitrator) to control the access to the bus
 - The master periodically broadcasts the name of a variable (containing the name and the period) from the static list on the bus.
 - The node that holds this variable responds with a broadcast of the contents of the variable.
 - If the central master node fails, another node takes over the role of the central master
- **TDMA - TTP**
 - **TDMA round:** A sequence of sending slots which are used by an ensemble of nodes and repeated with some periodicity
 - The sequence of all different TDMA rounds is called a **cluster cycle**.

Feb-08 34

UCI DREAM Lab

Media Access Protocols (cont)



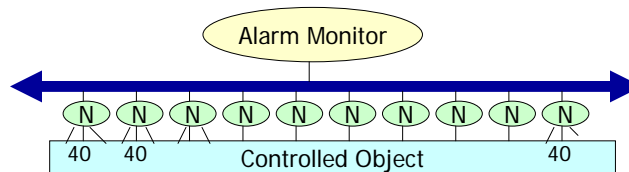
Feb-08 35

UCI
DREAM Lab



Performance Comparison: ET versus TT

- The performance
 - ET protocol > TT protocol : if there are exchanges of many sporadic messages with unknown request time
 - ET protocol < TT protocol : if there are exchanges of many periodic messages
- Problem specification
 - Consider a cluster consisting of 10 interface nodes connected to the controlled object, and one alarm monitoring node that processes the alarms and displays them to the operator
 - Each interface node must observe 40 binary alarm signals in the controlled object
 - Operator must know the change of an alarm signal within 100msec
 - The communication channel supports a bandwidth of 100kbits/second



Feb-08 36

UCI
DREAM Lab



Performance Comparison: ET versus TT

- Both ET and TT solutions based on the basic protocol, e.g., a CAN protocol
- Event-Triggered implementation
 - Sends an **event message** to the operator as soon as an alarm has been recognized
 - Consider a CAN message with a data field of **1 byte**
 - Consider the overhead of a CAN message is 44bits and an inter-message gap is 4bits
 - ➔ The total length of an alarm event message is **56bits**
 - Bandwidth of 100kbits/sec
 - ➔ about **180 messages** in the given latency interval of 100msec
 - ➔ clearly **lower than the peak load** scenario of 400 messages (40 signals from each of the 10 nodes)

Feb-08

37

UCI
DREAM Lab



Performance Comparison: ET versus TT

- Time-Triggered implementation
 - Every node sends a **periodic state message** every 100msec
 - CAN message with a data field of **40bits, one bit for each alarm**
 - **Synchronous operations enable this**
 - Consider the overhead of 44bits and the inter-message gap of 4bits
 - ➔ total state message length is **88bits**
 - Bandwidth of 100kbits/sec
 - ➔ about **110 state messages** in the given latency interval of 100msec
 - Only **10 periodic state messages** are needed to cover the specified peak-load scenario
 - ➔ **10% of the available bandwidth**
 - TT implementation provides an error detection capability at the alarm monitor

Feb-08

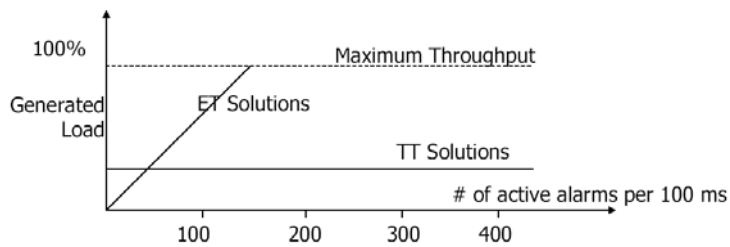
38

UCI
DREAM Lab



Performance Comparison: ET versus TT

- Comparison of the solutions
 - Break-even point is at about **16 alarms per 100msec** (4% of the peak load)
 - **Less than 16 alarms** within a time interval of 100msec → **ET generates less load**
 - **More than 16 alarms** within a time interval of 100msec → **TT generates less load**



