angefertigt von M. Holzhey

0.1 Glossar

Ab.	Erläuterung
ABP	Alternating Bit Protocol
PDU	Protocol Data Unit
ARQ	Automatic Repeat Request
ARQ(SR)	Selective Repeat(errornous will retransmit)
GBN	Go-Back-N
RNR (frame)	Recieve not Ready
RR (frame)	Recieve Ready
HDLC	High Level Data Link Control
OFDM	Orthogonal Frequency Division Multiplexing
ISI	Intersymbol Interference
FHSS	Frequency Hopping Spread Spectrum
CDMA	Code Division Multiple Access (at UMTS and GPS)
CSMA	Carrier Sense Multiple Access
CSMA/CD	Carrier Sense Multiple access Collision Detection
UMTS	Universial mobile Telecommunication System
DSSS	Direct Sequence Spread Spectrum
PN	Pseudonoise
FDM	Frequency Division Multiplexing
TDM	Time Division Multiplexing
TDD	Time Division Duplex
FDD	Frequency Division Duplex
GSM	Global System for Mobile Communications
GPRS	General Packet Radio Service
EDGE	enhanced datarates for GSM
FEC	Forward Error Correction
TCP	Transmission Control Protocol
CRC	Cyclic Redundancy Check
SONET	Synchronous Optical Network
PDH	Plesiochronous Digital Hierachy
SDH	Synchronous Digital Hierachy
STS-1	Synchronous Transport Signal -level 1

Ab.	Erläuterung
POTS	Plain Old Telepone System
MAU	Medium Attachment Unit
POP	Point of Presence
IXC	Inter Exchange Carriers
LATA	Local Acess and Transport Area
DSL	Digital Subscriber Line
SAP	Service Acces Point
SDU	Service Data Unit
MAC	Medium Acess Control
DTE	Data Terminal Equipment
DCE	Data Carrier Equipment
CTS	Clear to Send (Handshake, RS-232)
USART	Universial Synchronous Asynchronous Receiver and Transmitter
SnW	Send and Wait protocol (part of ARQ)
GbN	Go-Back-N (part of (continous) ARQ)
SR	Selective Repeat (part of (continous) ARQ)
ATM	Asynchronous Transfer Mode
OSI	Open System Interconnection Model
ABM	Asynchronous Balanced Mode
FRMR	Frame Reject
AAL	ATM Adaption Layer
VPC	Virtual Path connection
VCC	Virtual Channel Connection
PPP	Point to Point Protocol
ADSL	Asymmetric Digital Subscriber Line
ISDN	Integrated Services Digital Network
LLC	Logical Link Control Sublayer
CSMA/CR	CSMA collision resolution
CSMA/CA	CSMA collision avoidance
LAN	Local Area Network
MAN	Metropolitan Area Network
SD	Start Frame Delimiter
ED	End Frame Delimiter
FS	Frame Status IEEE 802.5 (token ring)
RPR	Resilent Packet Ring
RPC	Remote Procedure Calls
LPC	Local Procedure Calls
DNS	Domain Name System
TLD	Top level Domain (server)

Ab.	Erläuterung
URL	Uniform ressource Locator
FDB	Forwarding Data Base
DHCP	Dynamic Host Configuration Protocol
SAP	Service Access Point
BGP	Border Gateway Protocol
IHL	IP header length
TTL	Time to live
ICMP	Internet Control Message Protocol
ISP	Internet Service Provider
CIDR	Classless InterDomain Routing
ARP	Address Resolution Protocol
NAT	Network Address Translation
ICMP	Internet Control Message Protocol
MTU	Maximum Transfer Unit
MN	Mobile Node
HA	Home Agent
FA	Foreign Agent
COA	Care-of Address (tunnel end-point of the MN)
CN	Correspnding Node
QoS	Qualtity of Service
SDL	Specification and Description Language
FSM	Finite State Machines
AIMD	Additive Increase Multiplicative Decrease
ISN	Initial Sequence Number
PAWS	Protect Agains Wrapped Sequence number
OSPF	Open Shortest Path First
ISIS	Intermediate System to Intermediate System Protocol
LSP	Link State Packet: OSPF
IGBP	Internal BGP
EGBP	External BGP
MSS	Maximum Segment Size (TCP)
SACK	Selective ACK (TCP)
RTO	Retransmission Timeout (TCP)
(P)GPS	(Packetised) General Processor Sharing
WFQ	Weighted Fair Queueing
RSVP	Reservation Protocol
ASN.1	Abstract syntax Notation
HTML	Hypertext Markup Language
SNMP	Simple Network Management Protocol

Kapitel 1

Vorlesung

Es folgen, i. d. R. nach Units von Prof. Wolisz gegliederte, stichpunktartige Zusammenfassungen, der in der Vorlesung besprochenen Inhalte.

