

# EE445M/EE380L.12 Embedded and Real-Time Systems/ Real-Time Operating Systems

## Lecture 10: Networking, CAN bus, Ethernet, Embedded Internet, TCP/IP

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

1

## Network

- Collection of interfaces that share a physical medium and data protocol
  - Communicate & synchronize tasks across computers
  - Distributed computing

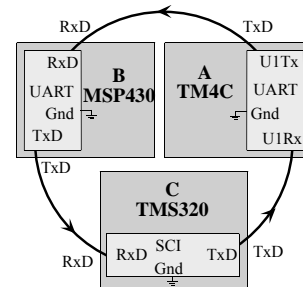
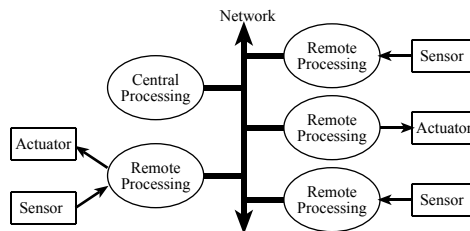


Figure 9.1. A simple ring network with three nodes, linked using the serial ports.

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

2

# Network Topology

- Ring (previous slide)
- Shared bus
- Star
- Mesh
- ...

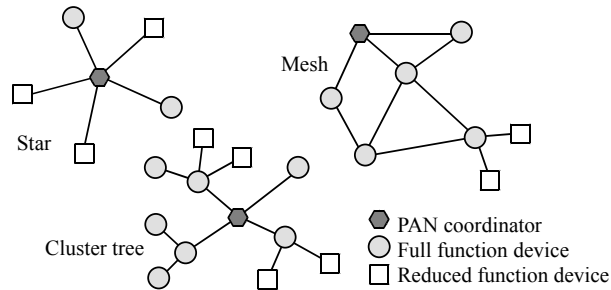


Figure 9.2. ZigBee wireless networks communicate by hopping between nodes.

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

3

# Layered Approach

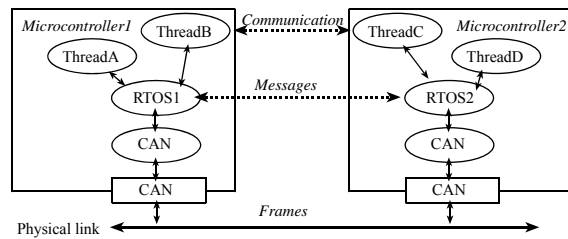


Figure 9.3. A layered approach to communication systems.

- Physical layer (frames)
- OS layer (message & packets)
- Application layer (sessions)

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

4

## Framing and Messaging



Figure 9.19. An Ethernet frame can hold 46 to 1500 bytes.

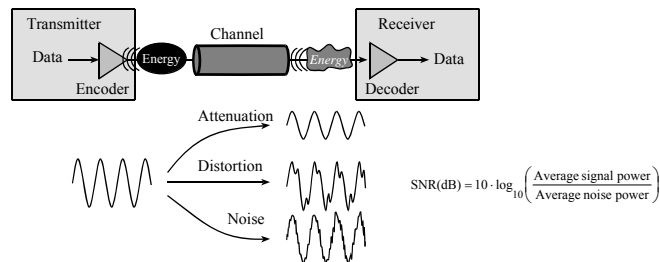
1. Address information field
  - Physical address specifying destination/source computer
  - Logical address specifying destination/source processes
2. Synchronization or handshake field
  - Physical synchronization, e.g. shared clock, start/stop bits
  - OS/process synchronization, e.g. request/ack, semaphore
3. Data field
  - ASCII or binary (raw or compressed)
4. Error detection and correction field
  - Parity, checksum, error correcting codes (ECC)

