

Vorlesung Telematik (Computer Networks and the Internet)
WS2004/05

Data Link Layer

Prof. Dieter Hogrefe
Dr. Xiaoming Fu

Telematics group
University of Göttingen, Germany

Telematics group
University of Göttingen, Germany

Data Link Layer

- **Reliable transmission of packets over a link**
 - Framing: determine the start and end of packets
 - Error detection and correction (*see first part*)
 - Automatic Repeat ReQuest (ARQ)
 - PPP
- **Sharing a broadcast channel: multiple access**
 - Channel partitioning: TDMA, FDMA, CDMA
 - Random access: Aloha, CSMA, CSMA/CD
 - Taking turns: Token Ring, Token Bus
 - LAN examples: Ethernet and Wireless LAN
- **Interconnection devices (*self-learning*)**
 - Hubs, bridges, switches, and routers

Credits:

- > Eytan Modiano, MIT
- > James Kurose & Keith Ross: Computer Networking(2nd Ed.), Addison-Wesley, 2002

WS 2004/05, fu@cs.uni-goettingen.de

Telematics group
University of Göttingen, Germany

Framing

- Q: what is the lowest layer in the OSI model speaks sort of "*protocol*"?

010100111010100100101010100111000100

- Q: where is the DATA?
- A: Framing techniques (synchronization at data link layer) are needed: *protocol*
 - Method 1: Character oriented framing
 - Method 2: Length counts
 - » fixed length
 - Method 3: Bit oriented protocols (flags)

WS 2004/05, fu@cs.uni-goettingen.de

Telematics group
University of Göttingen, Germany

Character Based Framing

Frame

SYN	SYN	STX	Header	Packet	ETX	CRC	SYN	SYN
-----	-----	-----	--------	--------	-----	-----	-----	-----

- SYN is synchronous idle
- STX is start text
- ETX is end text
- **Standard character codes such as ASCII and EBCDIC contain special communication characters that cannot appear in data**
- **Entire transmission is based on a character code**

WS 2004/05, fu@cs.uni-goettingen.de

Issues with Character Based Framing

- Character code dependent
 - How do you send binary data?
- Frames must be integer number of characters
- Errors in control characters are messy

NOTE: Primary Framing method from 1960 to ~1975

Length field approach (DECNET)

- Use a header field to give the length of the frame (in bits or bytes)
 - Receiver can count until the end of the frame to find the start of the next frame
 - Receiver looks at the respective length field in the next packet header to find that packet's length
- Length field must be $\log_2(Max_Size_Packet) + 1$ bits long
 - This restricts the packet size to be used
- Issues with length counts
 - Difficult to recover from errors
 - Resynchronization is needed after an error in the length count

Fixed Length Packets (e.g., ATM)

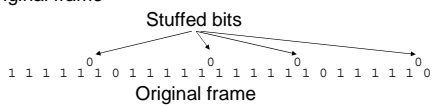
- All packets are of the same size
 - In ATM networks all packets are 53 Bytes
- Requires synchronization upon initialization
- Issues:
 - Message lengths are not multiples of packet size
 - Last packet of a message must contain idle fill (efficiency)
 - Synchronization issues
 - Fragmentation and re-assembly is complicated at high rates

Bit Oriented Framing (Flags)

- A flag is some fixed string of bits to indicate the start and end of a packet
 - A single flag can be used to indicate both the start and the end of a packet
- In principle, any string could be used, but appearance of flag must be prevented somehow in data
 - Standard protocols use the 8-bit string 01111110 as a flag
 - Use 01111111..1110 (<16 bits) as abort under error conditions
 - Constant flags or 1's is considered an idle state
- Thus 01111111 is the actual bit string that must not appear in data
- INVENTED ~ 1970 by IBM for SDLC (synchronous data link protocol)

BIT STUFFING (Transmitter)

- Used to remove flag from original data
- A 0 is stuffed after each consecutive five 1's in the original frame



- Why is it necessary to stuff a 0 in 0111110?
 - If not, then 0111110111 -> 0111110111 011111111 -> 0111110111
 - How do you differentiate at the receiver?

DESTUFFING (Receiver)

- If 0 is preceded by 011111 in bit stream, remove it
- If 0 is preceded by 0111111, it is the final bit of the flag.

