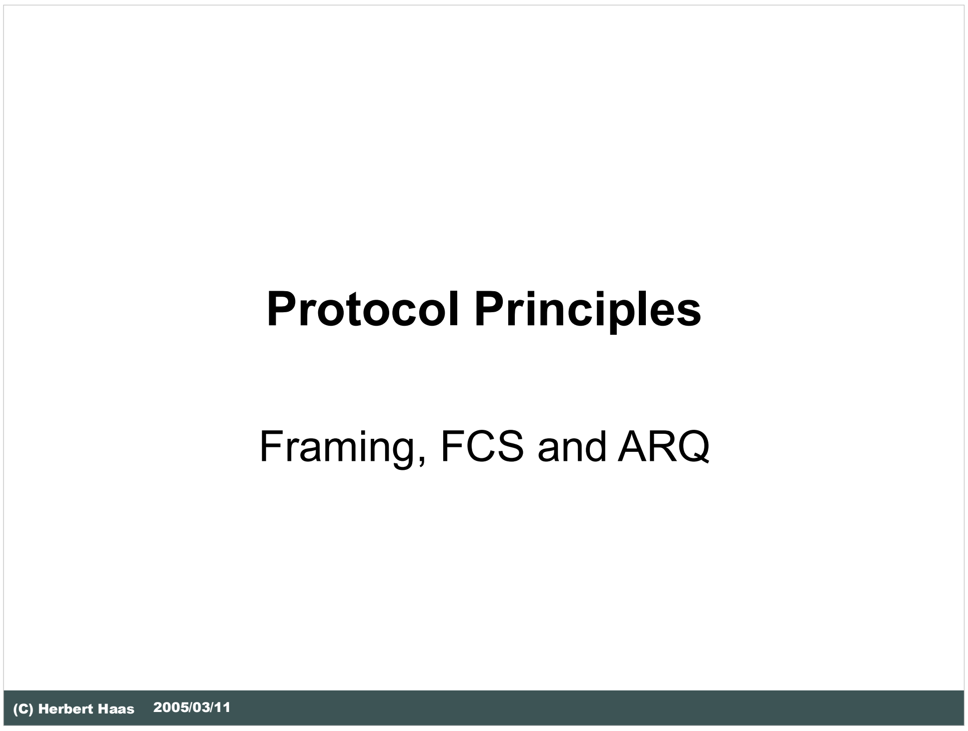
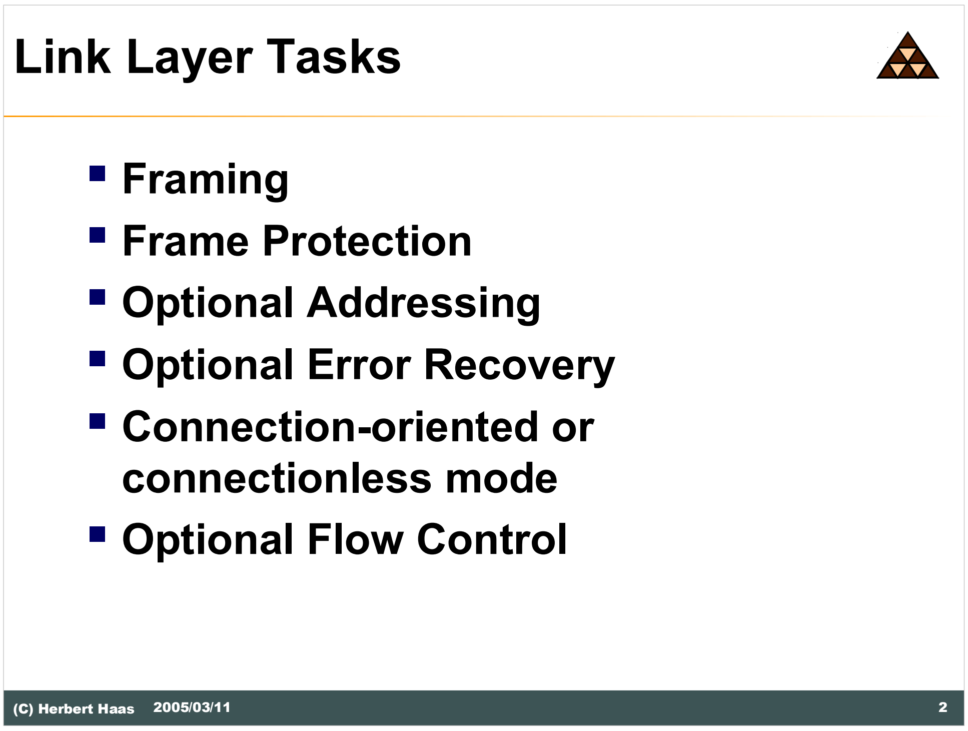


Protocol Principles

Framing, FCS and ARQ



Link Layer Tasks

- Framing
- Frame Protection
- Optional Addressing
- Optional Error Recovery
- Connection-oriented or connectionless mode
- Optional Flow Control

(C) Herbert Haas 2005/03/11 2

The Data-link layer of the OSI model is the first layer above the physical layer that is used to perform logical tasks like:

•**Framing** is the task of packing the information of higher layers to provide for example start and end of packet detection plus some optional features which will be discussed on the following slides

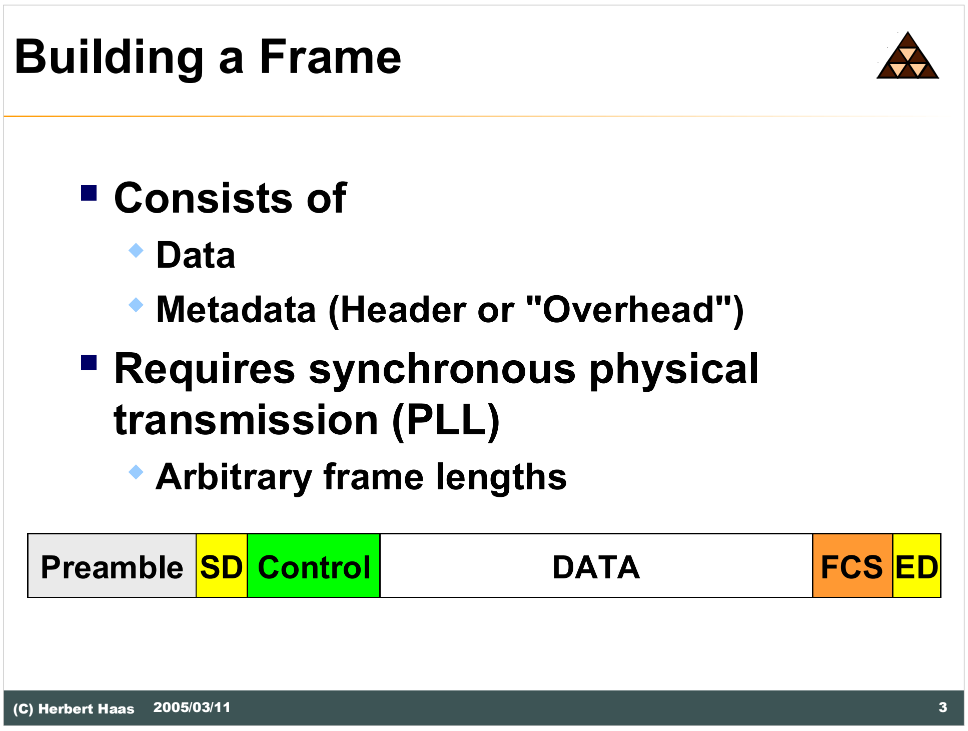
•**Frame protection** is used to detect possible errors during data transmission

•**Addressing** is optional and is normally only used in point-to-multipoint Data-link technologies.

•**Error recovery** can be used to allow packet retransmissions if data errors are detected by the frame protection mechanism

•The Data-link can be driven in **connection oriented** mode or **connection-less** mode. Newer technologies mainly use the connection-less mode because the connection-oriented functionality is typically provided by higher OSI layers e.g. TCP on OSI layer 4.

•**Flow control** can be used to prevent buffer overflow situations on the receiver side



Building a Frame

- Consists of
  - Data
  - Metadata (Header or "Overhead")
- Requires synchronous physical transmission (PLL)
  - Arbitrary frame lengths

Preamble	SD	Control	DATA	FCS	ED

(C) Herbert Haas 2005/03/11 3

A basic layer 2 transmission frame consists of following components:

•**Preamble -** is used to provide synchronization between the sender and the receiver transmission clock. This is necesarry to allow the detection of the single bit borders.

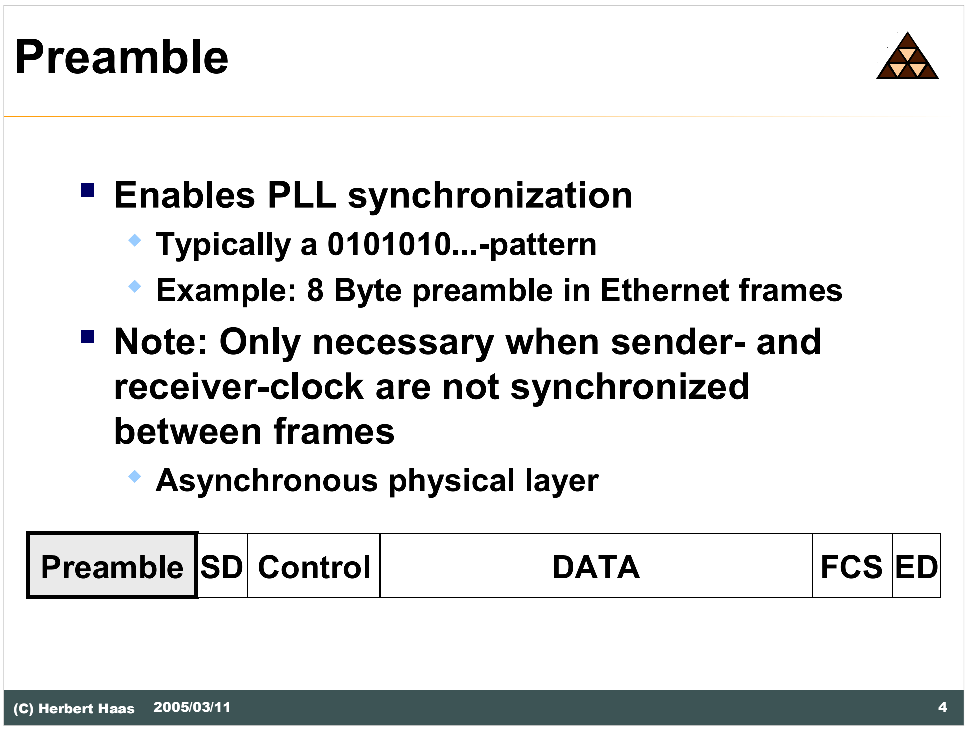
•**SD** - Start Delimiter is needed to detect the actual beginning of the layer 2 transmission frame. From this point on data is fed from the physical layer into the receive buffer.

•**Control Field** - provides optional addressing, connection establishment, error recovery and flow control

•**Data** - is the payload provided by higher OSI layers

•**FCS** - Frame Check Sequence is used for error detection

•**ED** - End Delimiter is used to determine the end of the layer 2 frame



Preamble

- Enables PLL synchronization
  - Typically a 0101010...-pattern
  - Example: 8 Byte preamble in Ethernet frames
- Note: Only necessary when sender- and receiver-clock are not synchronized between frames
  - Asynchronous physical layer

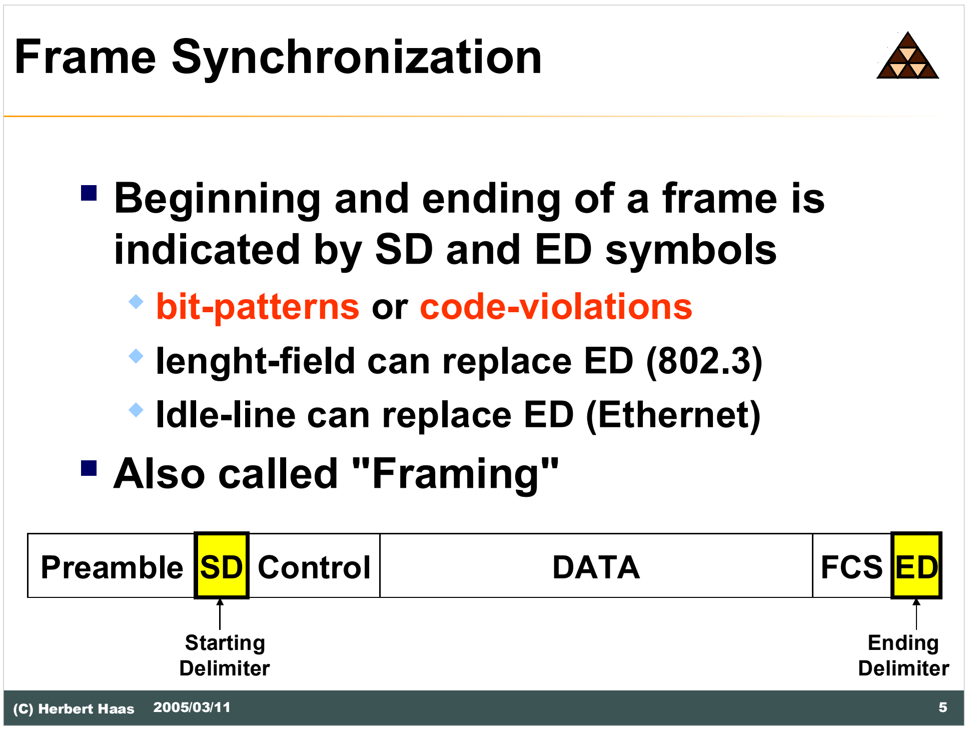
Preamble	SD	Control	DATA	FCS	ED

(C) Herbert Haas 2005/03/11 4

The purpose of the Preamble is to lock the receiver clock towards the sender clock by the help of Phase Locked Loop (PLL) circuits. The Preamble is different depending on the type of Data-link technology that is used.

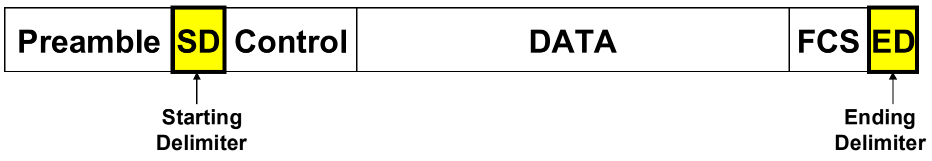
In Ethernet technology for example the bit pattern consists of 62 clock changes between a logical 1 and a logical 0, followed by two logical 1´s to indicate the start of frame.

The Preamble is obviously only needed for synchronous physical layers. In the case that an asynchronous physical layer is used, e.g. COM port on PC or async serial interface on a router, the Preamble is not needed. Because sender and receiver clocks don t need to be synchronized.



Frame Synchronization

- Beginning and ending of a frame is indicated by SD and ED symbols
  - bit-patterns or code-violations
  - lenght-field can replace ED (802.3)
  - Idle-line can replace ED (Ethernet)
- Also called "Framing"

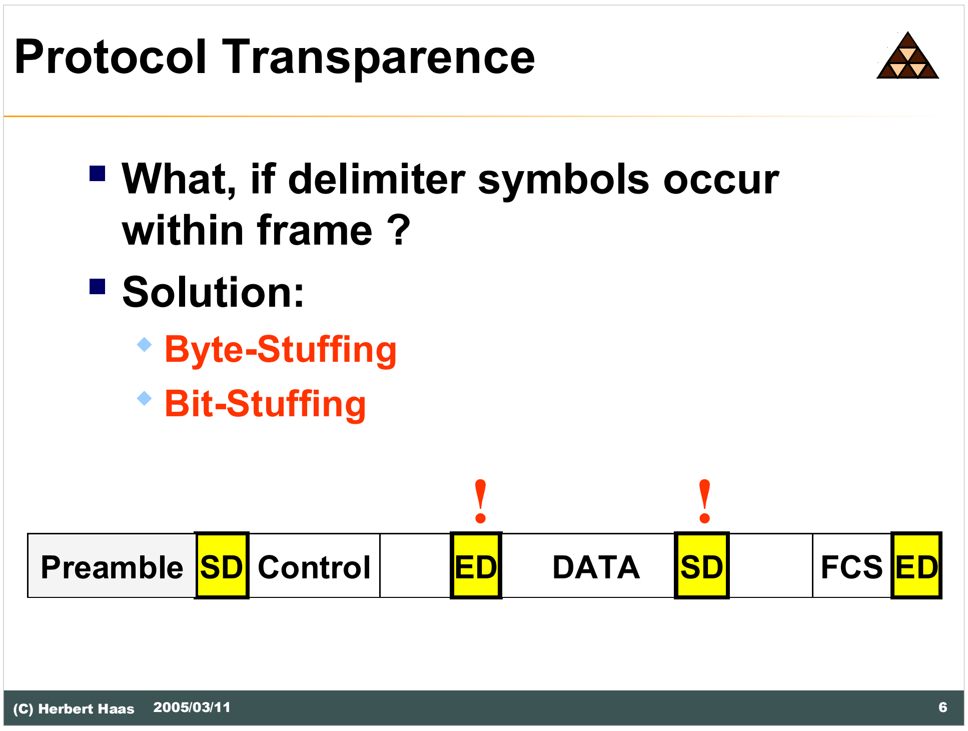


There are different methods available to indicate the start and the end of a data frames. The simplest method is the use of a special bit pattern. In the HDLC protocol and its derivates the bit pattern “01111110” is used to indicate start and end of frame.

In Token Ring technology code violation is used. Code violation is an intended brake in the rules of coding.

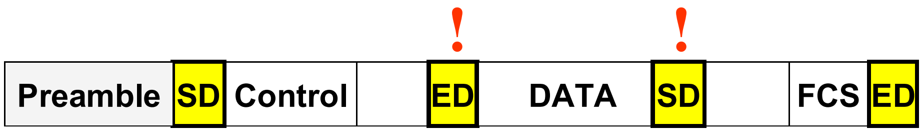
In Ethernet technology the SD is indicated by the bit pattern “11” following the Preamble. But the end of the frame is indicated by an idle line, this means silence on the wire for a specified period of time.

Optionally the ED can be omitted if an length field is present inside the Data-link frame. In this case the end of frame can be calculated by counting the number of bytes received.