Feb-08

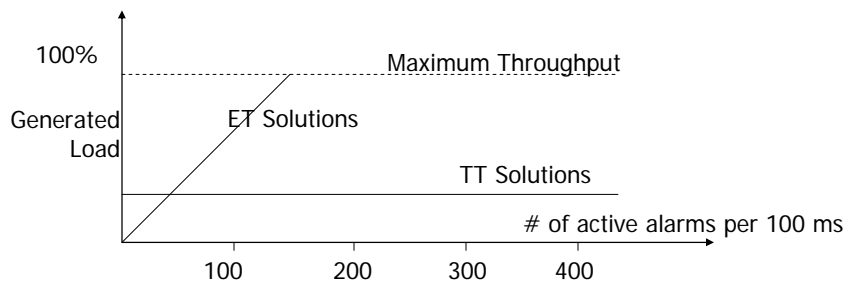
39

UCI
DREAM Lab



Performance Comparison: ET versus TT

- A question on the comparison of ET and TT implementations
 - Is it possible to exploit the opportunities for packing multiple alarm signal detections into one report in ET implementations ?



Feb-08

40

UCI
DREAM Lab



Physical Layer

- The physical layer specifies the transmission codes
 - Coding of the bit pattern
 - Transmission medium
 - Transmission speed
 - Physical shape of the bit cells
- Properties of Transmission Codes
 - Asynchronous communication
 - Receiver synchronizes its receiving logic with that of the sender **only at the beginning** of a new msg.
 - Clocks of the receiver and the sender drift apart → Msg length is limited, e.g., 10 bits in UART.
 - Synchronous communication
 - Receiver **resynchronizes** its receive logic during the reception of a msg **to the ticks of the sender's clock**.
 - Possible only if the data encoding guarantees frequent transitions in the bit stream. → Synchronizing code

Feb-08 41

UCI
DREAM Lab



Physical Layer (cont)

- Examples of Transmission Codes
 - NRZ (Non-return-to-zero) code
A non-synchronizing code

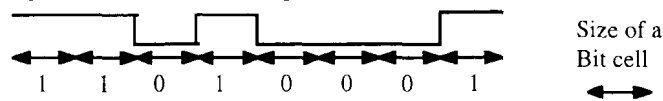


Figure 7.14: Encoding of the bit sequence "1101 0001" in the NRZ code.

- Manchester code
"0": high-low bit cell; "1": low-high bit cell

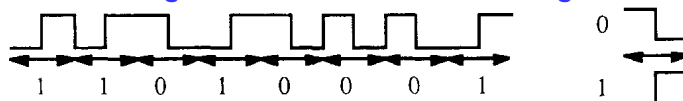


Figure 7.15: Encoding of the bit sequence "1101 0001" in the Manchester Code.

Feb-08 42

UCI
DREAM Lab



Physical Layer (cont)

- Modified Frequency Modulation (MFM)
 - "0": No signal change; "1": Signal change;
 - "00": Signal change at clock points

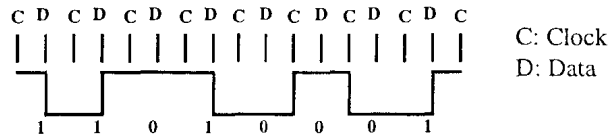


Figure 7.16: Encoding of the bit sequence "1101 0001" in MFM.

Feb-08

43

UCI
DREAM Lab



Physical Layer (cont)

- Signal shape
 - Steep edges lead to high-frequency EMI

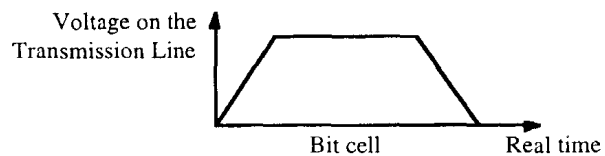


Figure 7.17: Form of a feature element to reduce EMI.

Feb-08

44

UCI
DREAM Lab