Inhaltsverzeichnis

1.1 Formeln und Zusammenhänge

- Limits of Transmission:

$$P_s - a_f \cdot x - S_f > 0$$

  - $P_s :=$ Senderleistung
  - $a_f :=$ Medium attenuation for unitary distance [db/km]
  - $S_f :=$ Receiver Sensivity

- Dezibel

$$[db]a_f = 10 \cdot \log\left(\frac{P_s}{P_r}\right) = 20 \cdot \log\left(\frac{A_s}{A_r}\right)$$

  - $dbm := A_R = 1mW$
  - $dbW := A_r = 1W$

- Attenuation and Delay distortion in telephone line
- 64kbit/s telephone

$$2 \cdot 4kHz \cdot 8Bit$$

- propagation path attenuation: (low frequencies cover greater distance with equal power at Transmitter)

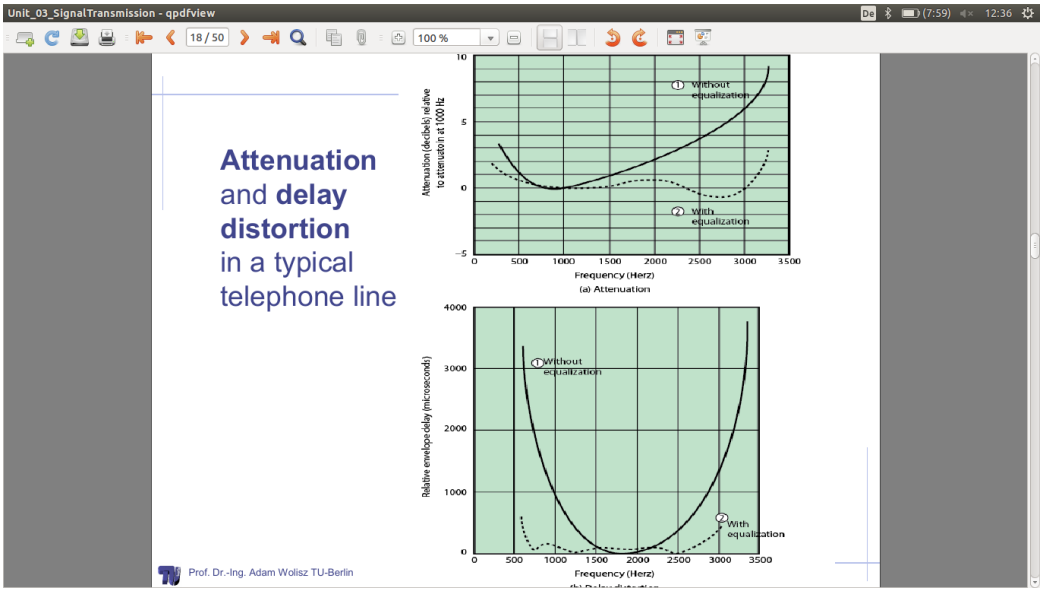
$$P_r = \frac{P_t \lambda^2}{(4\pi d)^2}$$