Example: Bits to be removed are underlined below
1001111101100111011111011001111110
flag

Framing Errors

- All framing techniques are sensitive to errors
 - An error in a length count field causes the frame to be terminated at the wrongpoint (and makes it tricky to find the beginning of the next frame)
 - An error in DLE, STX, or ETX causes the same problems
 - An error in a flag, or a flag created by an error causes a frame to disappear or an extra frame to appear
- Flag approach is least sensitive to errors because a flag will eventually appear again to indicate the end of a next packet
 - Only thing that happens is that an erroneous packet was created
 - This erroneous packet can be removed through an error detection technique
- Error detection & correction: Parity check, Cyclic Redundancy Check (CRC)

Internet checksum

Goal: detect "errors" (e.g., flipped bits) in transmitted segment (note: used at transport layer *only*)

Sender:

- treat segment contents as sequence of 16-bit integers
- checksum: addition (1's complement sum) of segment contents
- sender puts checksum value into UDP checksum field

Receiver:

- compute checksum of received segment
- check if computed checksum equals checksum field value:
 - NO - error detected
 - YES - no error detected. *But maybe errors nonetheless?* More later

Telematics group
University of Göttingen, Germany

Checksumming: Cyclic Redundancy Check

- view data bits, **D**, as a binary number
- choose $r+1$ bit pattern (generator), **G**
- goal: choose r CRC bits, **R**, such that
 - $\langle D, R \rangle$ exactly divisible by G (modulo 2)
 - receiver knows G , divides $\langle D, R \rangle$ by G . If non-zero remainder: error detected!
 - can detect all burst errors less than $r+1$ bits
- widely used in practice (ATM, HDCL)

$\overbrace{\hspace{2cm}}^{d \text{ bits}} \quad \overbrace{\hspace{1cm}}^{r \text{ bits}}$
 $D: \text{data bits to be sent} \quad R: \text{CRC bits}$

bit pattern

$D \cdot 2^r \text{ XOR } R$ *mathematical formula*

WS 2004/05, fu@cs.uni-goettingen.de 13

Telematics group
University of Göttingen, Germany

CRC Example

Want:
 $D \cdot 2^r \text{ XOR } R = nG$
 equivalently:
 $D \cdot 2^r = nG \text{ XOR } R$
 equivalently:
 if we divide $D \cdot 2^r$ by G ,
 want remainder R

$R = \text{remainder} \left[\frac{D \cdot 2^r}{G} \right]$

```

      101011
    1001 ) 101110000
          1001
          ---
            101
             000
             ---
              1010
               1001
               ---
                110
                 000
                 ---
                  1100
                   1001
                   ---
                    1010
                     1001
                     ---
                      011
                       R
    
```

WS 2004/05, fu@cs.uni-goettingen.de 14

Telematics group
University of Göttingen, Germany

Automatic Repeat ReQuest

- When the receiver detects errors in a packet, how does it let the transmitter know to re-send the corresponding packet?
- Systems which automatically request the retransmission of missing packets or packets with errors are called **ARQ systems**.
- Three common schemes
 - Stop & Wait
 - Go Back N
 - Selective Repeat

WS 2004/05, fu@cs.uni-goettingen.de 15

Telematics group
University of Göttingen, Germany

Pure Stop and Wait Protocol

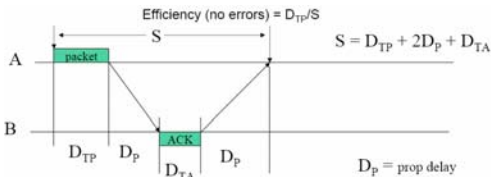
- Problem: Lost Packets
 - Sender will wait forever for an acknowledgement
- Packet may be lost due to framing errors

WS 2004/05, fu@cs.uni-goettingen.de 16

Efficiency of stop and wait

Let S = total time between the transmission of a packet and reception of its ACK

D_{TP} = transmission time of the packet



$$E = D_{TP} / (D_{TP} + 2D_p + D_{TA})$$

WS 2004/05, fu@cs.uni-goettingen.de

Stop and wait in the presence of errors

Let P = the probability of an error in the transmission of a packet or in its acknowledgment

$$S = D_{TP} + 2D_p + D_{TA}$$

TO = the timeout interval

X = the amount of time that it takes to transmit a packet and receive its ACK. This time accounts for retransmissions due to errors

$$E[X] = S + TO \cdot P / (1 - P), \text{ Efficiency} = D_{TP} / E[X]$$

Where,

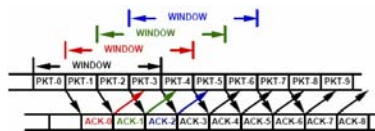
$TO = D_{TP}$ in a full duplex system

$TO = S$ in a half duplex system

WS 2004/05, fu@cs.uni-goettingen.de

Go Back N ARQ (Sliding Window)