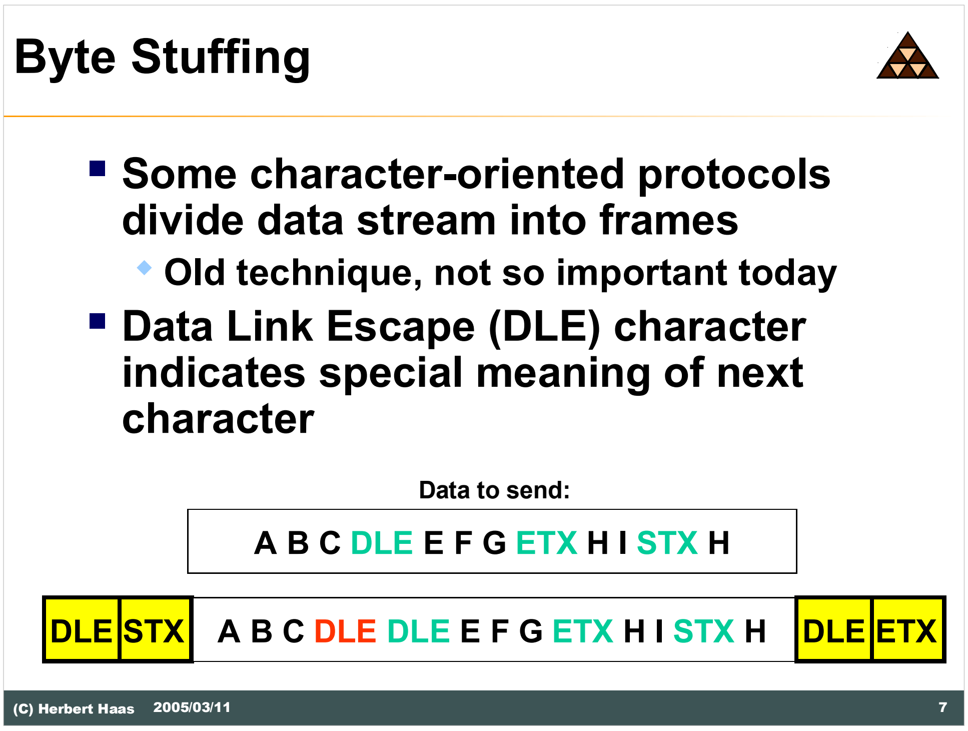
Protocol Transparence

- What, if delimiter symbols occur within frame ?
- Solution:
  - Byte-Stuffing
  - Bit-Stuffing



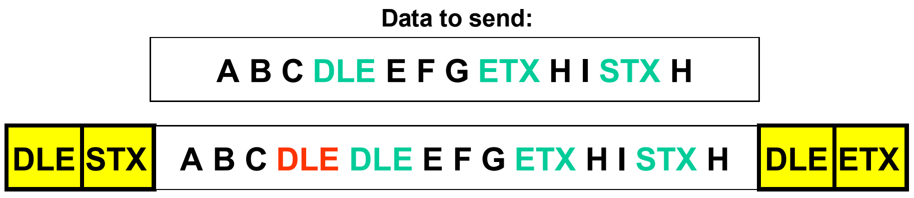
In the case that a special bit pattern is used to indicate the start and end of a frame, it is necessary to prevent this pattern inside the data portion of the frame. Otherwise this would lead to frame misinterpretation.

There are two principle methods to achieve this goal either by modifying single bits of the data stream (bit-stuffing) or by replacing the whole byte (byte-stuffing).



Byte Stuffing

- Some character-oriented protocols divide data stream into frames
  - Old technique, not so important today
- Data Link Escape (DLE) character indicates special meaning of next character

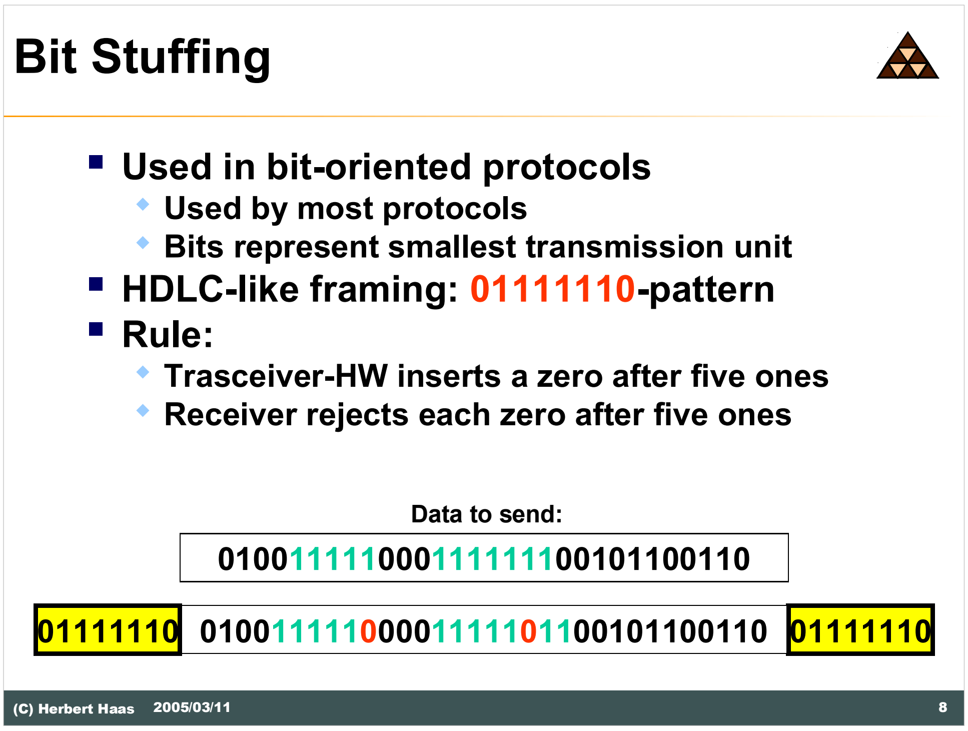

Data to send:
A B C DLE E F G ETX H I STX H
DLE STX A B C DLE DLE E F G ETX H I STX H DLE ETX

Typically, character-oriented protocols use asynchronous transmission (start-stop bits), so the receiver gets a bunch of characters but has to find the frames in it. In this case special control characters Start of Text (STX, 0x02) and End of Text (ETX, 0x03) are used to indicate start/end of a transmission frame.

Obviously STX and ETX must not appear inside the date portion, so an additional special control character Data Link Escape (DLE, 0x10) is used.

STX and ETX are only interpreted as control characters if the DLE pattern is found in front of them. This means if STX or ETX bit pattern are found inside the data stream, with the DLE pattern in front of them, they are interpreted as data.

If the DLE pattern itself would appear inside the data portion it is doubled to indicate that it is only data.



Bit Stuffing

- Used in bit-oriented protocols
  - Used by most protocols
  - Bits represent smallest transmission unit
- HDLC-like framing: 01111110-pattern
- Rule:
  - Trasceiver-HW inserts a zero after five ones
  - Receiver rejects each zero after five ones

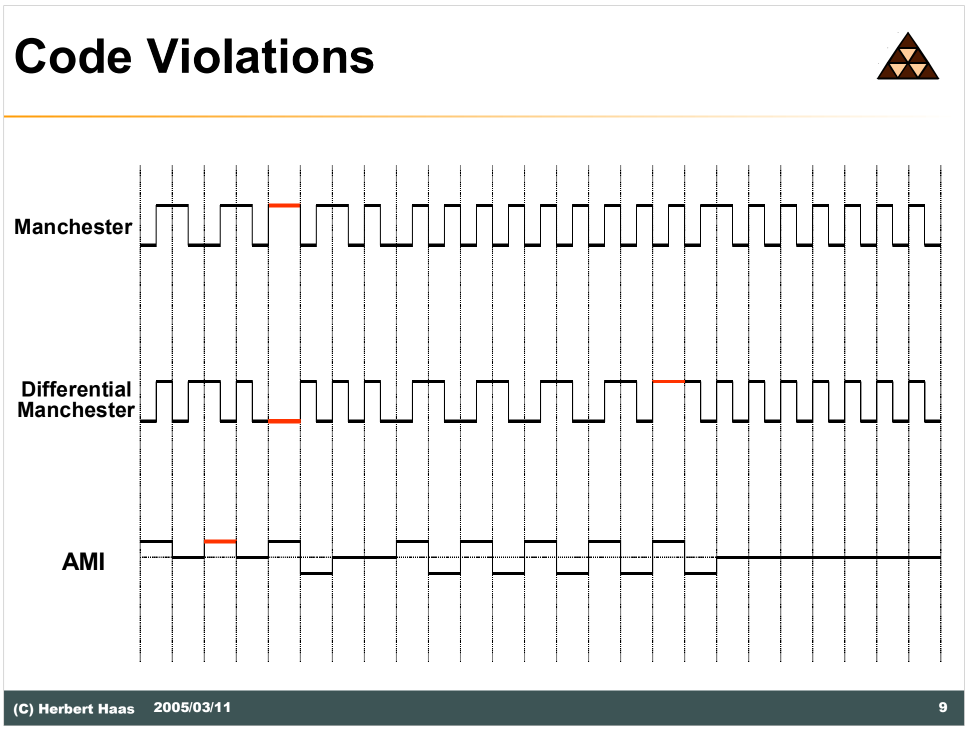
Data to send:

010011111000111111100101100110

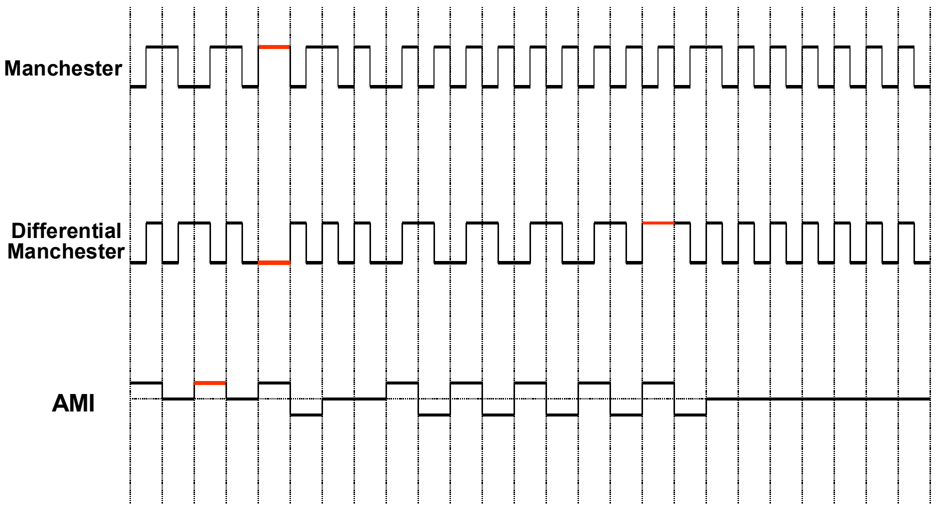
01111110	01001111100001111101100101100110	01111110

In HDLC technology and its derivates bit stuffing is used to prevent the appearing of the SD, ED pattern inside the data frame.

This means there is a common rule between sender and receiver that after every fifth´s 1 an additional logical 0 will be inserted by the sender in the data stream. The receiver itself removes all logical 0´s following five logical 1´s.



Code Violations

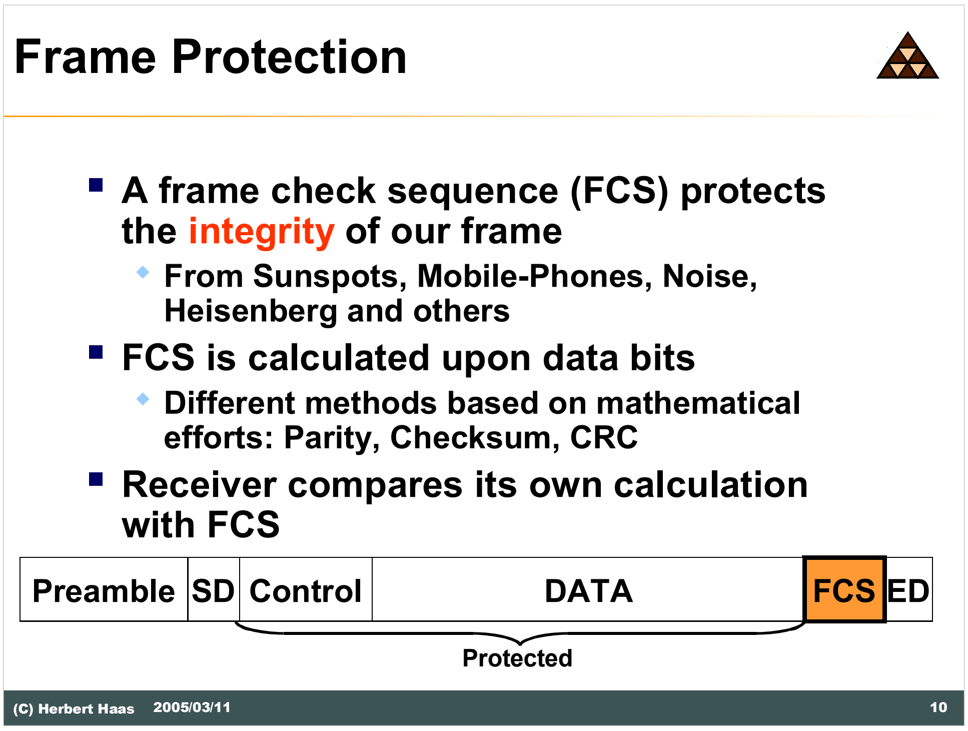


(C) Herbert Haas 2005/03/11 9

Code violation is an intended hurt of the rules of coding. It can be used for signaling purposes.

In Token Ring systems for example the differential manchester code is used. Violations of the differential manchester code are used for the SD and ED patterns to indicate the start and the end of frame.

The differential manchester code violation symbols are called J and K. The code violation in the differential manchester code is achieved by omitting the change from 1 to 0 or from 0 to 1 in the middle of a pulse.



Frame Protection

- A frame check sequence (FCS) protects the integrity of our frame
  - From Sunspots, Mobile-Phones, Noise, Heisenberg and others
- FCS is calculated upon data bits
  - Different methods based on mathematical efforts: Parity, Checksum, CRC
- Receiver compares its own calculation with FCS

Preamble
SD
Control
DATA
FCS
ED
Protected

(C) Herbert Haas 2005/03/11 10

The Frame Check Sequence (FCS) allows the receiver to detect possible errors in the data stream.

The FCS is calculated by the sender and is attached at the end of the frame. The calculation of the FCS is typically performed in hardware.

There are many different technologies available to calculate an FCS like:

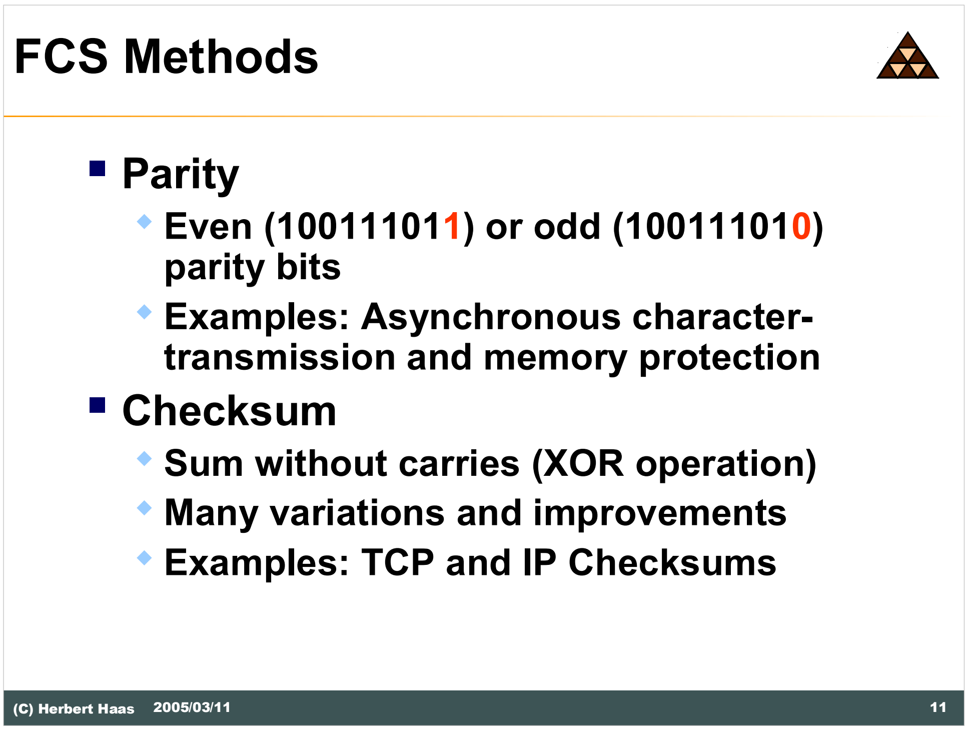
•Parity bit calculation

•XOR operation

•Modulo operations

•Cycle Redundancy Check (CRC)

•Forward Error correction (FEC)



FCS Methods

- Parity
  - Even (100111011) or odd (100111010) parity bits
  - Examples: Asynchronous character-transmission and memory protection
- Checksum
  - Sum without carries (XOR operation)
  - Many variations and improvements
  - Examples: TCP and IP Checksums

(C) Herbert Haas 2005/03/11 11

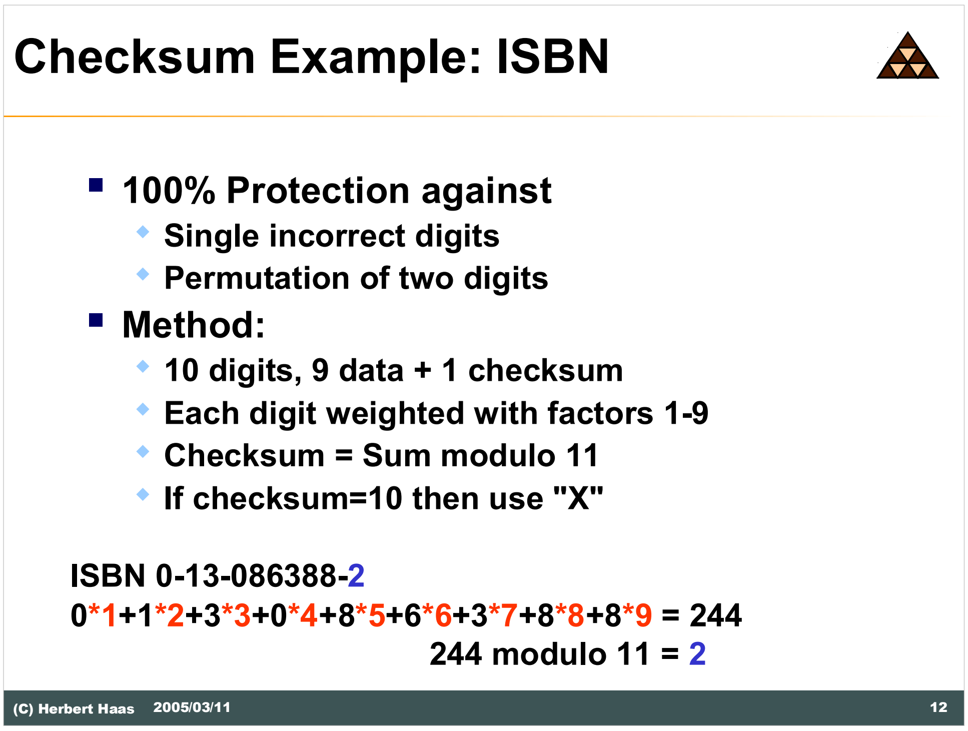
The simplest method of FCS technologies is the parity bit calculation. This method is typically used in asynchronous character based transmission systems.

One parity bit is computed for each single character. The first two least significant bits are XORed together and the output of this operation is then XORed with the next more significant bit, and so on. The output of the final operation represents the required parity bit, which can be even (1) or odd (0).

Obviously the parity check can only protect against single bit errors. A two bit failure for example in the opposite direction 1 to 0 and 0 to 1 cannot be detected by the parity check.

For packet based transmission systems a similar method to calculate a checksum can be used. Instead of a single bit, 16 or 32 bit long words are used in combination with the XOR operation.

The XOR operation is also known as a modulo-2 adder, since the output of the XOR operation between two binary digits is the same as the addition of the two digits without a carry bit.