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

5

## Physical Layer



- Shannon channel capacity
 
$$C = W \log_2 (1 + \text{SNR}) \text{ b/s}$$
- Example: Telephone channel
 
$$W = 3.4\text{kHz} \text{ and } \text{SNR} \sim 38 \text{ dB} \Rightarrow \text{SNR} \sim 6310$$

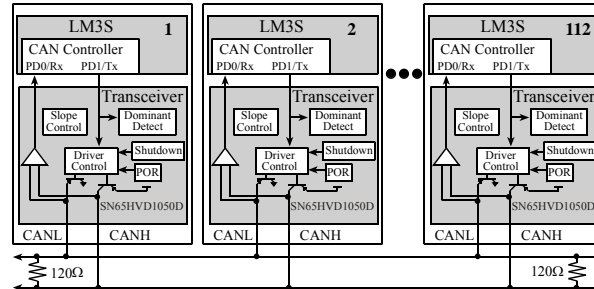
$$C = 3.4\text{kHz} \log_2 (6311) = 3.4\text{kHz} \times 12.62 \text{ b/s} = \underline{42.9 \text{ kb/s}}$$

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

6

# Controller Area Network (CAN)



- Originally for use in automobiles (car area network)
- High-integrity serial communications
- Real-time applications
- Up to 1 Mbits/second (@ max. 40m), up to 112 nodes
- Half duplex (both directions, but only one at a time)

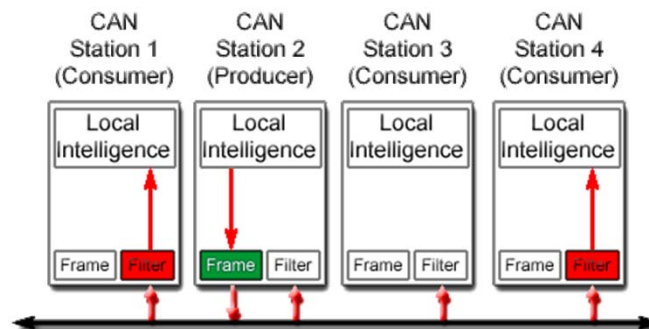
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

7

# CAN Bus

- Shared bus topology
  - Broadcast operation



Source: <http://can-cia.org/index.php?id=can>

© 2002, GWW in Automation - TS

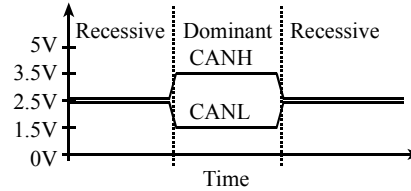
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

8

## CAN Signaling

- Wired-AND logic
  - Similar to wired-OR or open collector logic



- Dominant state is logic low (0)
- Recessive state is logic high (1)

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

9

## CAN Protocol

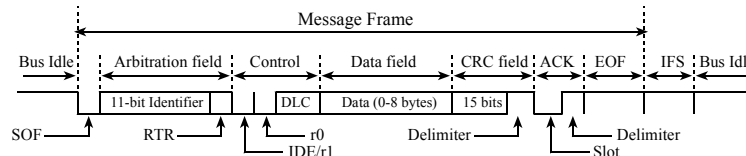


Figure 9.7. CAN Standard Format Data Frame.

- Data, Remote, Error or Overload frames
  - Arbitration Field
    - 11-bit identifier specifies data type (not address), e.g. Ping, IR, ...
    - RTR=IDE=0 means 11-bit standard format data frame
  - Control Field
    - DLC specifies the number of data bytes (0 to 8)
  - Data Field
    - Contains zero to eight bytes of data
  - CRC Field
    - 15-bit checksum used for error detection

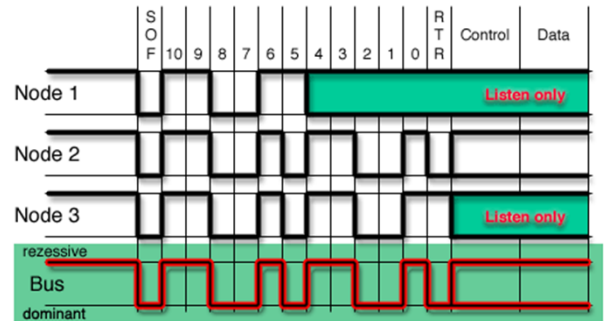
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