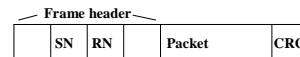
- Stop and Wait is inefficient when propagation delay is larger than the packet transmission time
 - Can only send one packet per round-trip time
- Go Back N allows the transmission of new packets before earlier ones are acknowledged
- Go back N uses a window mechanism where the sender can send packets that are within a "window" (range) of packets
 - The window advances as acknowledgements for earlier packets are received



WS 2004/05, fu@cs.uni-goettingen.de

Features of Go Back N

- Window size = N
 - Sender cannot send packet $i+N$ until it has received the ACK for packet i
- Receiver operates just like in Stop and Wait
 - Receive packets in order
 - Receiver cannot accept packet out of sequence
 - Send $RN = i + 1 \rightarrow$ ACK for all packets up to and including i
- Use of piggybacking
 - When traffic is bi-directional RN's are piggybacked on packets going in the other direction
 - Each packet contains a SN field indicating that packet's sequence number and a RN field acknowledging packets in the other direction



WS 2004/05, fu@cs.uni-goettingen.de

Go Back N ARQ

- The transmitter has a "window" of N packets that can be sent without acknowledgements
- This window ranges from the last value of RN obtained from the receiver (denoted SN_{min}) to $SN_{min}+N-1$
- When the transmitter reaches the end of its window, or times out, it goes back and retransmits packet SN_{min}

Let SN_{min} be the smallest number packet not yet ACKed

Let SN_{max} be the number of the next packet to be accepted from the higher layer (i.e., the next new packet to be transmitted)

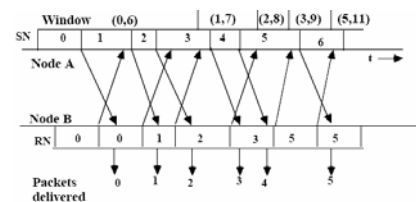
Go Back N: Sender Rules

- $SN_{min} = 0; SN_{max} = 0$
- Repeat
 - If $SN_{max} < SN_{min} + N$ (entire window not yet sent)
 - Send packet SN_{max} ;
 - $SN_{max} = SN_{max} + 1$;
 - If packet arrives from receiver with $RN > SN_{min}$
 - $SN_{min} = RN$;
 - If $SN_{min} < SN_{max}$ (there are still some unacknowledged packets) and sender cannot send any new packets
 - Choose some packet between SN_{min} and SN_{max} and re-send it
- The last rule says that when you cannot send any new packets you should re-send an old (not yet ACKed) packet
 - There may be two reasons for not being able to send a new packet
 - Nothing new from higher layer
 - Window expired ($SN_{max} = SN_{min} + N$)
 - No set rule on which packet to re-send
 - Least recently sent

Receiver Rules

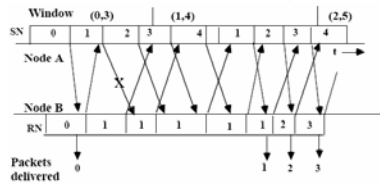
- $RN = 0$;
- Repeat
 - When a good packet arrives, if $SN = RN$
 - Accept packet
 - Increment $RN = RN + 1$
- At regular intervals send an ACK packet with RN
 - Most DLCs send an ACK whenever they receive a packet from the other direction
 - Delayed ACK for piggybacking
- Receiver reject all packets with SN not equal RN
 - However, those packets may still contain useful RN numbers
 - selective repeat: retransmit only those packets that are actually lost (due to errors)

Example of Go Back 7 ARQ



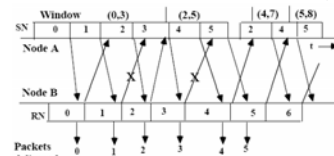
- Note that packet RN-1 must be accepted at B before a frame containing request RN can start transmission at B

RETRANSMISSION BECAUSE OF ERRORS FOR GO ACK 4 ARQ



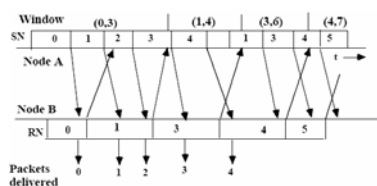
- Note that the timeout value here is taken to be the time to send a full window of packets
- Note that entire window has to be retransmitted after an error