Checksum Example: ISBN

- 100% Protection against
  - Single incorrect digits
  - Permutation of two digits
- Method:
  - 10 digits, 9 data + 1 checksum
  - Each digit weighted with factors 1-9
  - Checksum = Sum modulo 11
  - If checksum=10 then use "X"

ISBN 0-13-086388-2
0*1+1*2+3*3+0*4+8*5+6*6+3*7+8*8+8*9 = 244
244 modulo 11 = 2

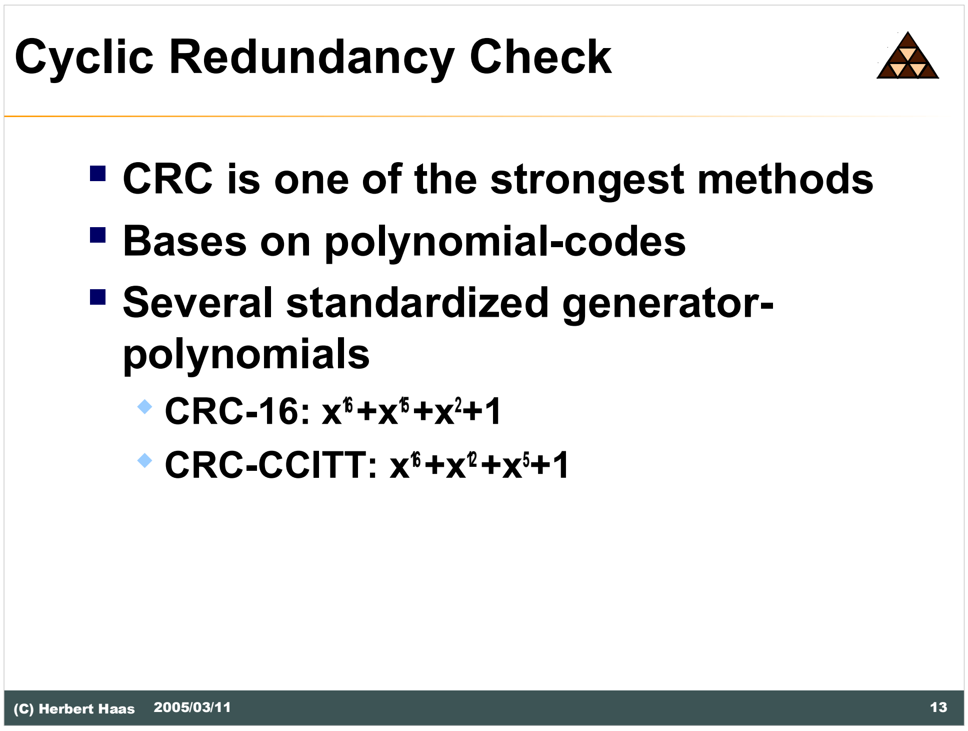
(C) Herbert Haas 2005/03/11 12

ISBN stands for International Standard for Book Numbering.

The ISBN code which you may use to order your favourite books is protected by an checksum as well. This checksum is built on the basis of an modulo 11 operation.

The ISBN code itself consists of 10 digits where the first nine digits represent the book code and the ten´s digit is the checksum used for error detection.

The ISBN checksum allows the detection of a single bit failure or the permutation of two bits with a probability of 100%.



Cyclic Redundancy Check

- **CRC is one of the strongest methods**
- **Bases on polynomial-codes**
- **Several standardized generator-polynomials**
  - **CRC-16: $x^{16}+x^{15}+x^{2}+1$**
  - **CRC-CCITT: $x^{16}+x^{12}+x^{5}+1$**

(C) Herbert Haas 2005/03/11 13

Checksums which are built on basis of the XOR operation do not provide a reliable detection scheme against error bursts.

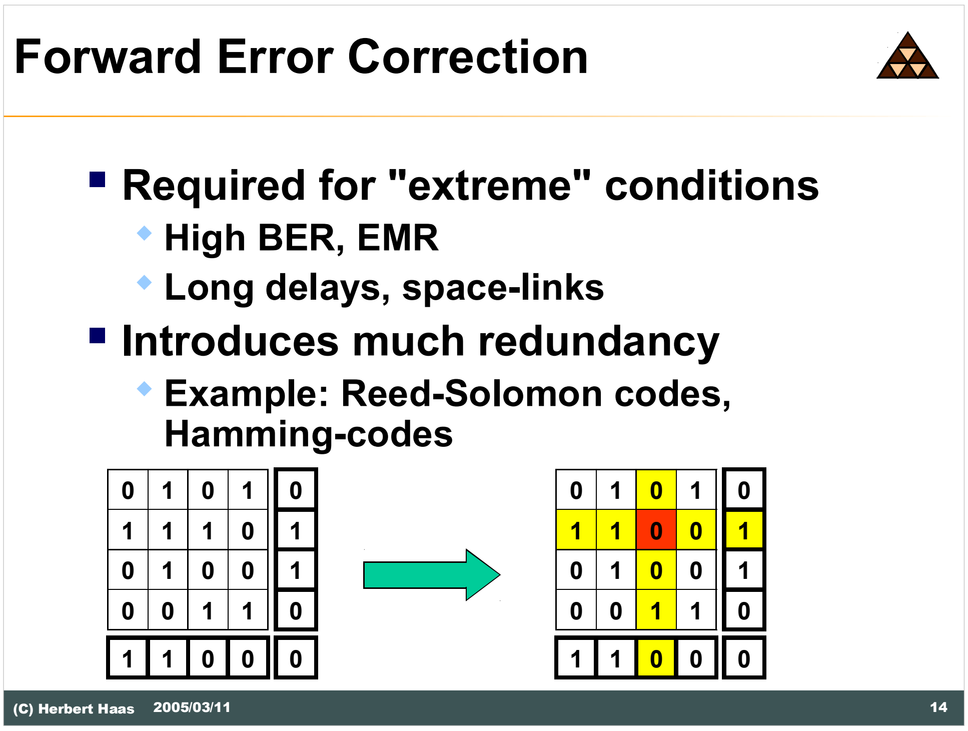
Therefore the Cycle Redundancy Check is based on mathematical polynomial equations and it is capable to detect even bursts of erroneous bits inside the data stream.

A single set of check bits is generated using an polynomial equation for each transmitted frame and appended to the tail of the frame by the sender. The receiver then performs the same computation on the complete frame and compares the received checksum with its own calculated checksum.

There are a lot of different standardized polynomial equations like the CRC-16, CRC-32 or the CRC-CCITT equations.

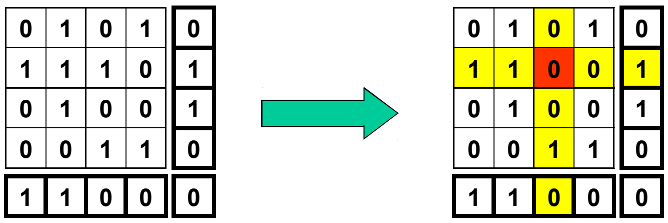
The CRC-16 check for example will detect all error bursts of less than 16 bits and most error bursts greater than 16 bits. The CRC-16 and the CRC-CCITT are mainly used in WAN technologies, while the CRC-16 is mainly used in LAN environments.

It is quite simple to implement these CRC checks in Hardware with the use of 16 or 32 bit long shift registers.



Forward Error Correction

- Required for "extreme" conditions
  - High BER, EMR
  - Long delays, space-links
- Introduces much redundancy
  - Example: Reed-Solomon codes, Hamming-codes



(C) Herbert Haas 2005/03/11 14

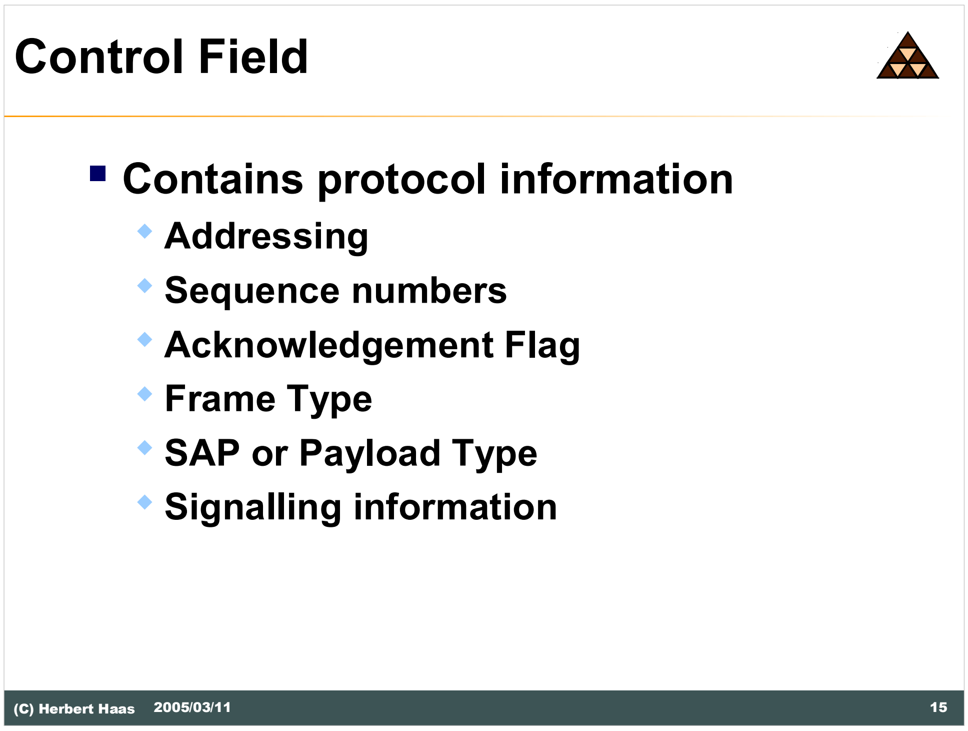
All till now discussed error protection technologies lead to an drop or optional retransmission of the corrupted data fame. With the help of Forward Error Correction (FEC) technologies it is possible to fix the faults inside a data frame up to a certain extend.

One error correction coding scheme is the Hamming FEC scheme. In this scheme the data bits plus the additional check bits are called the codeword. The minimum number of bit positions in which two valid codewords differ is known as the Hamming distance of the code.

It can be shown that to correct *n* errors we need a code with a Humming distance of $2n + 1$.

In practice the number of check bits needed for error correction is much larger than the number of bits needed just for error detection. Therefore in most transport systems ARQ techniques are still used for error correction.

FEC technologies are mainly used in nische technologies, e.g. communication with space probes, where the RTT is very high and sometimes only a unidirectional communication channels exists.



Control Field

- Contains protocol information
  - Addressing
  - Sequence numbers
  - Acknowledgement Flag
  - Frame Type
  - SAP or Payload Type
  - Signalling information

(C) Herbert Haas 2005/03/11 15

The contents of the control field depends on the tasks that need to be performed by the Data-link protocol.

So the control field could contain:

•Address information for addressing especially in point to multipoint environments

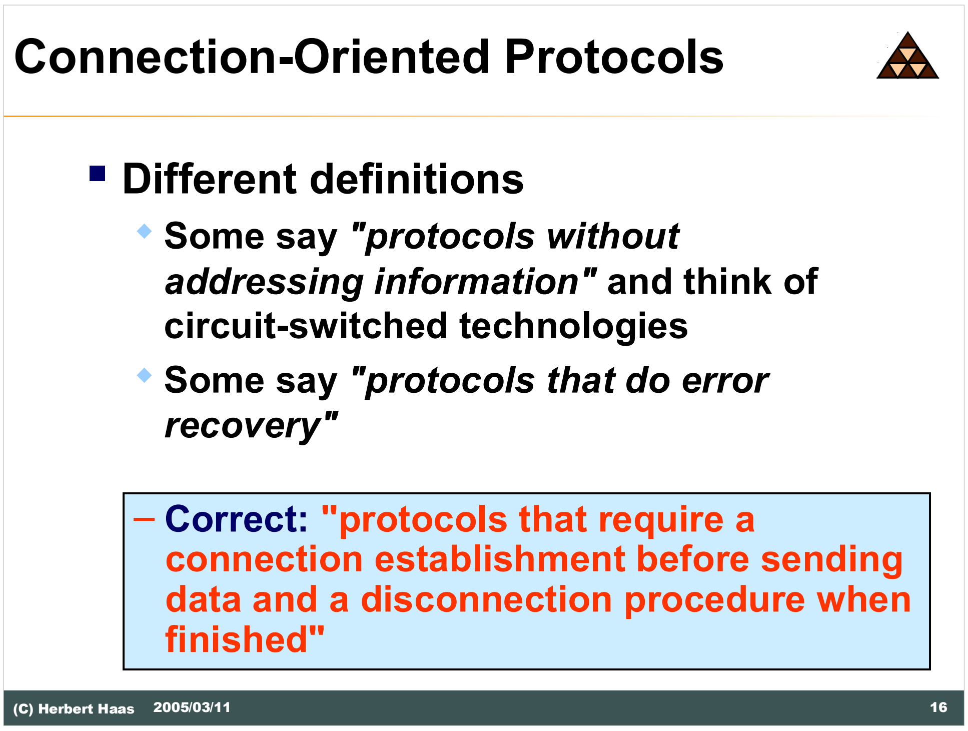
•Sequence numbers that can be seen like serial numbers for each single frame

•Acknowledgement Flags to indicate that the data was received properly

•Frame Type information to indicate whether it's a frame that carries data or control information

•Service Access Point (SAP) or payload type information to indicate what is transported by the frame

•Signaling information in case of connection oriented protocols to build up an connection


Connection-Oriented Protocols

- Different definitions
  - Some say *"protocols without addressing information"* and think of circuit-switched technologies
  - Some say *"protocols that do error recovery"*

– **Correct: "protocols that require a connection establishment before sending data and a disconnection procedure when finished"**

(C) Herbert Haas 2005/03/11 16

In the past an connection oriented protocol was very often seen as a protocol that performs a connection setup procedure and supports error recovery and flow control.

But newer technologies like frame relay typically do not carry address information, do not need an connection setup procedure in case of PVC service and do not support error recovery and flow control. But they are still connection oriented.

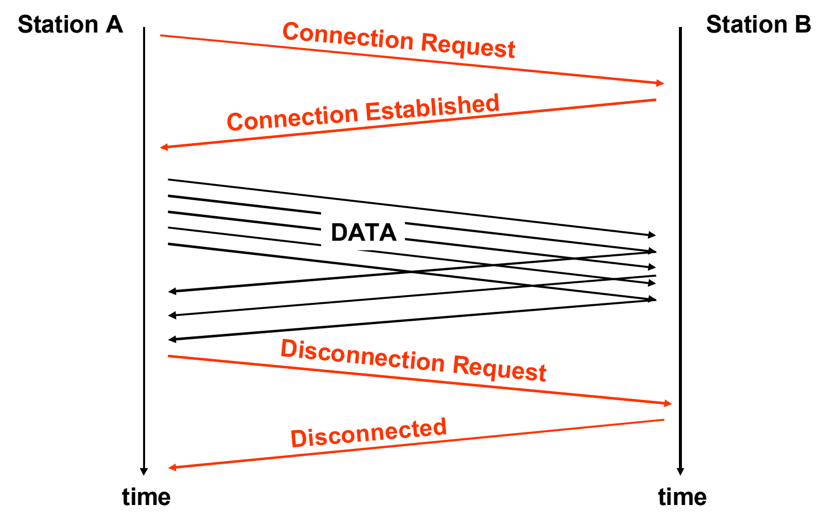
So we may say there are two different types of connections temporary (like SVC) and permanent (like PVC).

For temporary connections we obviously will need addressing and connection setup procedures.

For permanent connections a virtual circuit identifier is enough.

In both cases temporary or permanent connections we may optionally support error recovery and flow control.

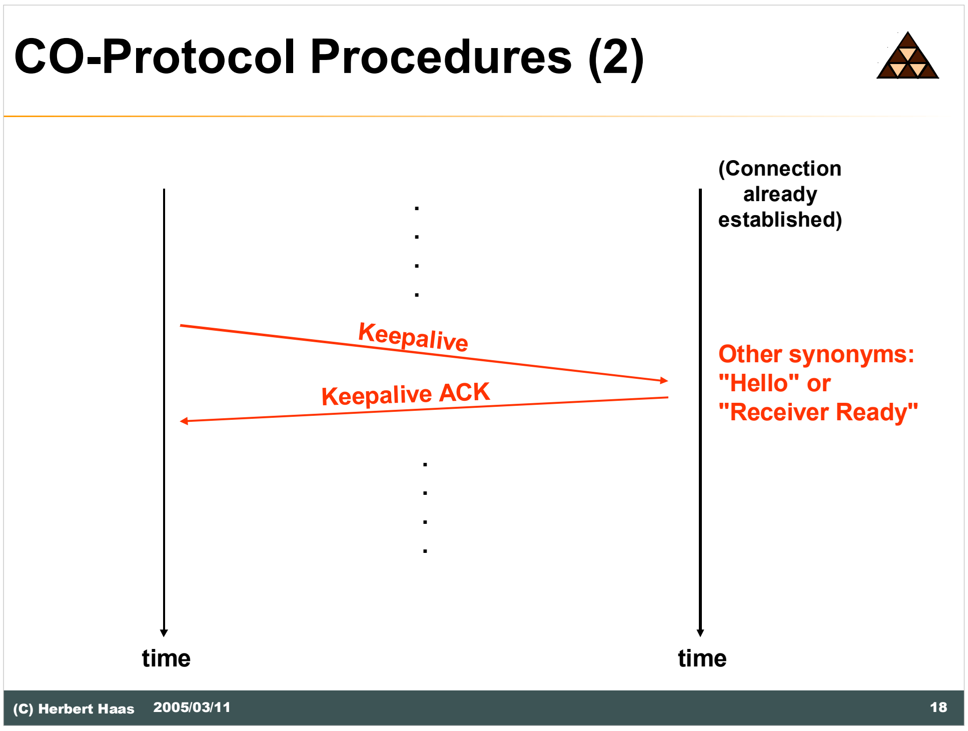
CO-Protocol Procedures (1)



In connection oriented protocols a connection is established before data is allowed to flow.

The connection establishment is done by special control frames like connection request and connection established.

Then we find the data exchange phase which may use error recovery and flow control. When the data transmission is finished we have special control frames like disconnect request and disconnected to tear down the connection again.

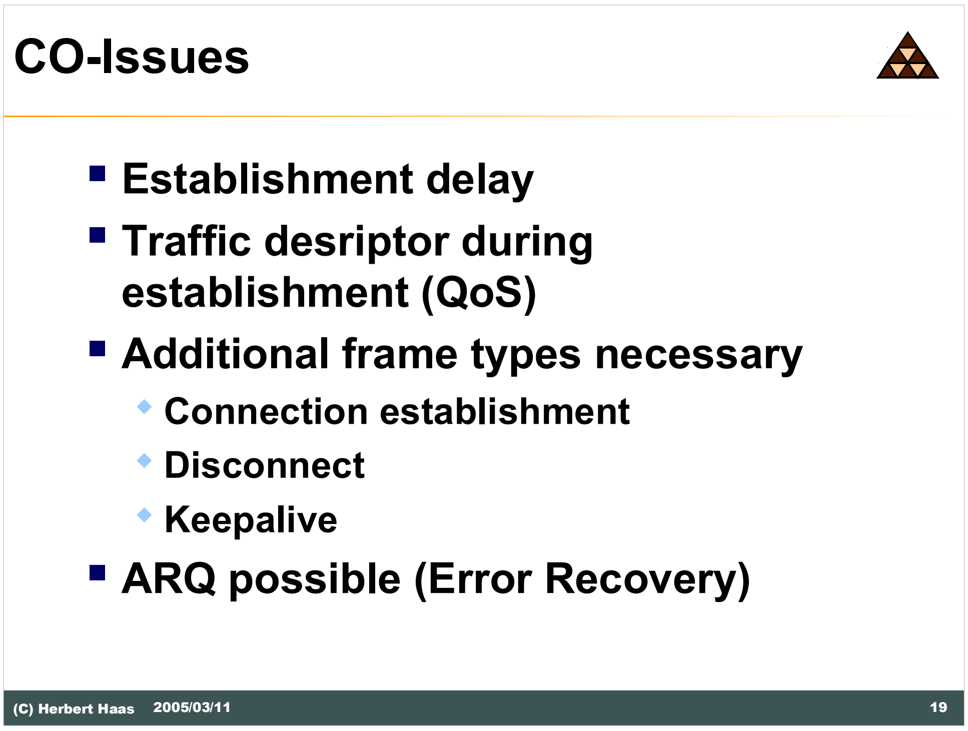


CO-Protocol Procedures (2)

(Connection already established)
Keepalive
Keepalive ACK
Other synonyms: "Hello" or "Receiver Ready"
time
time

(C) Herbert Haas 2005/03/11 18

Connection oriented protocols use special keep-alive procedures to make sure the connection is up in periods where no data frames are transmitted.