10

## CAN Arbitration

- Bit-wise arbitration
  - Priority handled by dominant win over recessive
    - Lower IDs are higher priority



Source: <http://can-cia.org/index.php?id=can>

© 2002, CAN in Automation - TS

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

11

## CAN Bandwidth

$$\text{Bandwidth} = \frac{\text{number of information bits/frame}}{\text{total number of bits/frame}} \cdot \text{baud rate}$$

- Number of bits in a CAN message frame
  - ID: 11 or 29 bits
    - Standard CAN 2.0A vs. extended CAN 2.0B
  - Data: 0, 8, 16, 24, 32, 40, 48, 56, or 64 bits
  - Remaining components: 36 bits
    - SOF (1)
    - RTR (1)
    - IDE/r1 (1)
    - r0 (1)
    - DLC (4)
    - CRC (15)
    - ACK/EOF/intermission (13)

**How many bits in a frame:**

Standard CAN 2.0A frame with 4 data bytes?

Extended CAN 2.0B frame with 8 data bytes?

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

12

## CAN Synchronization

- Synchronous bit transmission
  - Where is the clock? (Answer: in the data)
  - Needs edges so the receiver can synchronize
    - A long sequence of 0's or a long sequence of 1's
- Bit stuffing
  - Insert complementary bit after five equal bits
    - CAN 2.0A may add 3+n stuff bits  
(n is number of bytes)
    - CAN 2.0B may add 5+n stuff bits
  - Receiver has to un-stuff

Lecture 10

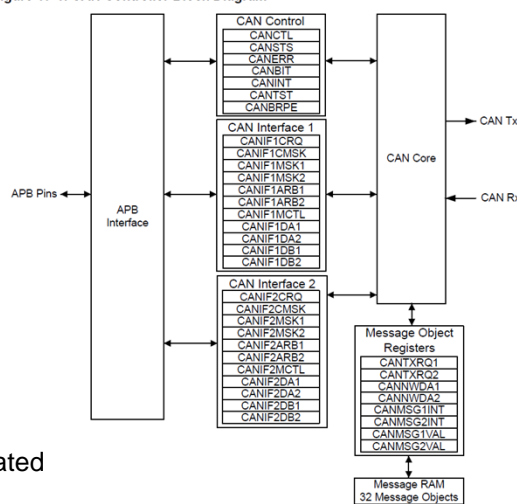
J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

13

## TM4C123 CAN Modules

Figure 17-1. CAN Controller Block Diagram

- 2 CAN Modules
  - Message RAM
    - 32 Msg. Objects
  - Message Objects
    - Tx Objects
      - One-time
      - Remote request
    - Rx Objects
      - Receive filters
      - Can be concatenated into FIFOs



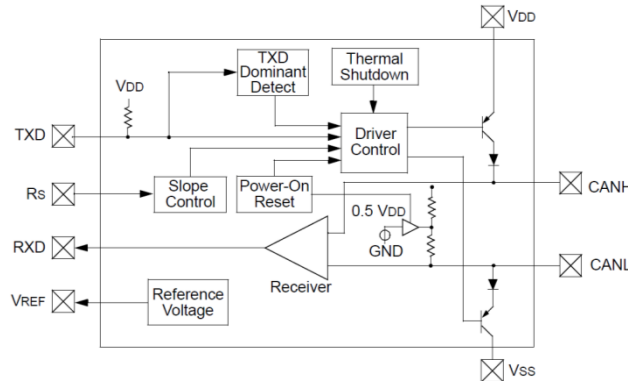
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