RETRANSMISSION DUE TO FEEDBACK ERRORS FOR GO BACK 4 ARQ



- When an error occurs in the reverse direction the ACK may still arrive in time. This is the case here where the packet from B to A with RN=2 arrives in time to prevent retransmission of packet 0
- Packet 2 is retransmitted because RN = 4 did not arrive in time, however it did arrive in time to prevent retransmission of packet 3
 - Was retransmission of packet 4 and 5 really necessary?
 - Strictly no because the window allows transmission of packets 6 and 7 before further retransmissions. However, this is implementation dependent

EFFECT OF LONG FRAMES



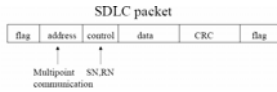
- Long frames in feedback direction slow down the ACKs
 - This causes a transmitter with short frames to wait or go back
- Notice again that the retransmission of packets 3 and 4 was not strictly required because the sender could have sent new packets within the window
 - Again, this is implementation dependent

Notes on Go Back N

- Requires no buffering of packets at the receiver
- Sender must buffer up to N packets while waiting for their ACK
- Sender must re-send entire window in the event of an error
- Packets can be numbered modulo M where $M > N$
 - Because at most N packets can be sent simultaneously
- Receiver can only accept packets in order
 - Receiver must deliver packets in order to higher layer
 - Cannot accept packet $i+1$ before packet i
 - This removes the need for buffering
 - This introduces the need to re-send the entire window upon error
- The major problem with Go Back N is this need to re-send the entire window when an error occurs. This is due to the fact that the receiver can only accept packets in order

Popular DLCs

- Older protocols (used for modems, e.g., xmodem) used stop and wait and simple checksums
- HDLC, LAPB (X.25), and SDLC are almost the same
 - HDLC/SDLC developed by IBM for IBM SNA networks
 - LAPB developed for X.25 networks
- They all use bit oriented framing with flag = 01111110
- They all use a 16-bit CRC for error detection
- They all use Go Back N ARQ with N = 7 or 127 (optional)



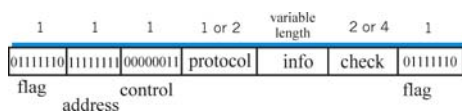
- Another DLC: Point-to-Point Protocol (PPP)

PPP Overview

- one sender, one receiver, one link: easier than broadcast link:
 - no **Media Access Control (MAC)**
 - no need for explicit MAC addressing
 - e.g., dialup link, ISDN line
- **error detection** (no correction)
- **connection liveness**: detect, signal link failure to network layer
- **Error recovery, flow control, data re-ordering all relegated to higher layers!**

PPP Data Frame

- **Flag**: delimiter (framing)
- **Address**: does nothing (only one option)
- **Control**: does nothing; in the future possible multiple control fields
- **Protocol**: upper layer protocol to which frame delivered (eg, PPP-LCP, IP, IPCP, etc)
- **info**: upper layer data being carried
- **check**: cyclic redundancy check for error detection



Byte Stuffing

- "data transparency" requirement: data field must be allowed to include flag pattern <01111110>
 - **Q**: is received <01111110> data or flag?
- **Sender**: adds ("stuffs") extra <01111110> byte after each <01111110> **data** byte
- **Receiver**:
 - two 01111110 bytes in a row: discard first byte, continue data reception
 - single 01111110: flag byte

Multiple Access protocols

- single shared broadcast channel
 - two or more simultaneous transmissions by nodes: interference
 - only one node can send **successfully** at a time
- multiple access protocol
- distributed algorithm that determines how nodes share channel, i.e., determine when node can transmit
 - communication about channel sharing must use channel itself!
 - what to look for in multiple access protocols:

MAC Protocols: Categories

Three broad classes (mostly used in Local Area Networks):

- **Channel Partitioning**
 - divide channel into smaller “pieces” (time slots, frequency, code)
 - allocate piece to node for exclusive use
 - Example: TDMA, FDMA, CDMA
- **Random Access**
 - channel not divided, allow collisions
 - “recover” from collisions
 - E.g., (slotted) ALOHA, CSMA(/CD), to be discussed
- **“Taking turns”**
 - tightly coordinate shared access to avoid collisions
 - E.g., token ring, token bus, to be discussed