CO-Issues

- Establishment delay
- Traffic desriptor during establishment (QoS)
- Additional frame types necessary
  - Connection establishment
  - Disconnect
  - Keepalive
- ARQ possible (Error Recovery)

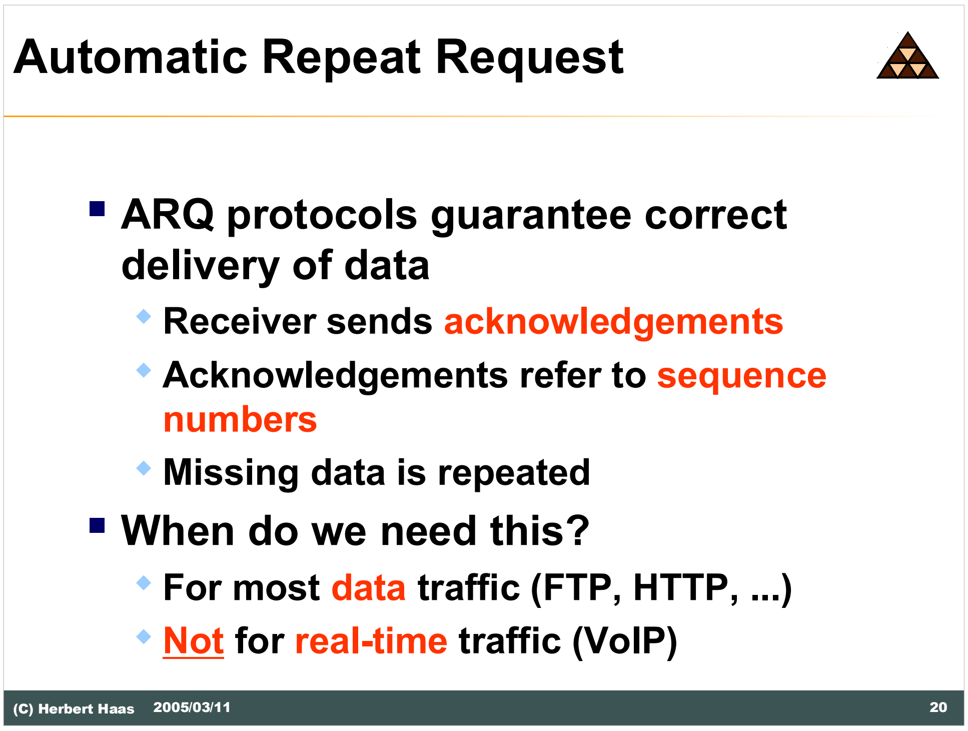
(C) Herbert Haas 2005/03/11 19

Obviously it takes some time to establish a connection before the data is allowed to flow. But the establishment delay is typically in the range of milliseconds. Even ISDN provides a connection establishment worldwide in less than one second.

A traffic descriptor may be used optionally for all technologies that support Quality of Service features. The traffic descriptor holds the information about the service parameters, e.g. delay, burst size etc, that should be used for this connection.

There are also separate frame types defined for connection establishment, disconnect procedures and keep-alives.

The use of Automatic Repeat Requests (ARQ) is optional depending on the technology used. The purpose of the ARQ is to provide error recovery in case of transmission failures.



Automatic Repeat Request

- ARQ protocols guarantee correct delivery of data
  - Receiver sends acknowledgements
  - Acknowledgements refer to sequence numbers
  - Missing data is repeated
- When do we need this?
  - For most data traffic (FTP, HTTP, ...)
  - Not for real-time traffic (VoIP)

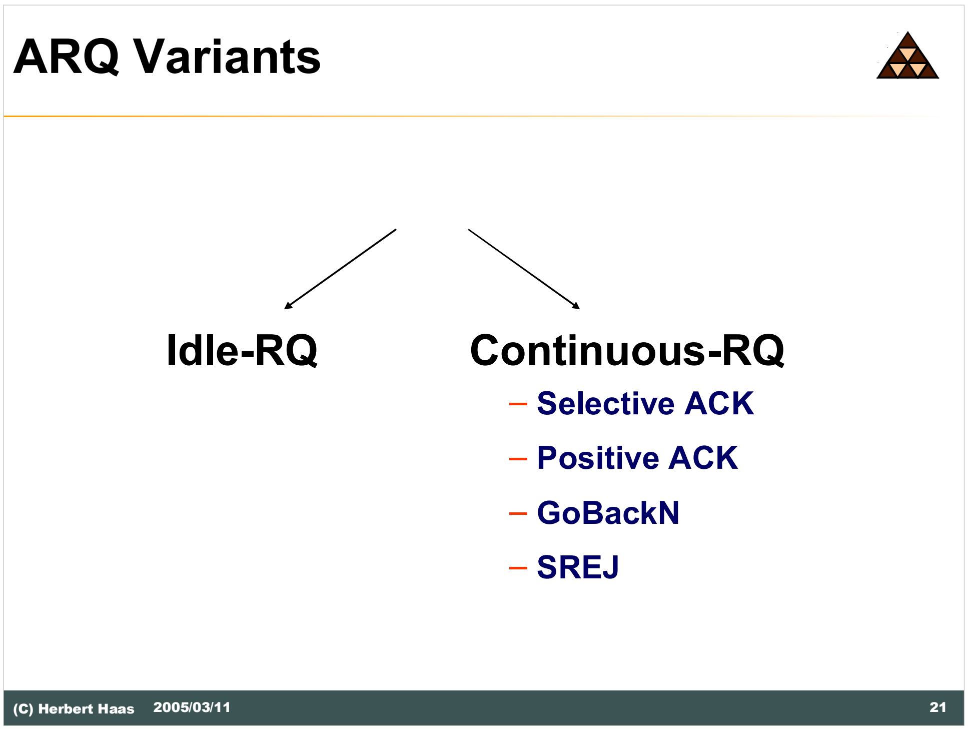
(C) Herbert Haas 2005/03/11 20

ARQ techniques are used to allow data retransmissions in the case of transmission errors and packet loss.

With the help of sequence numbers (serial number of a data packet), applied by the sender, the receiver is able to detect packet loss and is further able to acknowledge properly received frames.

Reliable transmission techniques are mainly used for data traffic e.g. SMTP, HTTP, FTP etc, because we don t want to receive corrupted mails or html pages.

For real time traffic like Voice over IP systems we prefer unreliable transmission techniques, because it makes no sense to retransmit a lost word a few milliseconds later again. This would destroy the harmony in the speech even more than the lost word. For real time systems Forward Error Correction systems would be much more useful.



ARQ Variants

Idle-RQ

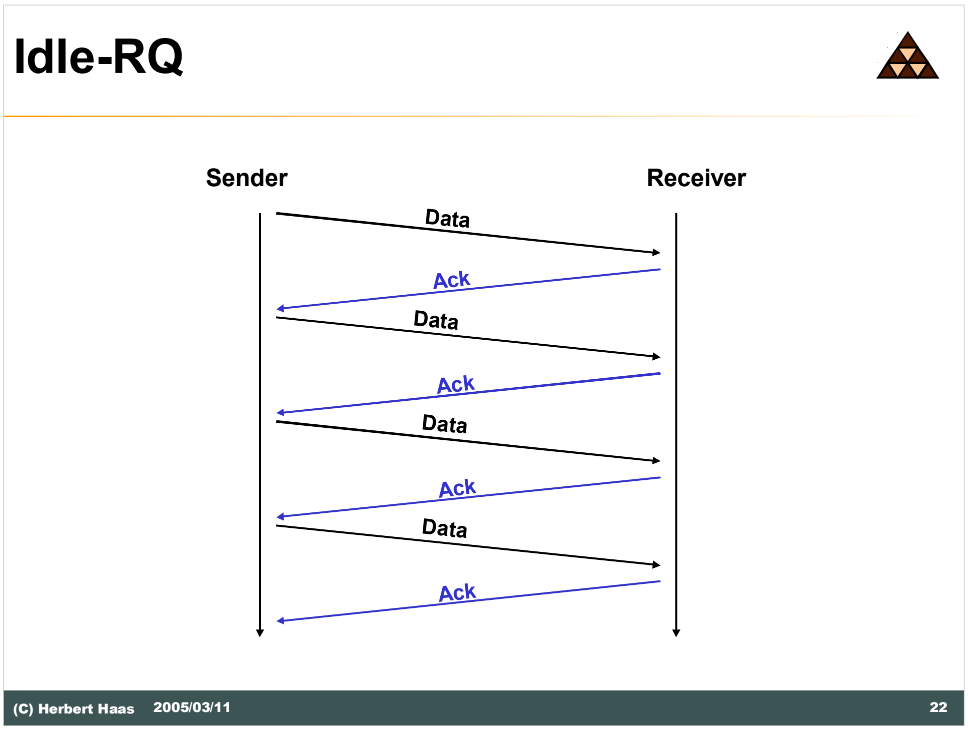
Continuous-RQ

– Selective ACK
– Positive ACK
– GoBackN
– SREJ

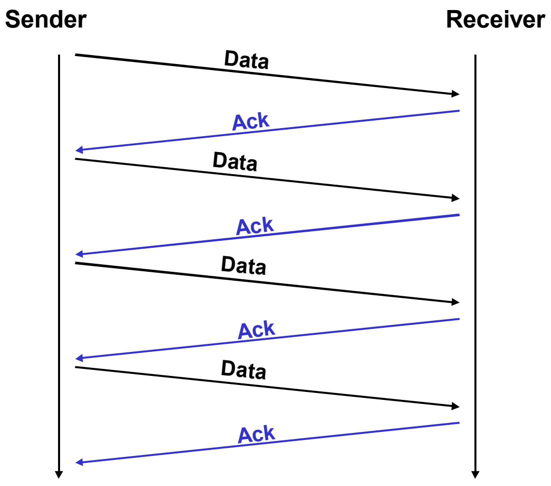
(C) Herbert Haas 2005/03/11 21

There are two major families of ARQ requests the idle-RQ system and the continuous-RQ system.

The continuous-RQ system consists of four different flavors. The Selective ACK, Positive ACK, GoBackN, the Selective Reject (SREJ) method and sometimes even a mix out of these basic methods.



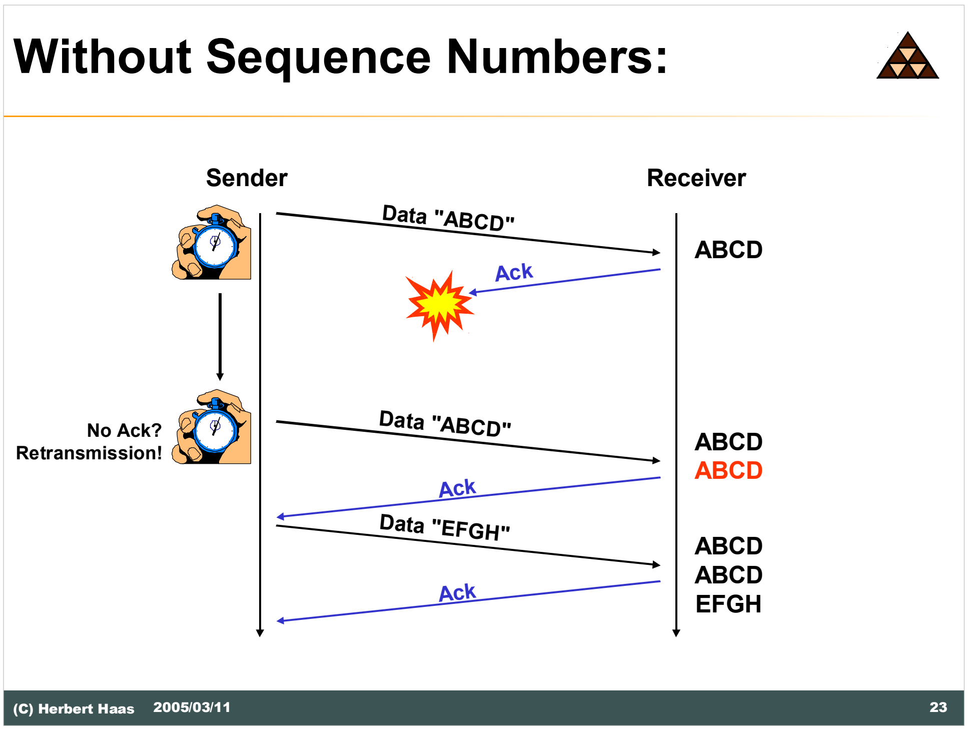
Idle-RQ



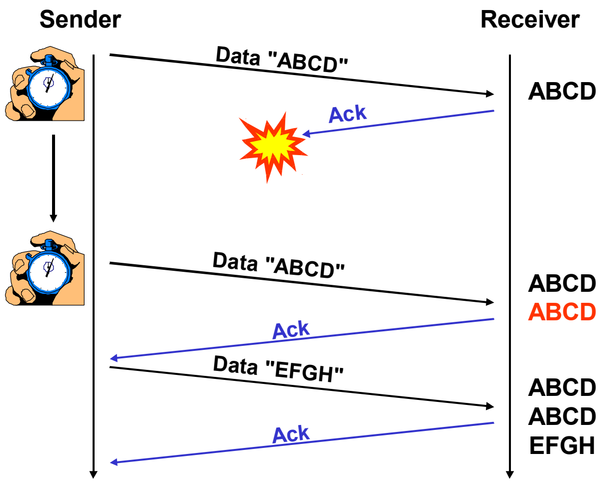
The idle-RQ technique is a stop and go protocol, this means when the sender has sent out one data frame he must wait for the according acknowledgement before he is allowed to sent the next frame.

The idle-RQ protocol operates in a half duplex mode and is typically used in master – slave environments. So the master sends a frame and must wait for the response of the slave whether the frame was properly received or not.

In today's networks the idle-RQ technique is seen very rarely, because it introduces large delays and is not able to fill data pipes we are currently used to.



Without Sequence Numbers:



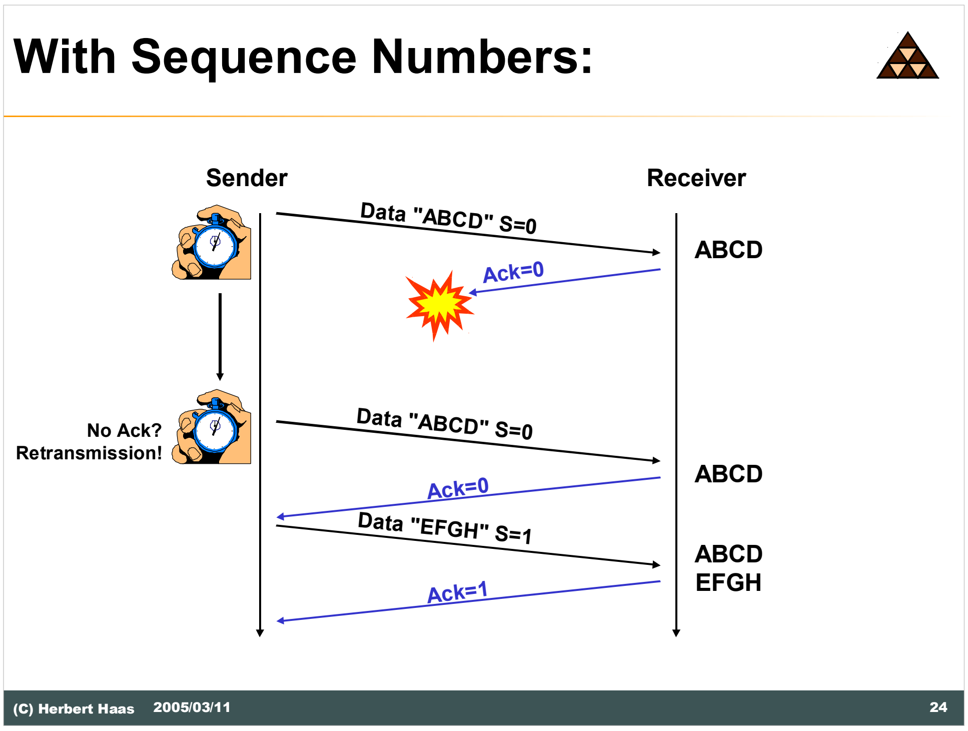
(C) Herbert Haas 2005/03/11 23

There are two different ways how idle-RQ technique might be implemented.

In this graphic an idle-RQ system without the use of sequence numbers is shown. The sender sends out a data frame, starts an retransmission timer and waits for the receive of an acknowledgement.

An already sent data frame remains in the send buffer and may only be deleted if an proper acknowledgement is received.

A data frame retransmission will happen, if the retransmission timer times out before an acknowledgement is received. Even if the transmission itself was successful and only the acknowledgement was lost. This could lead to an transport of duplicate data frames.

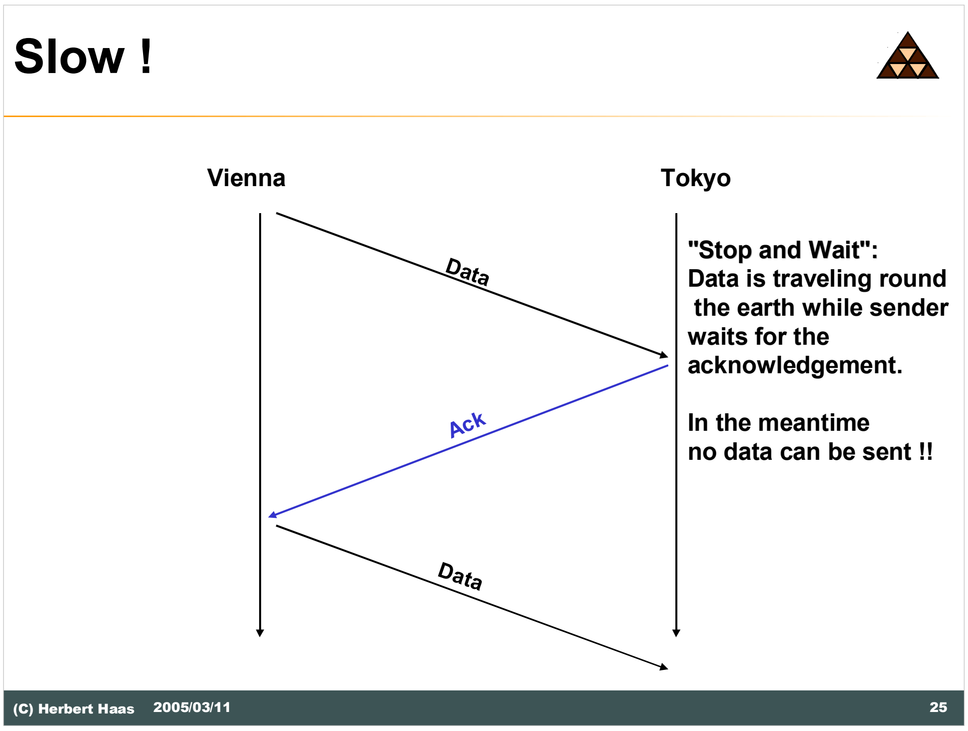


With Sequence Numbers:

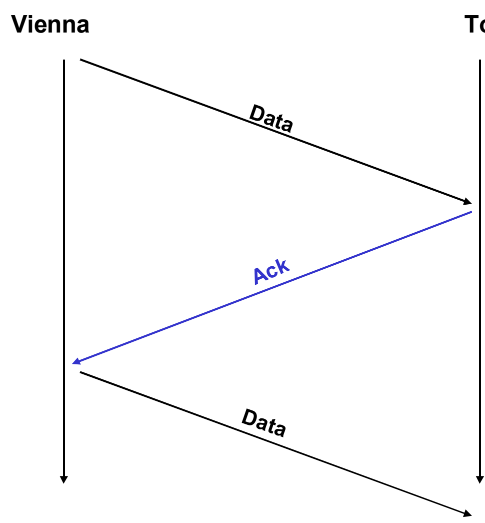
In the second scenario of idle-RQ systems sequence numbers are used.