14

# Microchip MCP2551

- CAN Transceiver



<http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010405>  
<http://ww1.microchip.com/downloads/en/DeviceDoc/21667f.pdf>

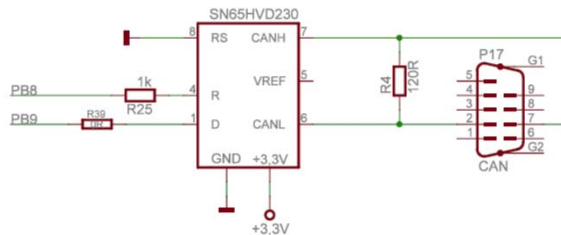
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

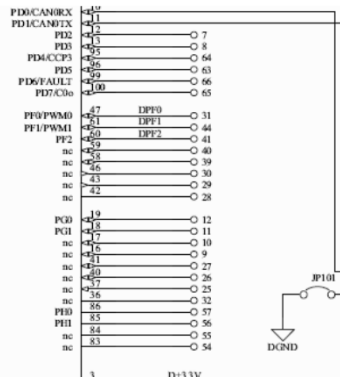
15

# CAN Interfacing

There must be a 120Ω resistor on each end of the CAN cable, and no resistor on middle nodes.



A transmission line if  $L > \lambda/4$ ,  $\lambda = v/f \approx v \tau$   
 Slew rate =  $25V/\mu s$ ,  $v = VF \cdot c = 2 \cdot 10^8$  m/s  
 $1V$  in  $40$  ns,  $\lambda = 2 \cdot 10^8$  m/s  $\cdot$   $40 \cdot 10^{-9}$ s =  $8$ m,  $\lambda/4 = 2$ m



Lecture 10

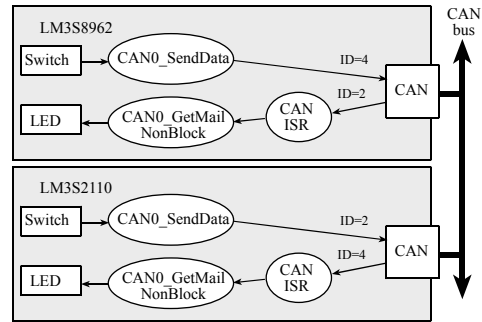
J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

16



# Simple CAN Network

- TivaWare driver
  - CANInit
  - CANBitRateSet
  - CANEnable
  - CANIntEnable
  - CANMessageSet
  - CANIntStatus
  - CANStatusGet
  - CANMessageGet



CAN\_4C123.zip

```
CANBitRateSet(CAN0_BASE, 8000000, CAN_BITRATE);
```

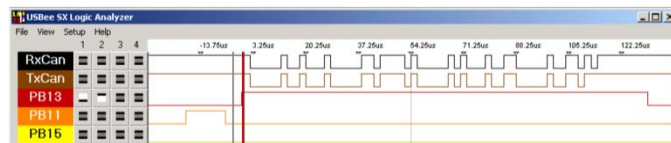
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

17

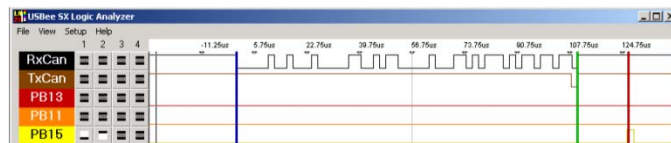
# Logic Analyzer Traces

- Transmit Frame



Why did it take 130µs to execute CAN\_Transmit?  
Why does the RxCan have more stuff than the TxCan?

- Receive Frame



What is the total number of bits in this frame?  
Why did it take 110µs to complete an entire frame?  
What is that blimp on TxCan?  
Where is the end of the frame?  
What is the bandwidth?