Slotted ALOHA

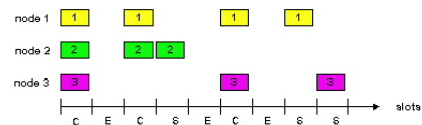
Assumptions

- all frames same size
- time is divided into equal size slots, time to transmit 1 frame
- nodes start to transmit frames only at beginning of slots
- nodes are synchronized
- if 2 or more nodes transmit in slot, all nodes detect collision

Operation

- when node obtains fresh frame, it transmits in next slot
- no collision, node can send new frame in next slot
- if collision, node retransmits frame in each subsequent slot with prob. p until success

Slotted ALOHA



Pros

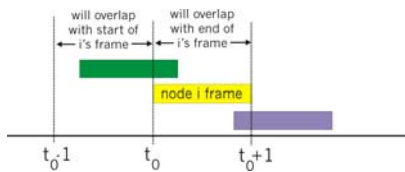
- single active node can continuously transmit at full rate of channel
- highly decentralized: only slots in nodes need to be in sync
- simple

Cons

- collisions, wasting slots
- idle slots
- nodes may be able to detect collision in less than time to transmit packet

Pure (unslotted) ALOHA

- unslotted Aloha: simpler, no synchronization
- when frame first arrives
 - transmit immediately
- collision probability increases:
 - frame sent at t_0 collides with other frames sent in $[t_0-1, t_0+1]$



CSMA (Carrier Sense Multiple Access)

CSMA: listen before transmit:

- If channel sensed idle: transmit entire frame
- If channel sensed busy, defer transmission
- Human analogy: don't interrupt others!
- **Collisions** can still occur in CSMA:
 - Propagation delay means two nodes may not hear each other's transmission
 - If collide, entire packet transmission time wasted

CSMA/CD (Collision Detection)

CSMA/CD: carrier sensing, deferral as in CSMA

- collisions *detected* within short time
- colliding transmissions aborted, reducing channel wastage
- collision detection:
 - easy in wired LANs: measure signal strengths, compare transmitted, received signals
 - difficult in wireless LANs: receiver shut off while transmitting
- human analogy: the polite conversationalist

"Taking Turns" MAC protocols

channel partitioning MAC protocols:

- share channel efficiently and fairly at high load
- inefficient at low load: delay in channel access, 1/N bandwidth allocated even if only 1 active node!

Random access MAC protocols

- efficient at low load: single node can fully utilize channel
- high load: collision overhead

"taking turns" protocols

look for best of both worlds!

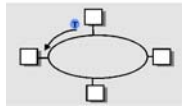
“Taking Turns” MAC protocols

Polling:

- master node “invites” slave nodes to transmit in turn
- concerns:
 - polling overhead
 - latency
 - single point of failure (master)

Token passing:

- control **token** passed from one node to next sequentially.
- token message
- concerns:
 - token overhead
 - latency
 - single point of failure (token)



Summary of MAC protocols

- What do you do with a shared media?
 - Channel Partitioning, by time, frequency or code
 - Time Division, Code Division, Frequency Division
 - Random partitioning (dynamic),
 - ALOHA, S-ALOHA, CSMA, CSMA/CD
 - carrier sensing: easy in some technologies (wire), hard in others (wireless)
 - CSMA/CD used in Ethernet
 - Taking Turns
 - polling from a central site, token passing

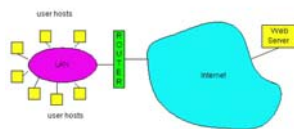
LAN technologies

Data link layer so far:

- services, error detection/correction, multiple access

Next: LAN technologies

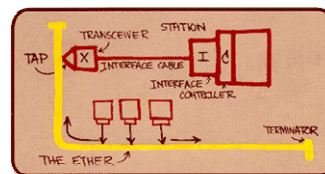
- Ethernet
- Self-study: WLAN, hubs, bridges, switches



Ethernet

“dominant” LAN technology:

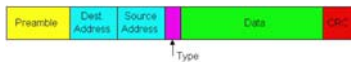
- cheap (<€20) for 100Mbps!
- first widely used LAN technology
- Simpler, cheaper than token LANs and ATM
- Kept up with speed race: 10, 100, 1000 Mbps



Metcalfe's Ethernet sketch

Ethernet Frame Structure

Sending adapter encapsulates IP datagram (or other network layer protocol packet) in **Ethernet frame**



Preamble:

- 7 bytes with pattern 10101010 followed by one byte with pattern 10101011
- used to synchronize receiver, sender clock rates

Ethernet Frame Structure (more)