In this case the sender sends out a data frame with a valid sequence number. Keeps the data frame in the send buffer and starts the retransmission timer. The acknowledgement frame is lost again and the data frame is retransmitted.

But now the receiver is able to recognize, by the help of the sequence number, that the received data frame is a duplicate and is able to discard it.



Slow !



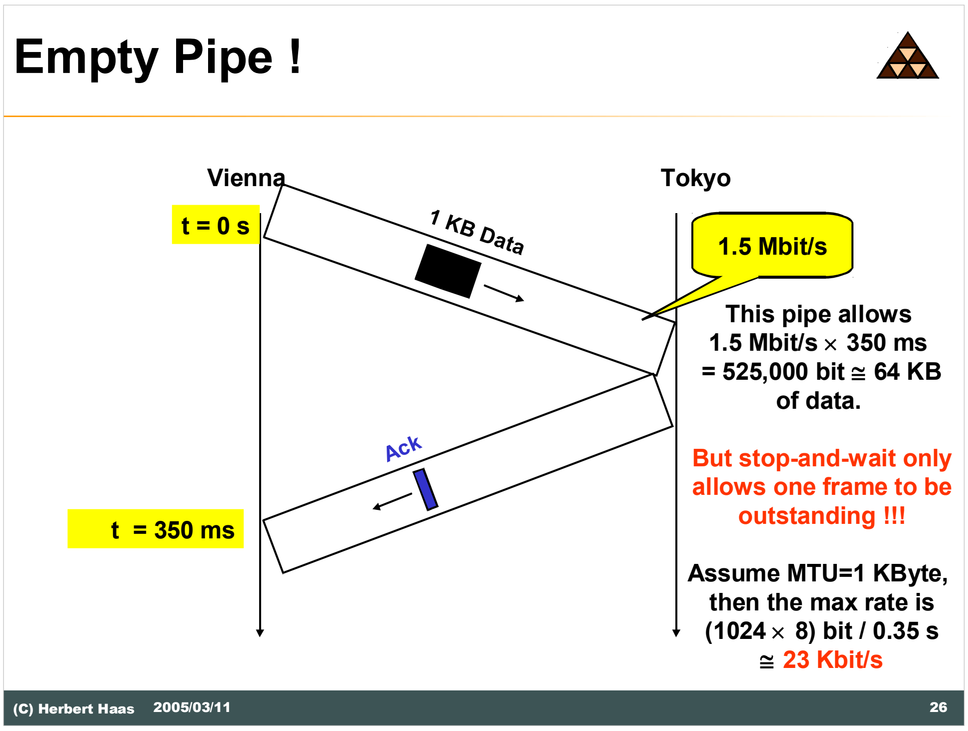
"Stop and Wait":
Data is traveling round the earth while sender waits for the acknowledgement.

In the meantime no data can be sent !!

(C) Herbert Haas 2005/03/11 25

As already mentioned before the idle-RQ technique is not suited for today's data transport systems. The stop and wait procedure would introduce a lot of delay, especially on long distance connections, and would no be able to efficiently use the network infrastructure.

Empty Pipe !



This pipe allows 1.5 Mbit/s × 350 ms = 525,000 bit ≅ 64 KB of data.

But stop-and-wait only allows one frame to be outstanding !!!

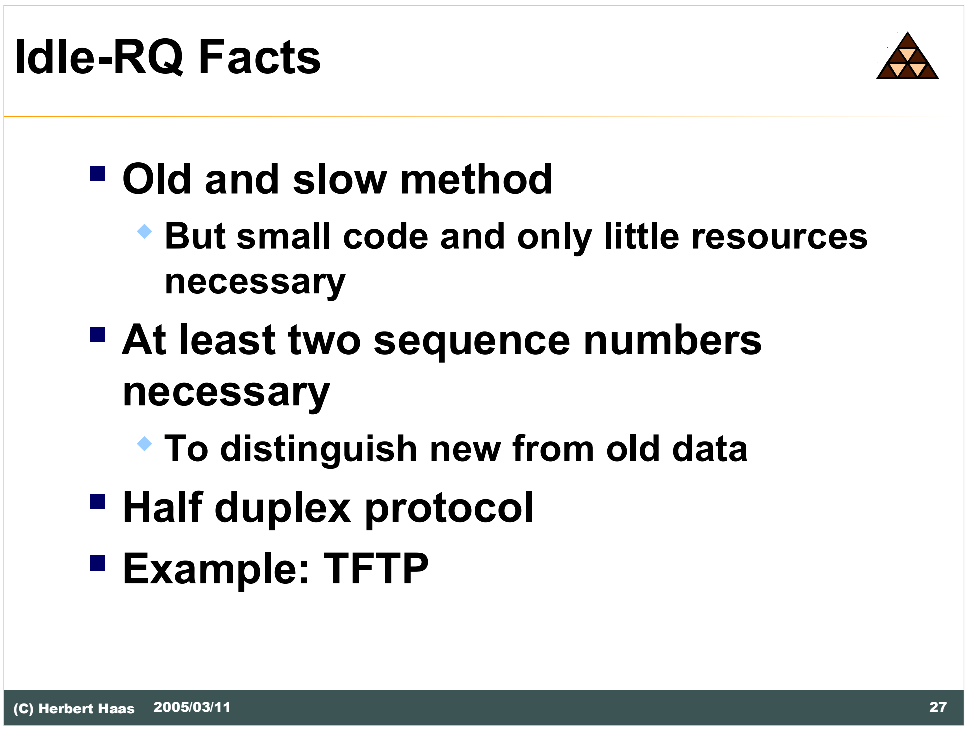
Assume MTU=1 KByte, then the max rate is (1024 × 8) bit / 0.35 s ≅ 23 Kbit/s

(C) Herbert Haas 2005/03/11 26

In this scenario we have a 1.5 Mbit/s connection between Vienna and Tokyo.

A 1 KByte data frame is sent from Vienna to Tokyo with a transport delay of 175 milliseconds in one direction. With the use of idle-RQ technique it would take at least 350 milliseconds before an acknowledgement is received and the next data frame is sent.

In this case a maximum throughput of only 23 Kbit/s can be achieved.



Idle-RQ Facts

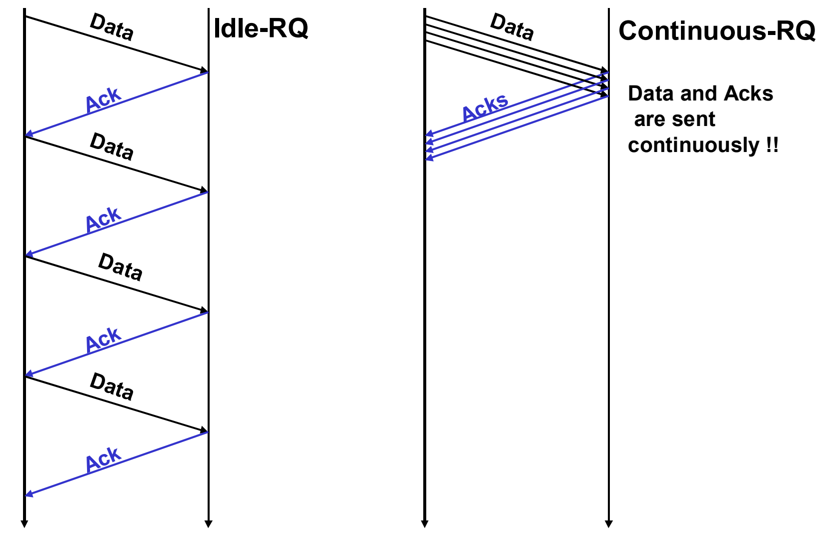
- Old and slow method
  - But small code and only little resources necessary
- At least two sequence numbers necessary
  - To distinguish new from old data
- Half duplex protocol
- Example: TFTP

(C) Herbert Haas 2005/03/11 27

The only advantage of the idle-RQ technique compared to the more sophisticated continuous RQ techniques is the little amount of memory and processor resources that is needed.

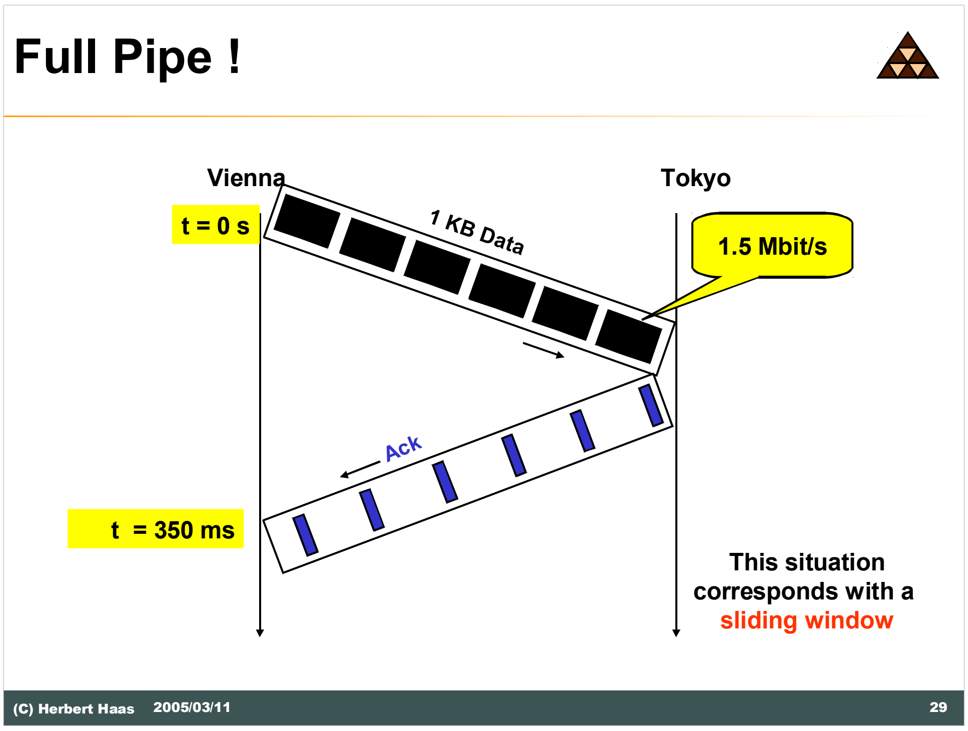
Therefore it is very easy to implement them in ROM based systems e.g. cisco router support TFTP from ROM monitor

Continuous RQ

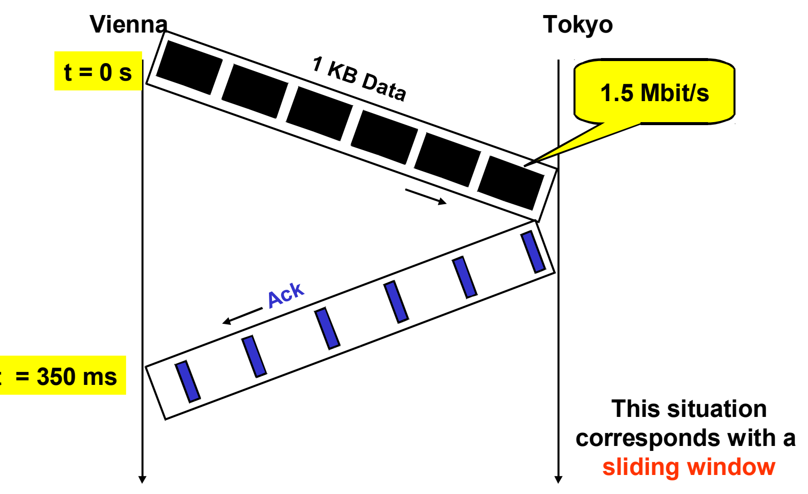


In continuous-RQ technology data frames and their according acknowledgements are sent continuously.

The sender is allowed to put a certain amount of frames into the send buffer and transmit them all in one go. The amount of frames the sender is allowed to send is either negotiated during the connection establishment phase or dynamically adjusted by max window size announcements of the received acknowledgements.



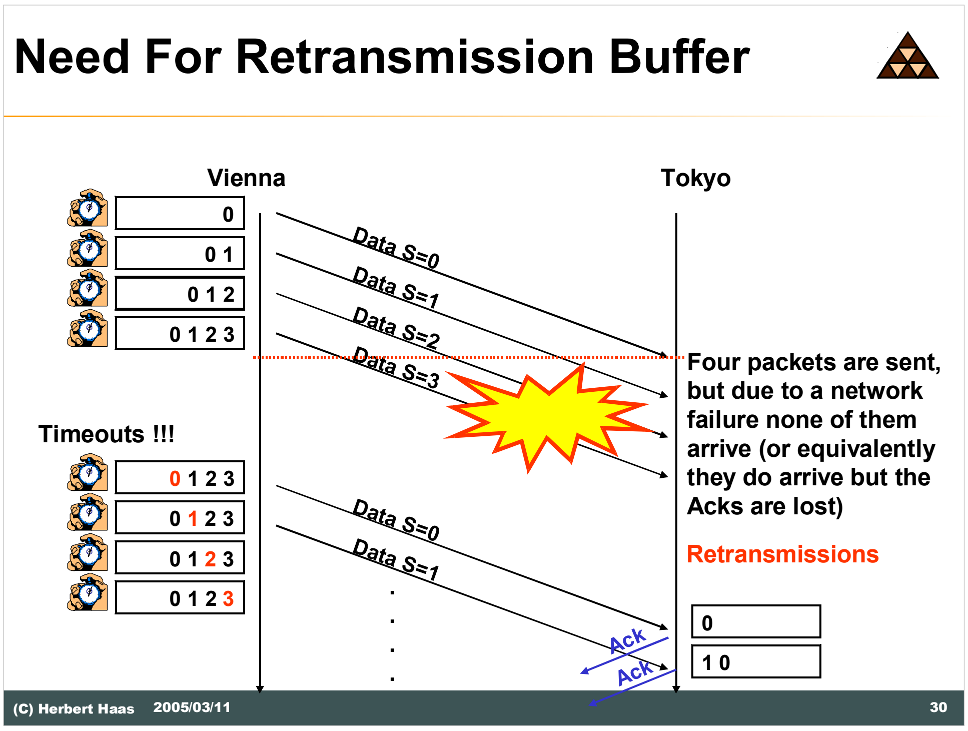
Full Pipe !



This situation corresponds with a sliding window

(C) Herbert Haas 2005/03/11 29

By the use of continous-RQ technique and an proper adjusted send window size it would now be possible to use the complete capacity of our Vienna – Tokyo connection.



Need For Retransmission Buffer

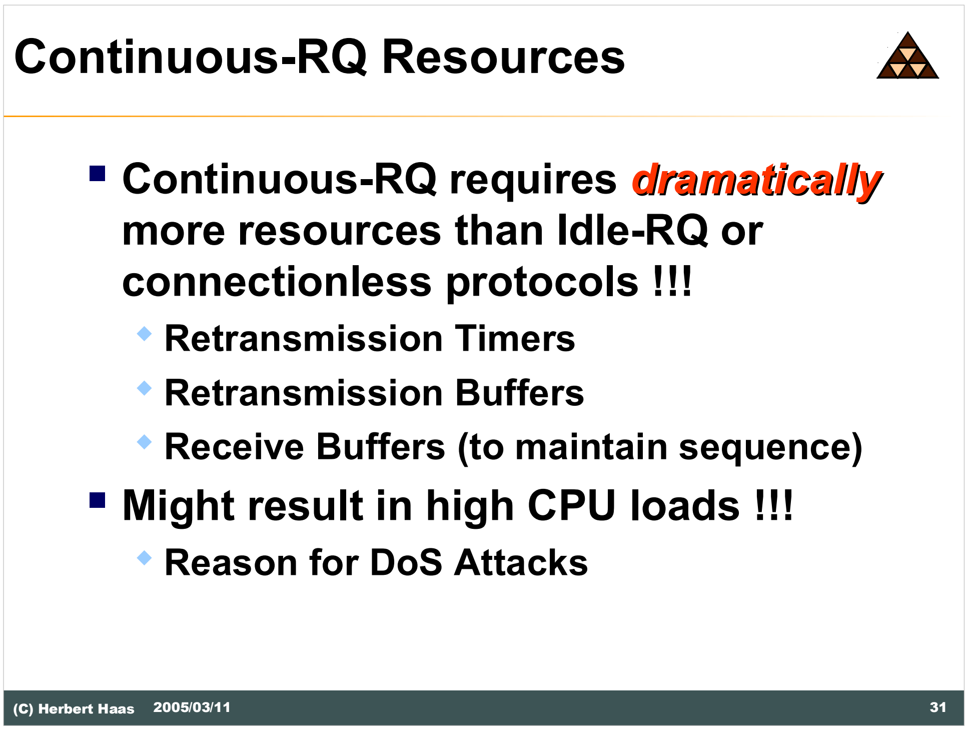
Four packets are sent, but due to a network failure none of them arrive (or equivalently they do arrive but the Acks are lost)

Retransmissions

(C) Herbert Haas 2005/03/11 30

In this example four data frames are sent from Vienna to Tokyo and all of them are lost or the according acknowledgements do not arrive.

This situation leads to an timeout of the retransmission timer in the Vienna location and causes a retransmission of all four data frames.



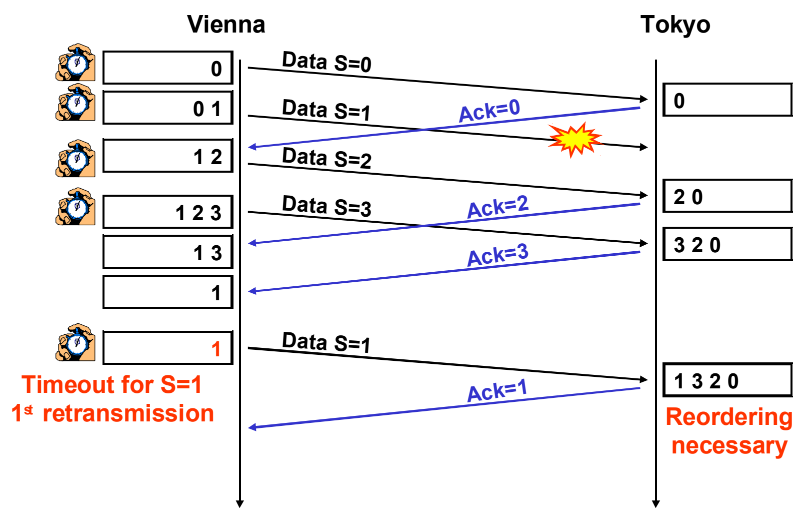
Continuous-RQ Resources

- Continuous-RQ requires *dramatically* more resources than Idle-RQ or connectionless protocols !!!
  - Retransmission Timers
  - Retransmission Buffers
  - Receive Buffers (to maintain sequence)
- Might result in high CPU loads !!!
  - Reason for DoS Attacks

(C) Herbert Haas 2005/03/11 31

Practically retransmission timers are started for groups of packets and not for each single packet. The retransmission timer calculation is based upon the measured round-trip-time (RTT) which may vary between milliseconds and minutes (Internet). Several more or less sophisticated algorithms adjust the retransmission timers in order to maximize overall data throughput.