Lecture 10

18

## Synchronization Issues

- How to connect transmitter/receiver threads?
  - How to start, handshake
  - Race conditions
- How to prevent streaming data from stalling?
  - Priority, buffer size

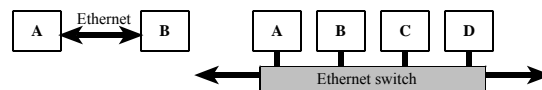
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

19

## Ethernet

- Bus-based topology
  - Hub (repeater) or switch



- Physical layer
  - Many variants evolved over the years

| Pin | Color        | Pair | Description |
|-----|--------------|------|-------------|
| 1   | white/orange | 2    | TxData +    |
| 2   | orange       | 2    | TxData -    |
| 3   | white/green  | 3    | RecvData +  |
| 4   | blue         | 1    | Unused      |
| 5   | white/blue   | 1    | Unused      |
| 6   | green        | 3    | RecvData -  |
| 7   | white/brown  | 4    | Unused      |
| 8   | brown        | 4    | Unused      |

Table 9.2. Pin assignments on a 568-B Ethernet connector.

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

20

# Ethernet Cabling

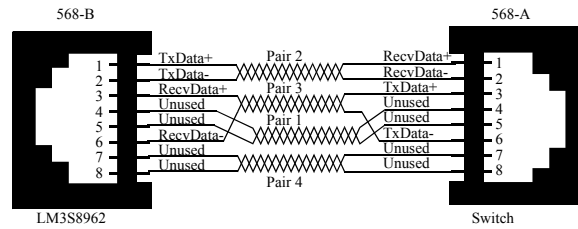


Figure 9.15. Ethernet cable between a microcontroller and a switch.

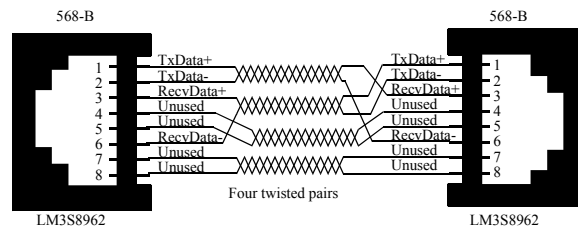


Figure 9.16. Ethernet cross-over cable between two microcontrollers.

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

21

# Ethernet Protocol

|            |           |                     |                |                 |                    |            |
|------------|-----------|---------------------|----------------|-----------------|--------------------|------------|
| Preamble   | SFD       | Destination Address | Source Address | Length/<br>Type | Data               | FCS        |
| 7<br>Bytes | 1<br>Byte | 6<br>Bytes          | 6<br>Bytes     | 2<br>Bytes      | 46 - 1500<br>Bytes | 4<br>Bytes |

- Ethernet Frame
  - Synchronization preamble (alternating 0/1)
  - Start frame delimiter (SFD)
  - Source & destination MAC addresses
  - Type/length field
  - Data
  - CRC frame check sequence (FCS)

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

22

## Media Access Control (MAC)

- Distributed arbitration protocol
  - Obsolete if centralized switches are used
- Carrier Sense Multiple Access with Collision Detection (CSMA/CD)
  - Sense free carrier before sending
  - Sense while transmitting to detect collision
    - If collision, send jam signal
  - Binary exponential backoff
    - Increment collision counter  $c$
    - Wait random period in range  $0 \dots 2^c - 1$

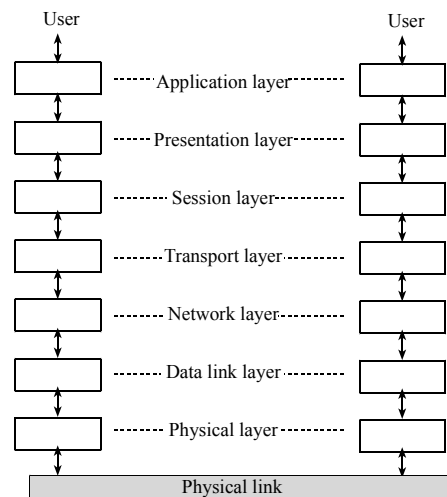
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