- **Addresses:** 6 bytes
 - if adapter receives frame with matching destination address, or with broadcast address (eg. ARP- Address Resolution Protocol packet), it passes data in frame to network-layer protocol
 - otherwise, adapter discards frame
- **Type:** indicates the higher layer protocol, mostly IP but others may be supported such as Novell IPX and AppleTalk)
- **CRC:** checked at receiver, if error is detected, the frame is simply dropped



Unreliable, connectionless service

- **Connectionless:** No handshaking between sending and receiving adapter.
- **Unreliable:** receiving adapter doesn't send acks or nacks to sending adapter
 - stream of datagrams passed to network layer can have gaps
 - gaps will be filled if app is using TCP
 - otherwise, app will see the gaps

Ethernet uses CSMA/CD

- No slots
- adapter doesn't transmit if it senses that some other adapter is transmitting, that is, **carrier sense**
- transmitting adapter aborts when it senses that another adapter is transmitting, that is, **collision detection**
- Before attempting a retransmission, adapter waits a random time, that is, **random access**

Telematics group
University of Göttingen, Germany

Ethernet CSMA/CD algorithm

1. Adaptor gets datagram from and creates frame
2. If adaptor senses channel idle, it starts to transmit frame. If it senses channel busy, waits until channel idle and then transmits
3. If adaptor transmits entire frame without detecting another transmission, the adaptor is done with frame !
4. If adaptor detects another transmission while transmitting, aborts and sends jam signal
5. After aborting, adapter enters **exponential backoff**: after the mth collision, adapter chooses a K at random from $\{0, 1, 2, \dots, 2^m - 1\}$. Adapter waits $K \cdot 512$ bit times and returns to Step 2

WS 2004/05, fu@cs.uni-goettingen.de 53

Telematics group
University of Göttingen, Germany

Ethernet's CSMA/CD (more)

Jam Signal: make sure all other transmitters are aware of collision; 48 bits;

Bit time: .1 microsec for 10 Mbps Ethernet ; for K=1023, wait time is about 50 msec

Exponential Backoff:

- *Goal:* adapt retransmission attempts to estimated current load
 - heavy load: random wait will be longer
- first collision: choose K from $\{0, 1\}$; delay is $K \times 512$ bit transmission times
- after second collision: choose K from $\{0, 1, 2, 3\}$...
- after ten collisions, choose K from $\{0, 1, 2, 3, 4, \dots, 1023\}$

WS 2004/05, fu@cs.uni-goettingen.de 54

Telematics group
University of Göttingen, Germany

IEEE 802.11 Wireless LAN

- **802.11b**
 - 2.4-5 GHz unlicensed radio spectrum
 - up to 11 Mbps
 - direct sequence spread spectrum (DSSS) in physical layer
 - all hosts use same chipping code
 - widely deployed, using base stations
- **802.11a**
 - 5-6 GHz range
 - up to 54 Mbps
- **802.11g**
 - 2.4-5 GHz range
 - up to 54 Mbps
- All use CSMA/CA (collision avoidance) for multiple access
- All have base-station and ad-hoc network versions

WS 2004/05, fu@cs.uni-goettingen.de 55

Telematics group
University of Göttingen, Germany

Base station approach v.s. Ad-hoc approach

- **Base station approach:**
 - Wireless host communicates with a base station
 - base station = access point (AP)
 - **Basic Service Set (BSS)** (a.k.a. "cell") contains:
 - wireless hosts
 - access point (AP): base station
 - BSS's combined to form distribution system (DS)
- **Ad-hoc approach:**
 - no AP, each node communicates with each other
 - May use other node(s) for routing its packets

WS 2004/05, fu@cs.uni-goettingen.de 56

Homework

1. Understand CRC:
 - A) For the generator string $G=110011$ and data string $M=11100011$ find the CRC and the transmitted string T . (Since G is 6 bits long, $r=5$, and the CRC should be 5 bits long)
 - B) Suppose $G=1001$ and the received $T=1010101$, did any transmission errors occur?
 - C) Suppose $G=101$ and the received $T=1100110$, did any transmission errors occur?
 - D) Suppose $G=1011$ and $M=10010$ Give the shift register implementation of the CRC generator and show the register sequence for generating the CRC with the above value of M .
2. Review all reliability mechanisms provided by DLC. Understand synchronization, ARQ by examples
3. Understand other concepts: packet v.s. circuit switching, layered architectures, connection-oriented v.s. connectionless services, protocols, multiplexing
4. What is Ethernet? Explain three ways of multi access control and three types of LAN technology