Selective Acknowledgements

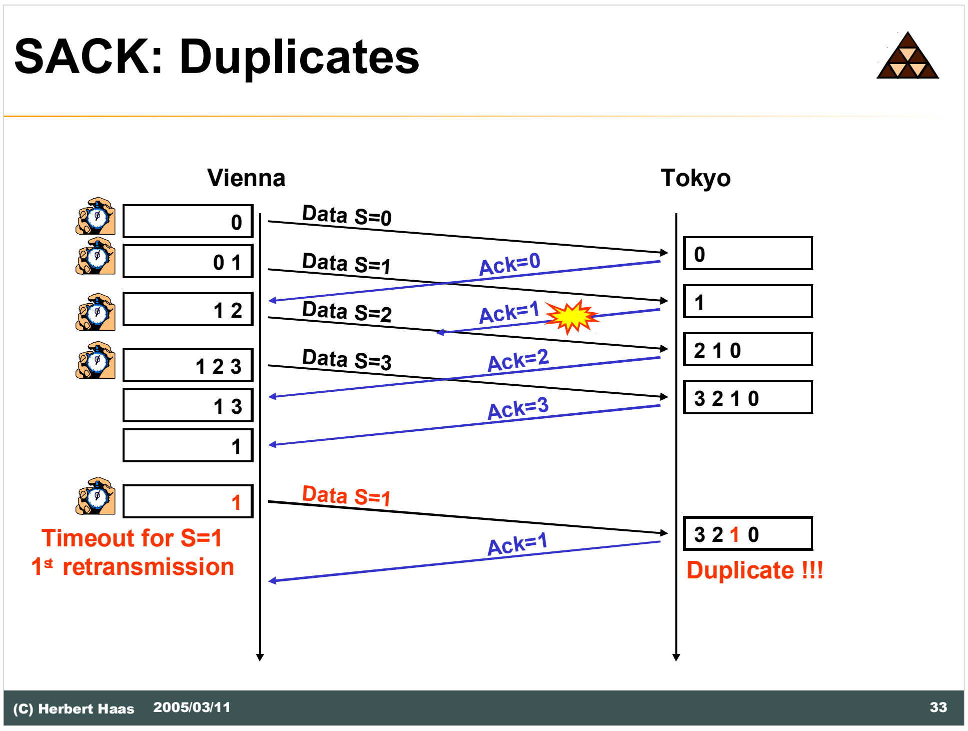


(C) Herbert Haas 2005/03/11 32

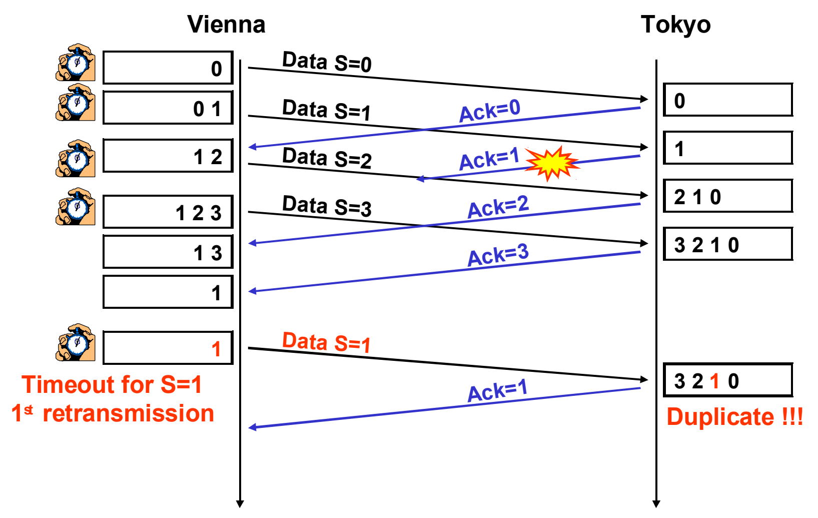
In Selective Acknowledgement technology every single frame that is sent out needs to be acknowledged. As soon as an acknowledgement arrives the according data frame may be deleted out of the send buffer.

If a data frame is lost no acknowledgement will be sent for this frame only. This causes again a retransmission timeout at the sender and leads to an retransmission of the data frame. But now the data frames are received out of order at the receiver side.

The receiver is now able to reorder the data frames by the help of the sequence numbers before they are handed over to the next higher level software process.



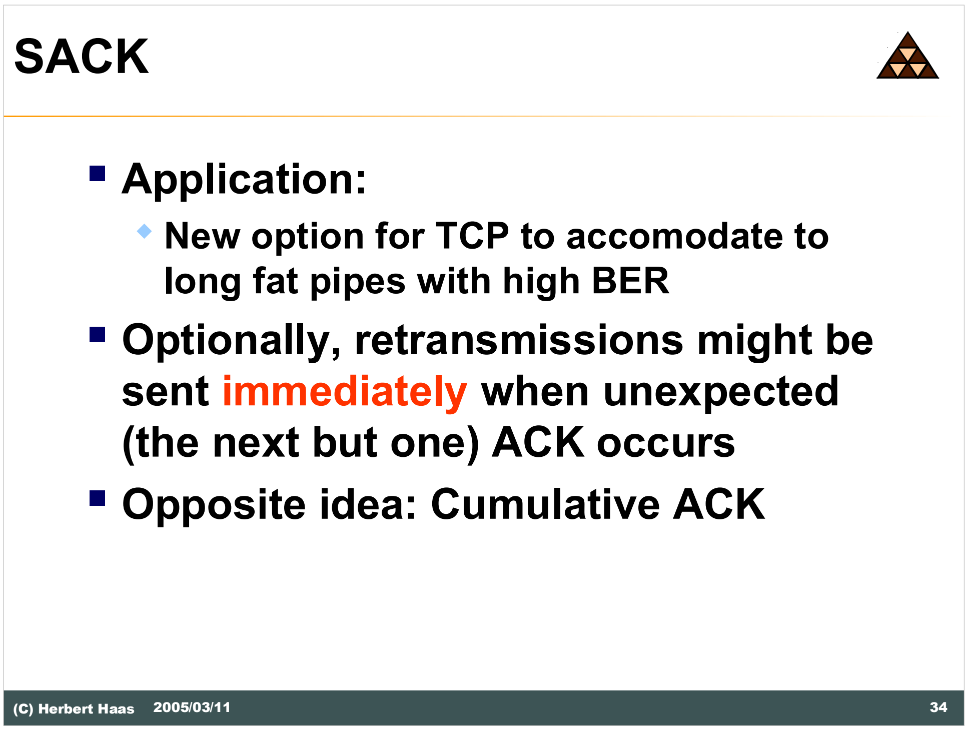
SACK: Duplicates



(C) Herbert Haas 2005/03/11 33

The Selective Acknowledgement technique causes retransmissions even in the case when only the acknowledgement frame is lost. Because if the Ack frame is missing the sender is not allowed to remove the according data frame from the send buffer.

But nevertheless duplicate data frames are recognized by the receiver with the help of the sequence numbers.



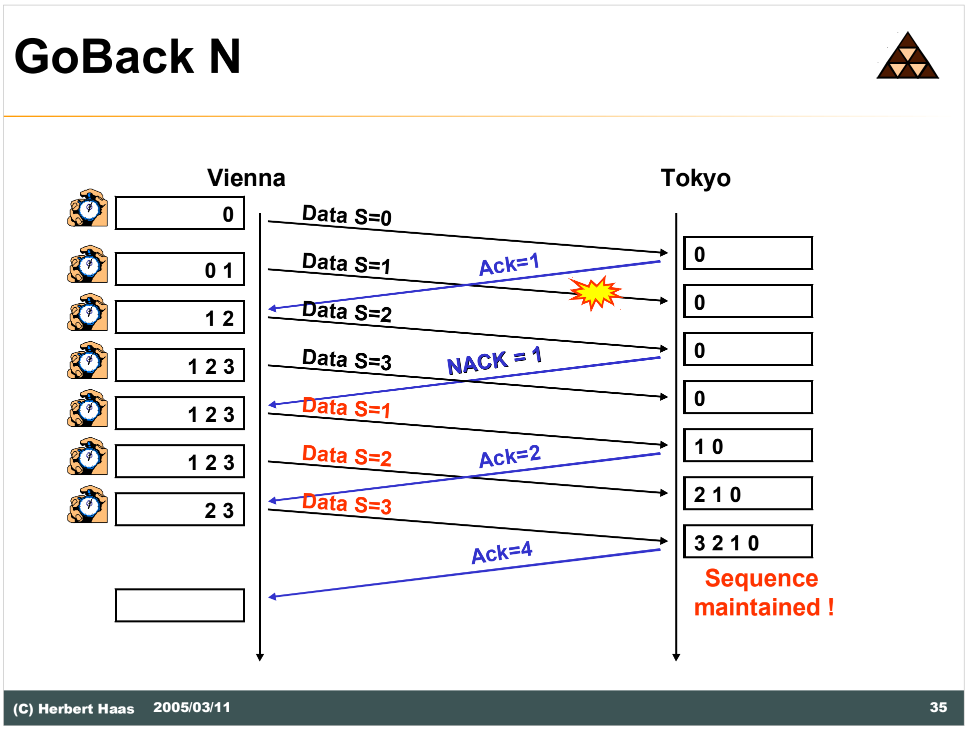
SACK

- **Application:**
  - **New option for TCP to accomodate to long fat pipes with high BER**
- **Optionally, retransmissions might be sent immediately when unexpected (the next but one) ACK occurs**
- **Opposite idea: Cumulative ACK**

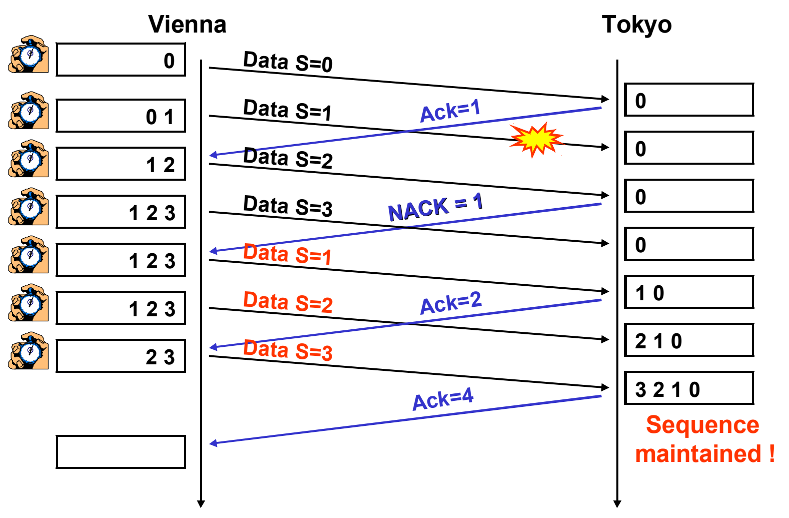
(C) Herbert Haas 2005/03/11 34

This SACK technology is nowadays also supported by modern TCP protocol stacks and its use is negotiated during connection setup.

Optionally retransmissions may occur immediately when an unexpected ACK is received.



GoBack N

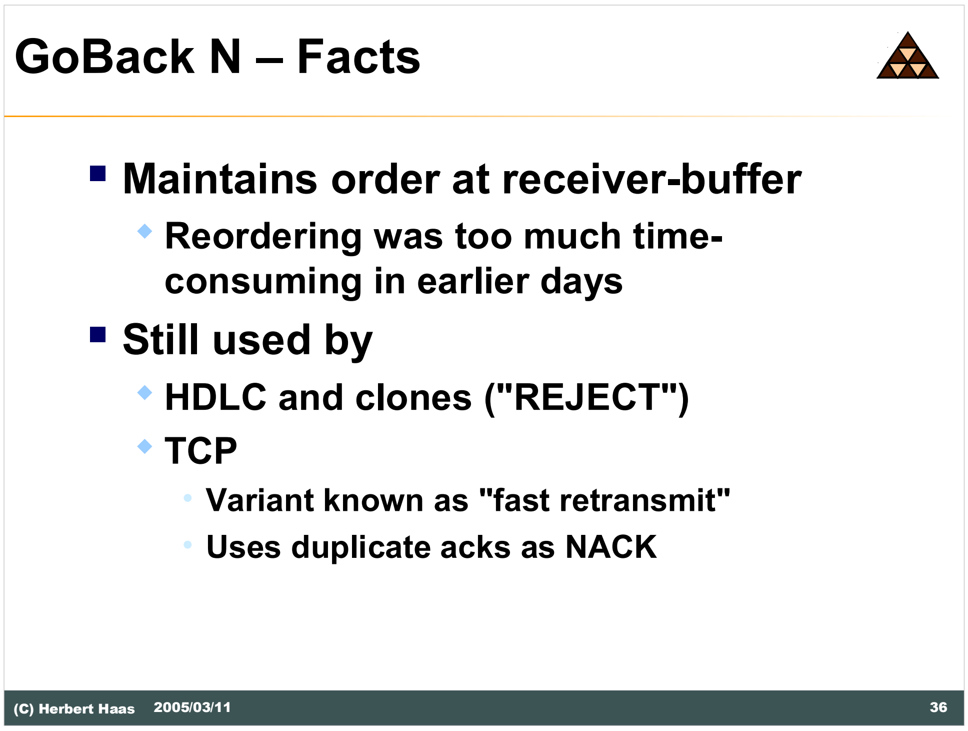


(C) Herbert Haas 2005/03/11 35

In GoBackN ARQ technique data frames are only acknowledged when the series of received sequence numbers is continuous.

In the above example the data frame with the sequence number S = 1 is lost. The receiver recognizes that a data frame is missing when he receives the frame with the sequence number S = 2. Now an so called Not Acknowledgement (NACK) data frame is sent back to the sender. The sender recognizes which frame was lost by the sequence number carried in the NACK data frame.

Starting with this sequence number all data frames are retransmitted. This causes more overhead than by the use of SACK but makes sure that the order of the data frames is maintained.

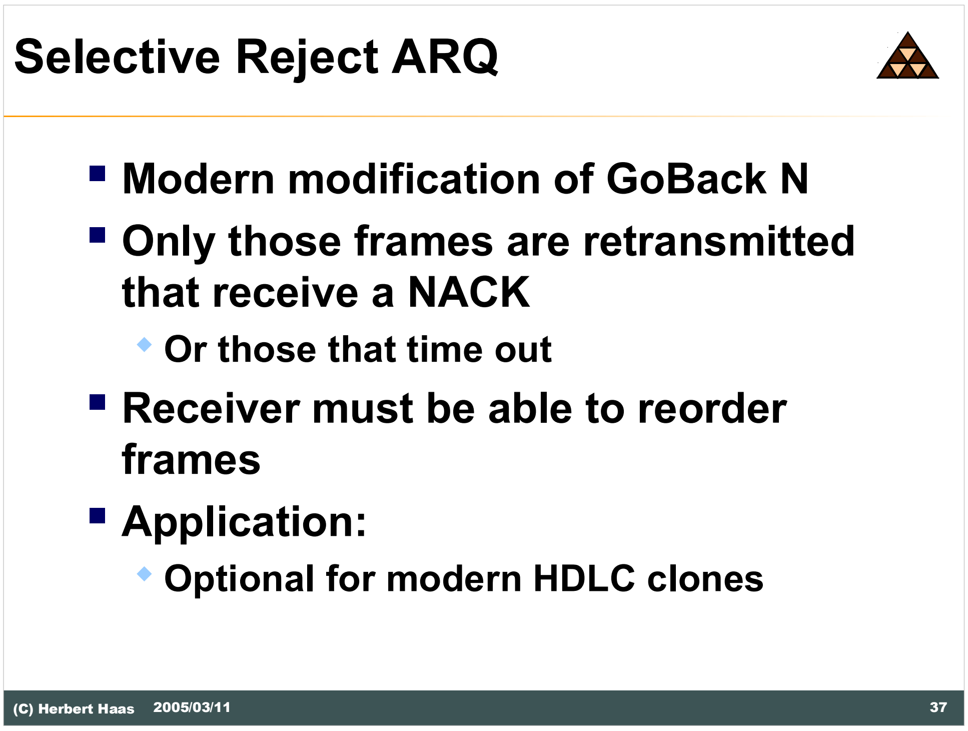

GoBack N – Facts

- Maintains order at receiver-buffer
  - Reordering was too much time-consuming in earlier days
- Still used by
  - HDLC and clones ("REJECT")
  - TCP
    - Variant known as "fast retransmit"
    - Uses duplicate acks as NACK

(C) Herbert Haas 2005/03/11 36

The GoBackN procedure is used by the quite old HDLC protocol because reordering of data packets was a too much time and memory consuming task for processors in the early days of data communication.

Also in today's TCP implementation a variant of the NACK procedure is used, the duplicate ACK. As soon as an TCP Stack recognizes that a data frame is missing it sends out an duplicate acknowledgement. So the sender recognizes that a data frame was lost and retransmits the according data frame. This speeds up the error recovery procedure because the retransmission timeout is omitted in his case.