23

## ISO/OSI 7-Layer Model

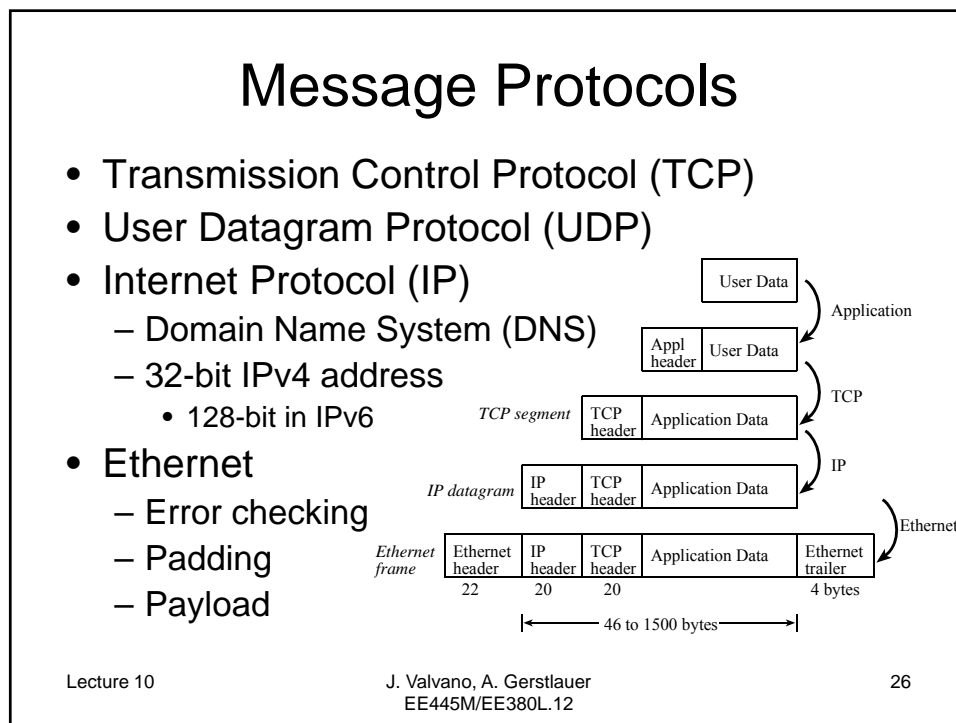
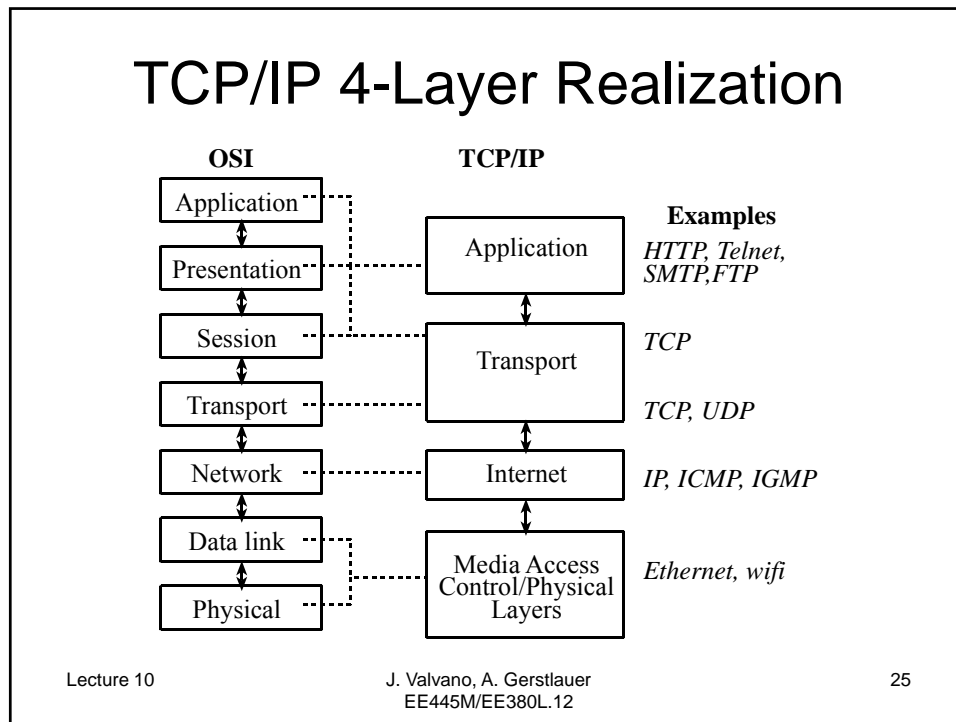
- Abstraction
  - International Standards Organization (ISO)
  - Open Systems Interconnection (OSI)



Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

24



## UDP and TCP

- UDP single datagram (connectionless)
  - 8-byte header

|                                                                               |
|-------------------------------------------------------------------------------|
| Source port: 16-bit number of the process that sent the packet, could be zero |
| Destination port: 16-bit number of the process to receive the packet.         |
| Length: 16-bit number specifying the size in bytes of the data to follow      |
| Checksum: 16-bit modulo addition of all data, UDP header, and IP header       |

- TCP session stream (error & flow control)
  - 20-byte header

|                                                                                |
|--------------------------------------------------------------------------------|
| Source port: 16-bit number of the process that sent the packet, could be zero  |
| Destination port: 16-bit number of the process to receive the packet.          |
| Sequence number: 32-bit number defining the position of this data              |
| Acknowledgement: 32-bit number of the next data expected to be received        |
| Hlen: 4-bit field of the header size (including options) divided by 4          |
| Flags: 6-bit field with FIN, SYN, RST, PSH, ACK, and URG                       |
| Window: 16-bit number specifying the number of bytes the receiver can accept   |
| Checksum: 16-bit modulo addition of all data, TCP header, and IP header        |
| Urgent pointer: 16-bit field pointing to a place in the stream urgently needed |

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

27

## Application Protocols

- Client/server models
  - HTTP (web), FTP, Telnet, SSH, ...
- Providing (device) services over the web
  - XML-based protocol: SOAP, WSDL
  - Simpler REST-ful API styles
    - Request encoded in URL
    - Return data using XML, JSON, ... formats

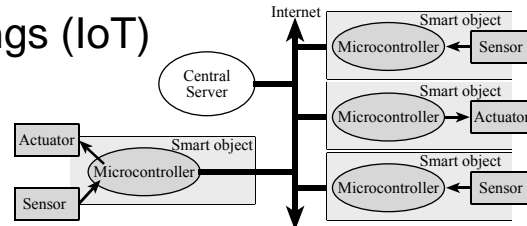
Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

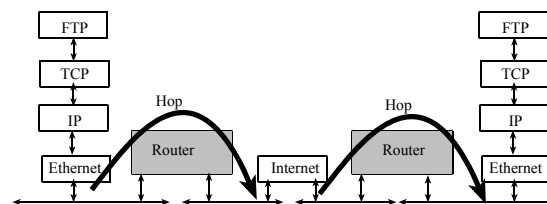
28

## Embedded Internet

- Internet of Things (IoT)



- Internet standardization: TCP/IP Protocol



Reference: W. Richard Stevens TCP/IP Illustrated, Volume 1: The Protocols

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

29

## Networking Summary

- Standardization is key
  - Communication standards
  - Across levels/layers
    - Physical, OS, application
- IoT is an open playing field
  - Internet as well-established foundation
  - Application-level standardization?
    - Embedded device-to-device interactions

Lecture 10

J. Valvano, A. Gerstlauer  
EE445M/EE380L.12

30