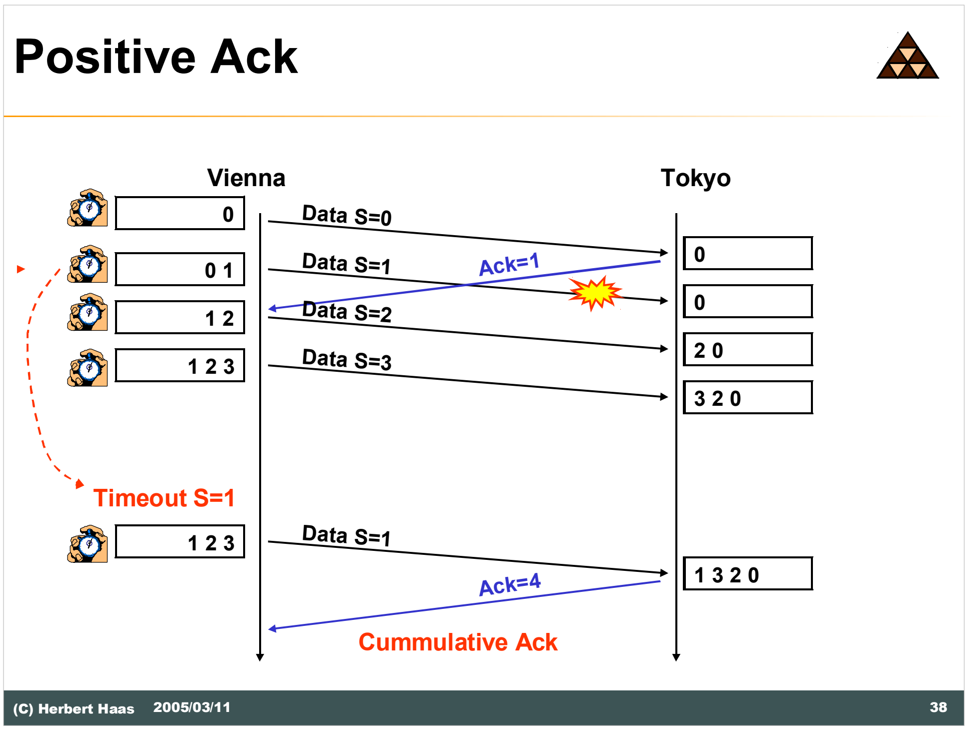
Selective Reject ARQ

- Modern modification of GoBack N
- Only those frames are retransmitted that receive a NACK
  - Or those that time out
- Receiver must be able to reorder frames
- Application:
  - Optional for modern HDLC clones

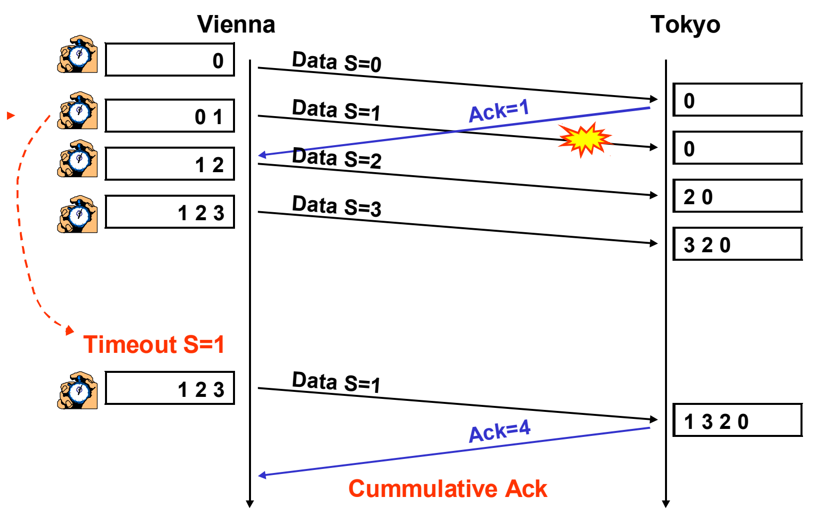
(C) Herbert Haas 2005/03/11 37

A combination of different ARQ procedures is the SACK procedure with the use of NACK frames. In this technique only data frames for which an NACK frame is received or those that time out will be retransmitted.

The correct order of the data frames might get lost.



Positive Ack



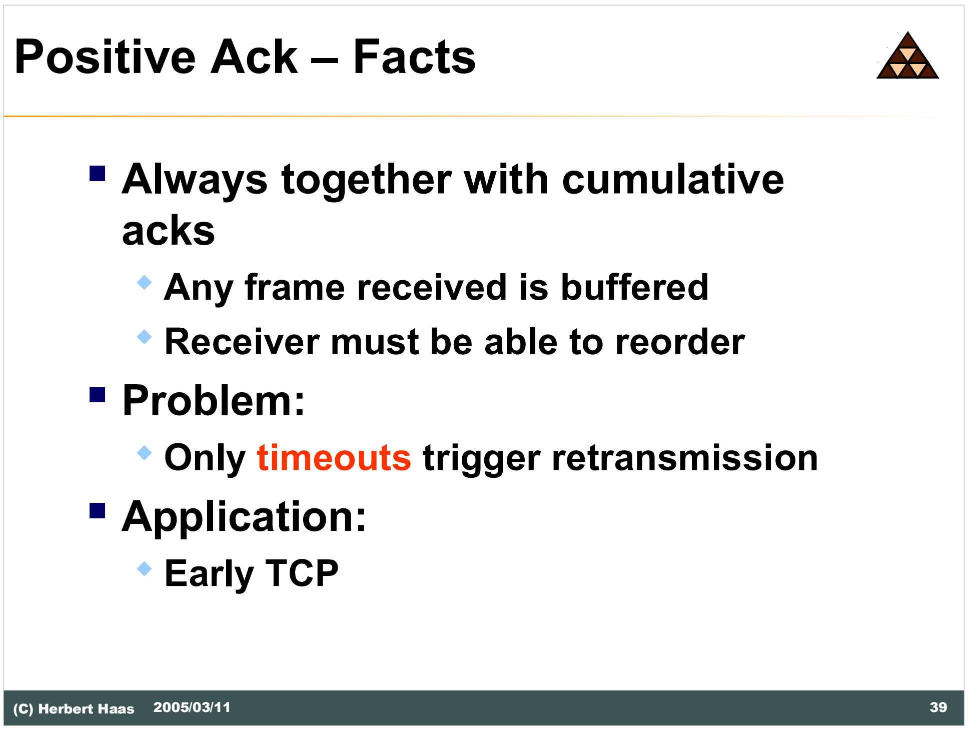
(C) Herbert Haas 2005/03/11 38

In the positive ARQ technique an ACK is only sent when the order of the arriving data frames is correct.

In the example above we find that the data frame with the sequence number S = 1 gets lost. The next frame that arrives is the data frame with S = 2. The receiver recognizes discontinuous sequence numbers and stops to transmit ACK frames. In the meantime all other frames are received and stored in the receive buffer but are not acknowledged.

When the data frame S = 1 falls into timeout, it is retransmitted. The receiver recognizes that the missing data frame has arrived and launches an ACK = 4 frame to acknowledge all till now received frames.

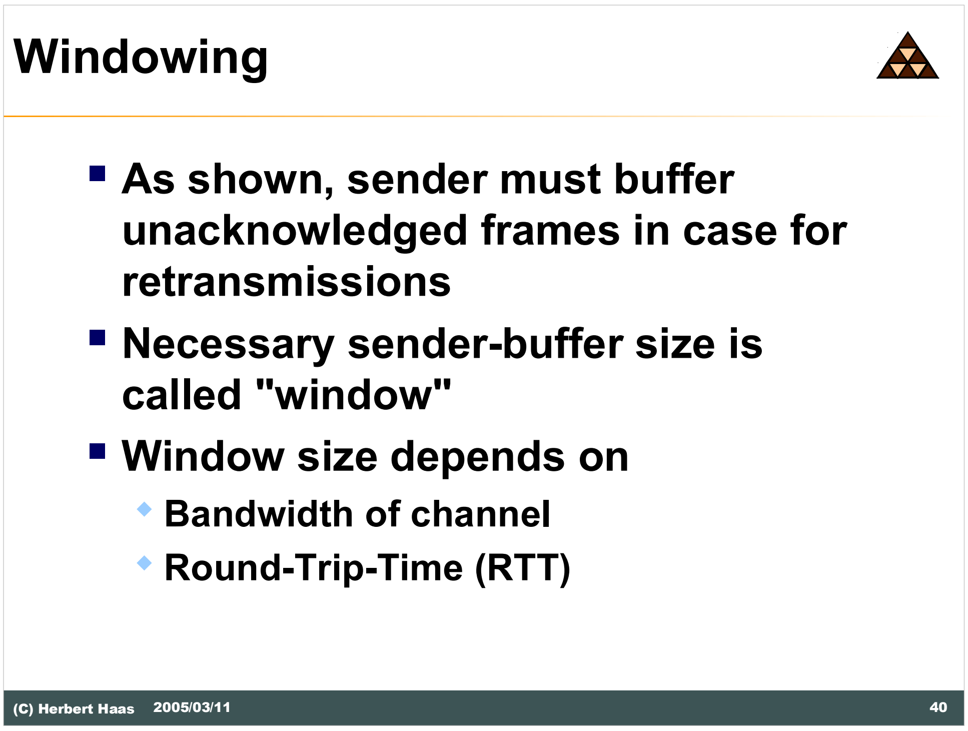
Obviously it may happen that another data frame is retransmitted depending on the remaining timeout period and the RTT of the connection.



Positive Ack – Facts

- Always together with cumulative acks
  - Any frame received is buffered
  - Receiver must be able to reorder
- Problem:
  - Only timeouts trigger retransmission
- Application:
  - Early TCP

The positive ACK procedure is always used together with cumulative acknowledgement technology. Disordering of data frames may occur and must be fixed. Only timeouts trigger retransmissions of lost data frames.



Windowing

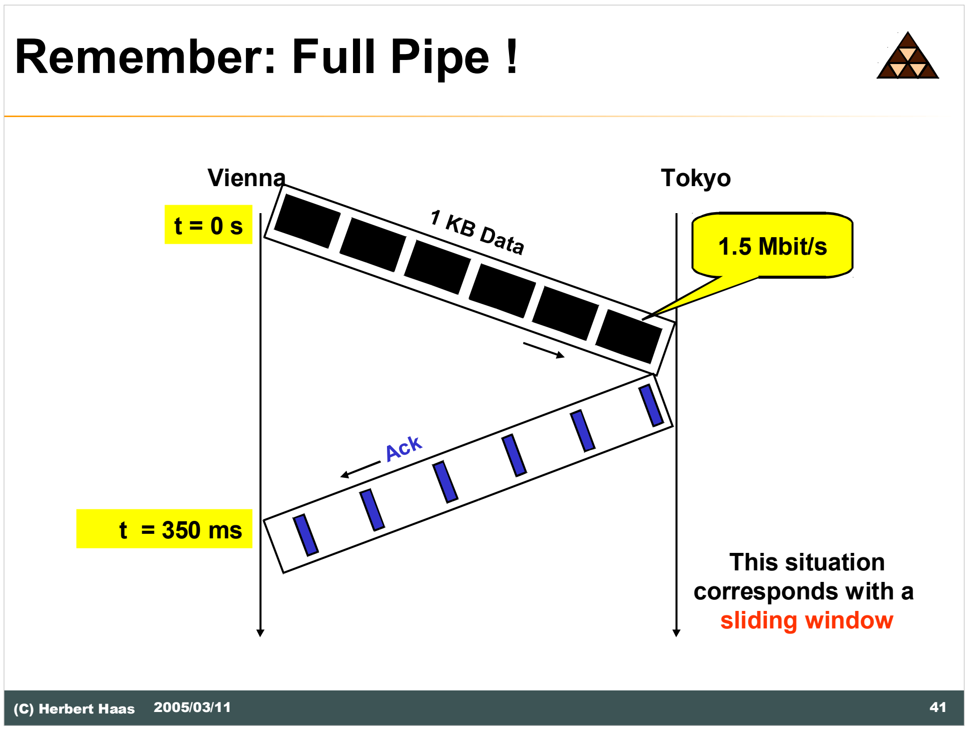
- As shown, sender must buffer unacknowledged frames in case for retransmissions
- Necessary sender-buffer size is called "window"
- Window size depends on
  - Bandwidth of channel
  - Round-Trip-Time (RTT)

(C) Herbert Haas 2005/03/11 40

The sender needs to buffer all transmitted frames until the according acknowledgement arrives. The size of this transmit buffer is called the window size.

The optimum window size directly depends on the bandwidth of the channel as well as on the Round-Trip-Time.

Elder protocols use a fixed window size negotiated during connection setup. More modern protocols such as TCP use a method called adaptive windowing which allows to automatically adapt the window size to needs of the transport system.

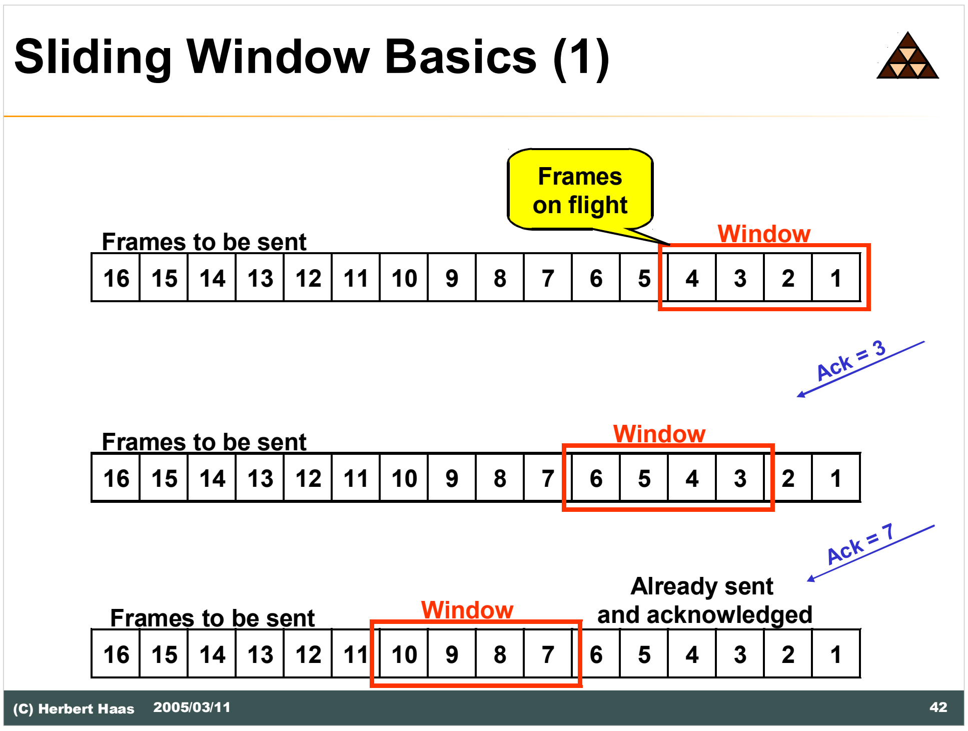


Remember: Full Pipe !

This situation corresponds with a sliding window

(C) Herbert Haas 2005/03/11 41

In this scenario a proper window size is used which guaranties a most efficient use of the transport capacity.



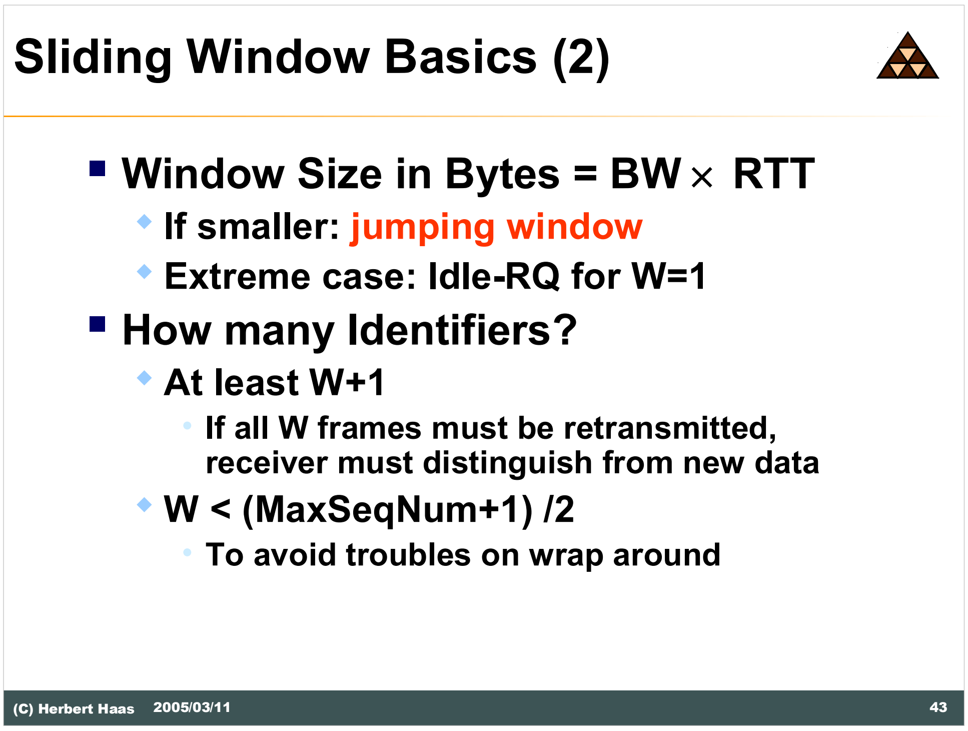
Sliding Window Basics (1)

(C) Herbert Haas 2005/03/11 42

In this example a fixed window size of four is used. This means that up to four data frames without explicit acknowledgments may be sent.

As soon as the first ACK = 3 arrives the sliding window moves to the left. This is possible because the data frames 1 and 2 are now removed out of the send buffer. Now the data frames 5 and 6 are sent and kept in the send buffer until their acknowledgements arrive.

On the left side of the window we find the data frames that will be sent in the future, while on the right side of the sliding window already sent and acknowledged frames can be found. So the window moves continuously from right to left therefore the name sliding window.



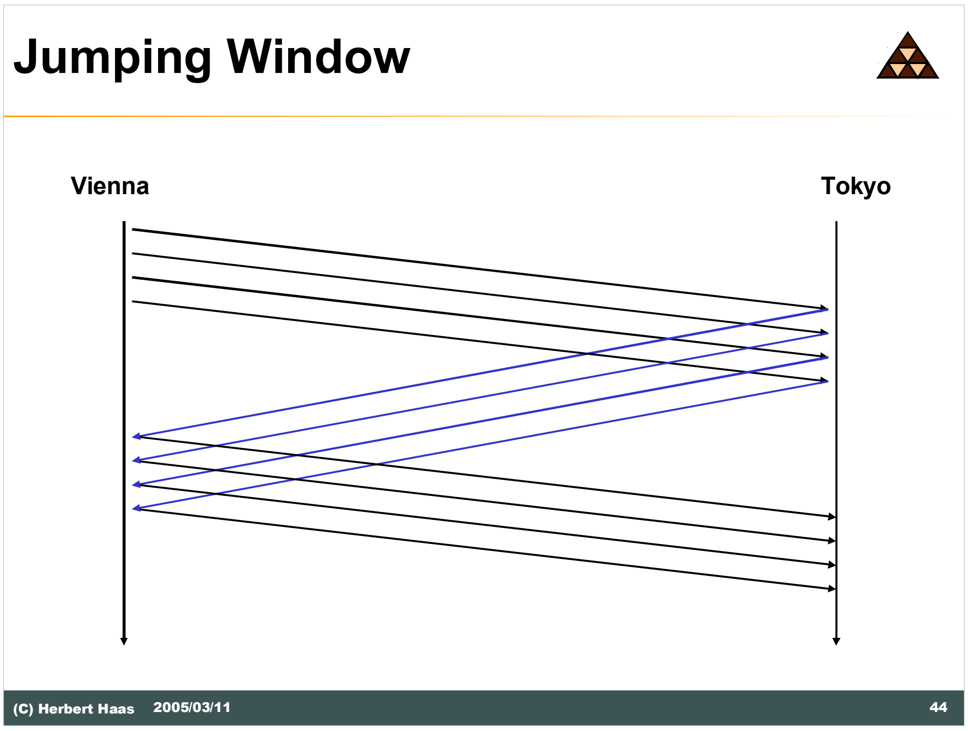
Sliding Window Basics (2)

- Window Size in Bytes = BW × RTT
  - If smaller: jumping window
  - Extreme case: Idle-RQ for W=1
- How many Identifiers?
  - At least W+1
    - If all W frames must be retransmitted, receiver must distinguish from new data
  - W < (MaxSeqNum+1) /2
    - To avoid troubles on wrap around

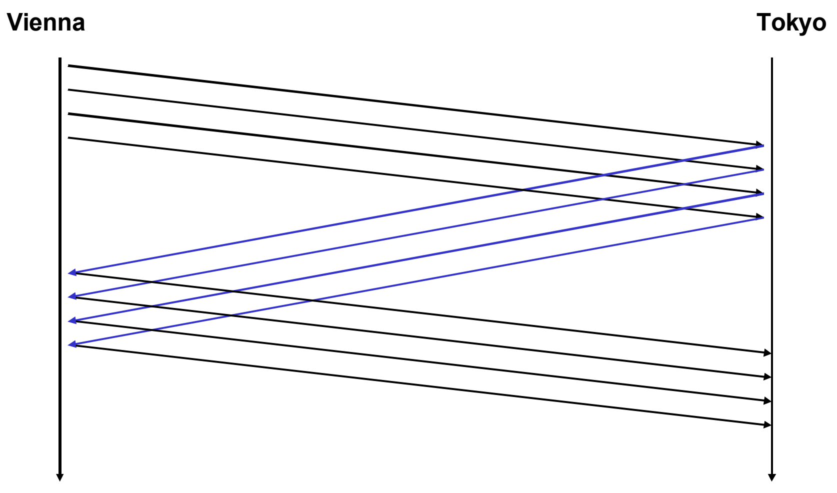
(C) Herbert Haas 2005/03/11 43

Assume we have a window size W=4 and the same number of identifiers (sequence numbers 0..3). When we send four frames at once and unfortunately all coresponding acknowledgements get lost, our timers expire and we must retransmit those frames. Now we again send frames 0..3 but the receiver cannot recognize them as the second incarnation. Thus, the first real new frame must have a sequence number of 4, that is the next frames have identifiers 4, 0, 1, 2. The result is that we need at least W+1 identifiers.

Even then, another tricky scenario might happen. Assume we have a sequence number space 0..7 (8 identifiers) and W=7. We send frames 0,1,2,3,4,5,6 and get all acknowledgements. Then we send 7, 0, 1, 2, 3, 4, 5. How can the receiver be sure that frames 0,1,2,3,4,5 are not retransmitted frames? (frame 7 might get lost!)
Because of this, we use a smaller window size W=4. Now we send frames 0,1,2,3 and get all acknowledgements. Then we send 4,5,6,7. The receiver is not confused.
So the final rule requires W to be smaller than half the number of identifers.



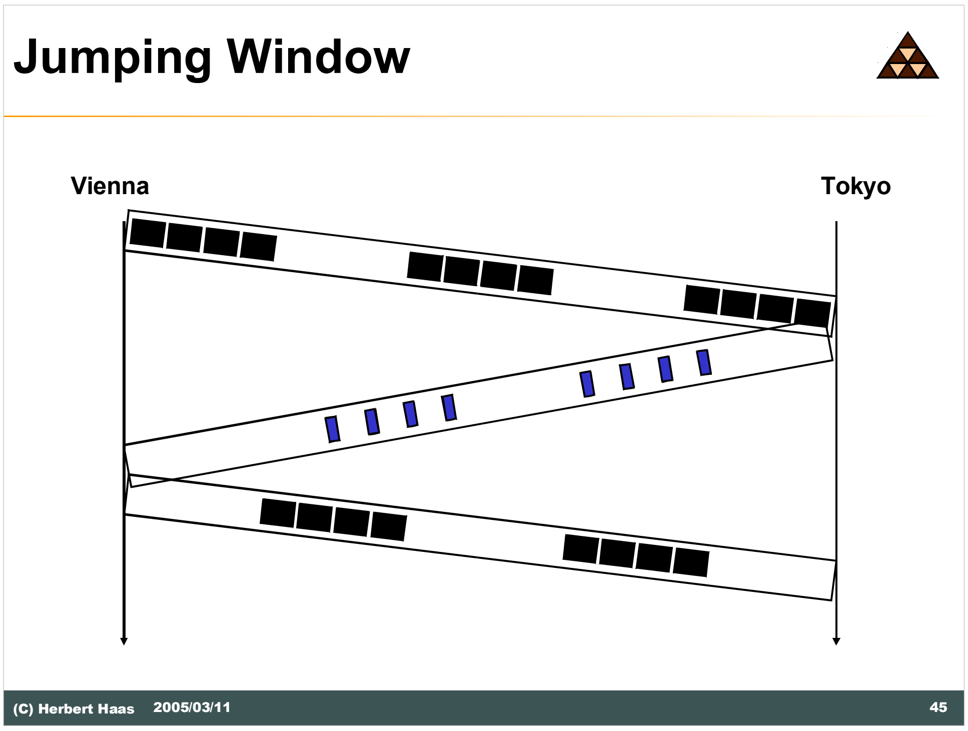
Jumping Window



In this example we find what will happen if the window size is to small.

The sender in Vienna has sent out all data frames until he reached the max window size. Now the sender in Vienna has to wait for acknowledgments to arrive.

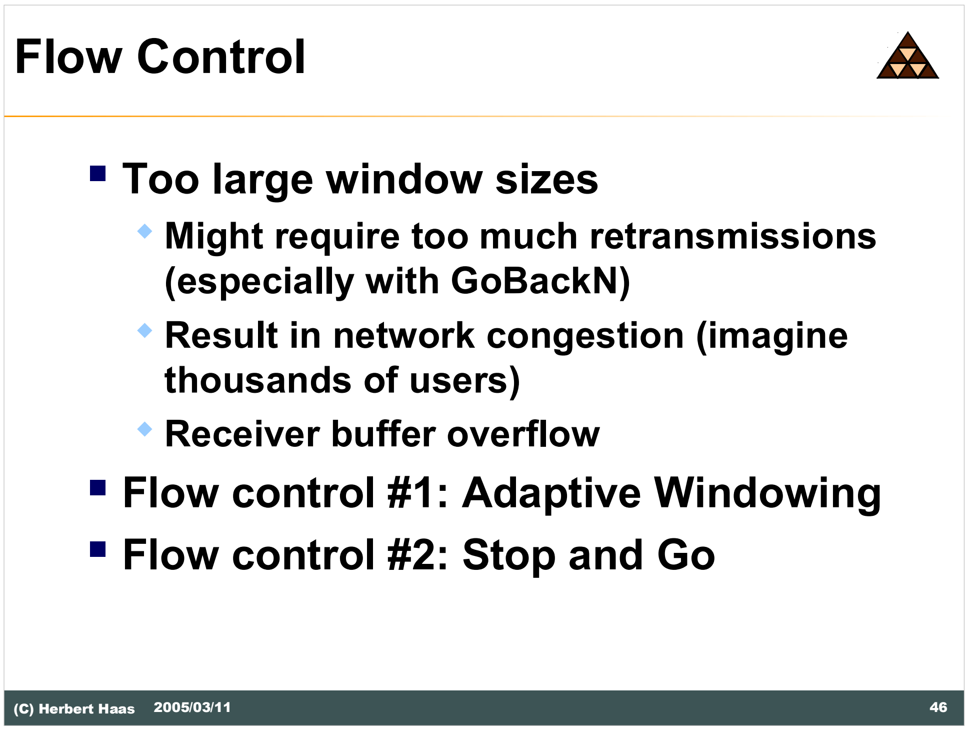
The incoming acknowledgements free up buffer space and the sender may continou to send. In this scenario the chosen window size is obviously to small leading to an insufficient use of the transport capacity.



Jumping Window

(C) Herbert Haas 2005/03/11 45

Evidently, this specific situation depicted above, can only occur when the senders employs a dynamic window size (assume the sender started with W=20 and suddenly reduced the window size to W=4.



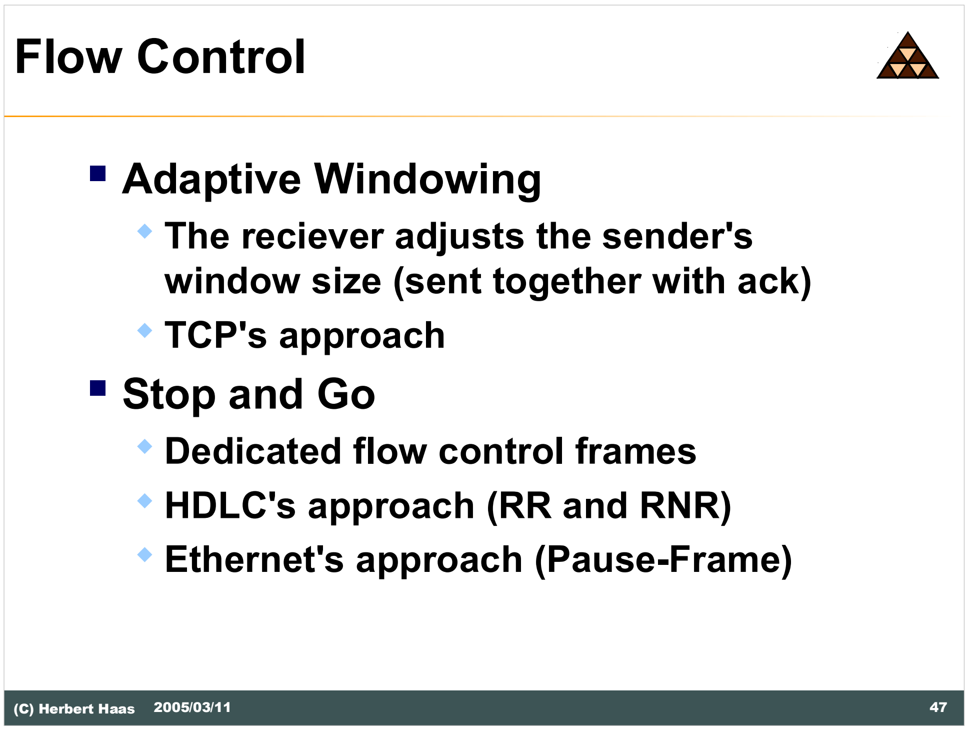
Flow Control

- Too large window sizes
  - Might require too much retransmissions (especially with GoBackN)
  - Result in network congestion (imagine thousands of users)
  - Receiver buffer overflow
- Flow control #1: Adaptive Windowing
- Flow control #2: Stop and Go

(C) Herbert Haas 2005/03/11 46

Imagine our window size is W=1,000,000 and after sending the 1,000,000th frame the receiver sends us a NACK(1). That is we have to repeat 999,999 frames, although most of them might have been transmitted without problems.

This would cause congestions in the network as well as buffer overflow problems at the receiver side.



Flow Control

- Adaptive Windowing
  - The reciever adjusts the sender's window size (sent together with ack)
  - TCP's approach
- Stop and Go
  - Dedicated flow control frames
  - HDLC's approach (RR and RNR)
  - Ethernet's approach (Pause-Frame)

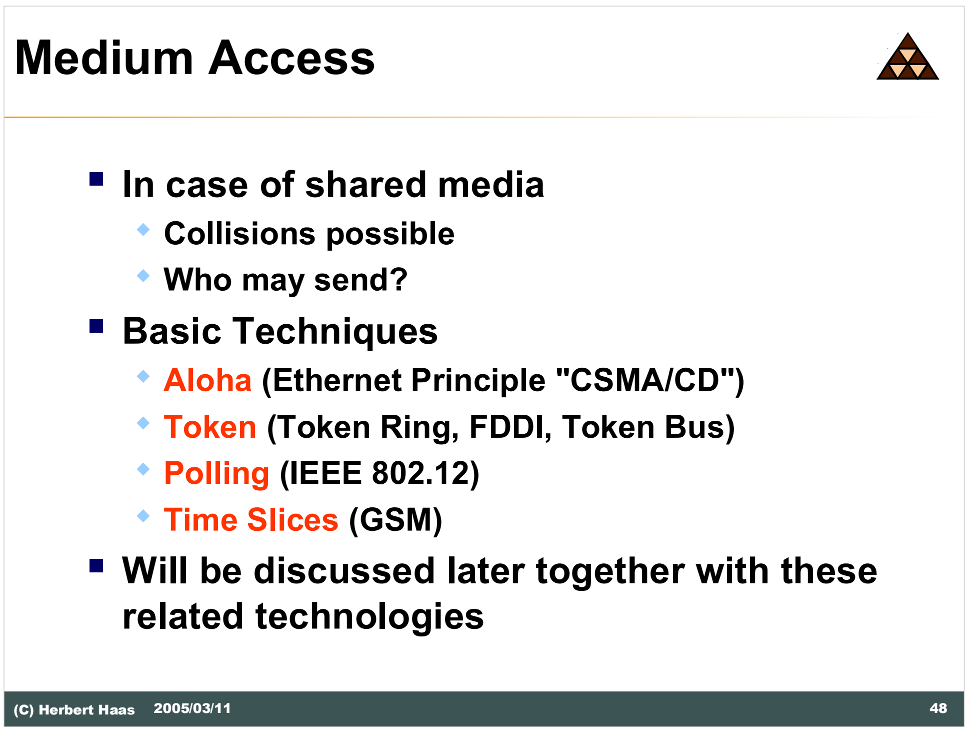
(C) Herbert Haas 2005/03/11 47

Flow control to prevent buffer overflow conditions at the receiver side can be implemented either by the use of adaptive windowing or by the use of some kind of stop and go procedure.

Adaptive windowing technique, which is used by the TCP protocol stack, announces the receivers available buffer size within the acknowledgements that are send back to the sender side. So the window size changes dynamically dependent on the buffer conditions at the receiver.

The Stop and Go technique, which is mainly used by some elder protocols like HDLC and its derivates, uses some special control frames Receiver Ready (RR) and Receiver Not Ready (RNR). These protocols typically are using a fixed window size of 7 or 127 packets.

There is also a new approach 802.3x to implement flow control in Ethernet technology which allows an congested device to specify transmission pauses in terms of bit times.



Medium Access

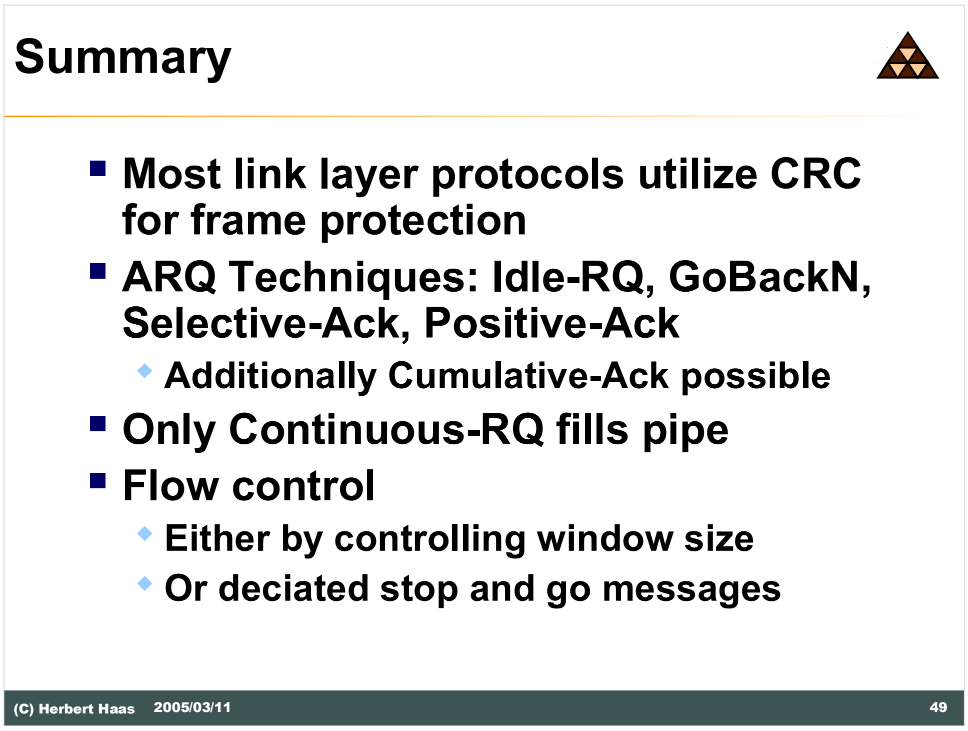
- In case of shared media
  - Collisions possible
  - Who may send?
- Basic Techniques
  - Aloha (Ethernet Principle "CSMA/CD")
  - Token (Token Ring, FDDI, Token Bus)
  - Polling (IEEE 802.12)
  - Time Slices (GSM)
- Will be discussed later together with these related technologies

(C) Herbert Haas 2005/03/11 48

The access onto the physical media depends whether the medium needs to be shared between many different users or if the usage is reserved for a single user only.

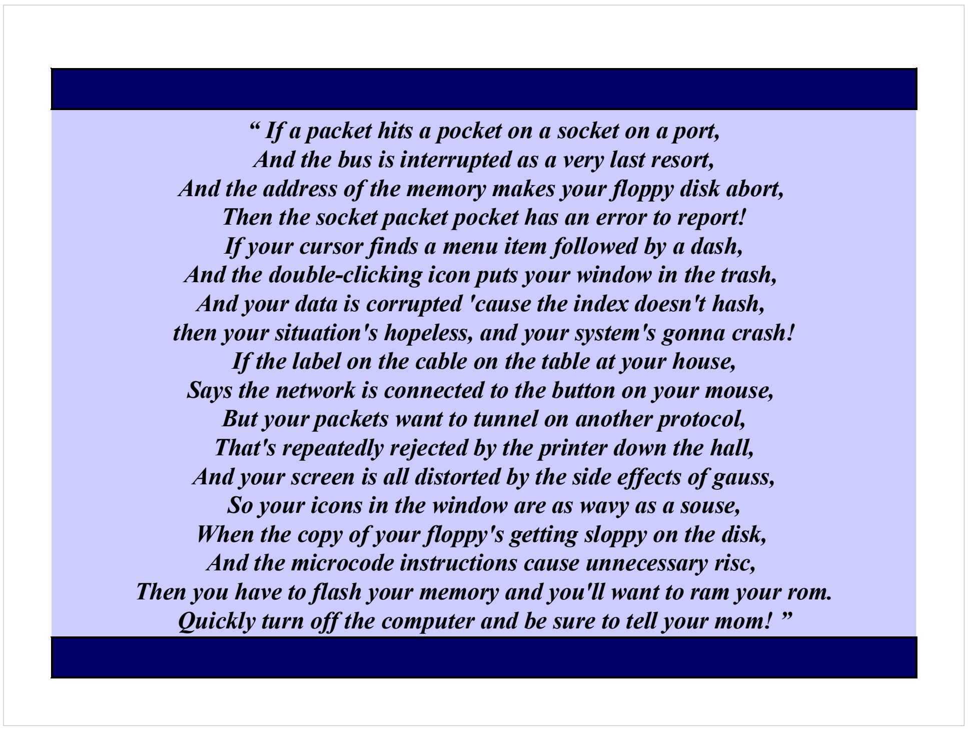
In the case of single user connections e.g. serial connections, ISDN circuits etc the medium access is grab it if you need it.

In shared medium environments many different technologies had been developed to control the access onto the medium e.g. CSMA/CD, Token controlled, Polling, Time Slices etc.



Summary

- Most link layer protocols utilize CRC for frame protection
- ARQ Techniques: Idle-RQ, GoBackN, Selective-Ack, Positive-Ack
  - Additionally Cumulative-Ack possible
- Only Continuous-RQ fills pipe
- Flow control
  - Either by controlling window size
  - Or deciated stop and go messages



“ If a packet hits a pocket on a socket on a port,
And the bus is interrupted as a very last resort,
And the address of the memory makes your floppy disk abort,
Then the socket packet pocket has an error to report!
If your cursor finds a menu item followed by a dash,
And the double-clicking icon puts your window in the trash,
And your data is corrupted 'cause the index doesn't hash,
then your situation's hopeless, and your system's gonna crash!
If the label on the cable on the table at your house,
Says the network is connected to the button on your mouse,
But your packets want to tunnel on another protocol,
That's repeatedly rejected by the printer down the hall,
And your screen is all distorted by the side effects of gauss,
So your icons in the window are as wavy as a souse,
When the copy of your floppy's getting sloppy on the disk,
And the microcode instructions cause unnecessary risc,
Then you have to flash your memory and you'll want to ram your rom.
Quickly turn off the computer and be sure to tell your mom! ”

Quiz

- What's the problem when putting IP-packets directly onto ISDN?
- What are Hamming-Codes?
- What maximum bit-rate can TCP-hosts utilize when connected via satellite?
- Explain why windowing protocols are more prone to DoS-attacks