- propagation inhomogenous space

$$P_r = \frac{P_0}{d^\alpha}$$

  for office $\alpha = [3..4]$

- Digital Signal Encoding - Def

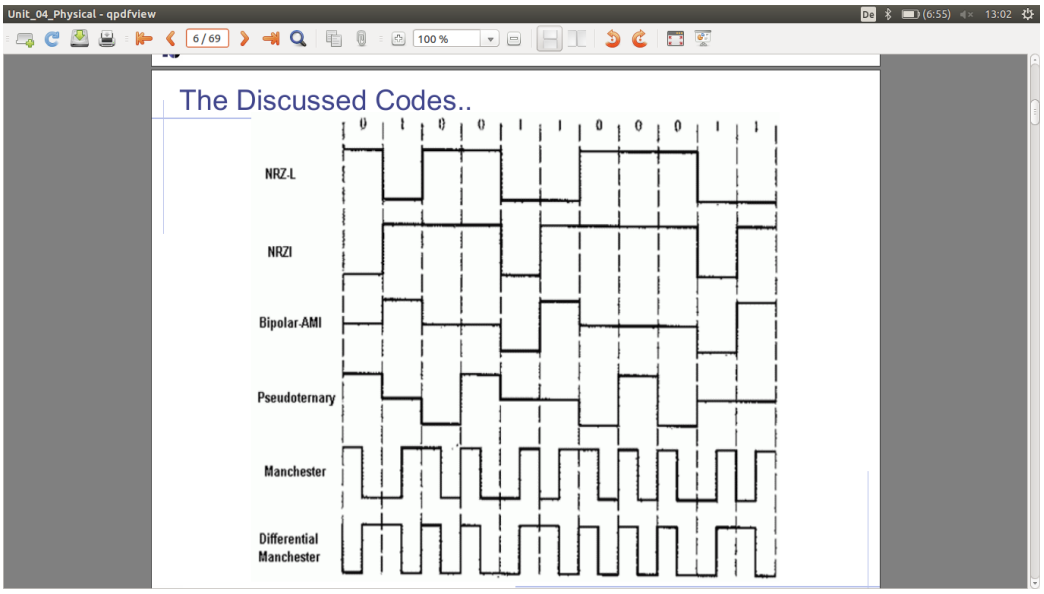


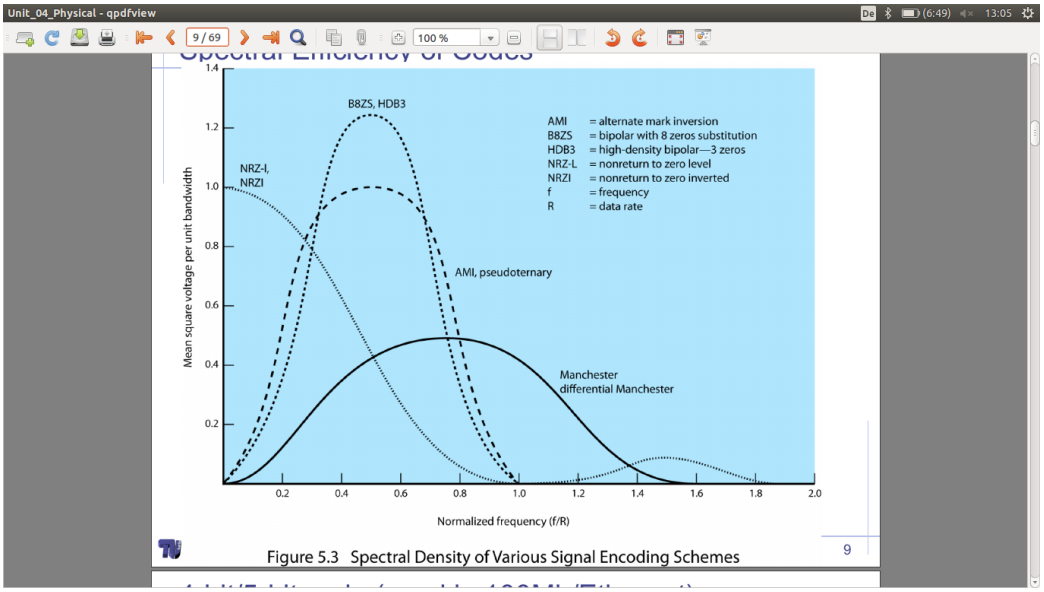
  - M = number of signal levels
  - within signal Period T - K changes are of M are possible: $\frac{K}{second} =$ Baudrate
  - M^k signal changes possible: $N < M^k$ signal changes permitted
  - achievable Bitrate

$$R = \frac{\lg_2 N}{T}$$

  - Bitrate not equal Baudrate

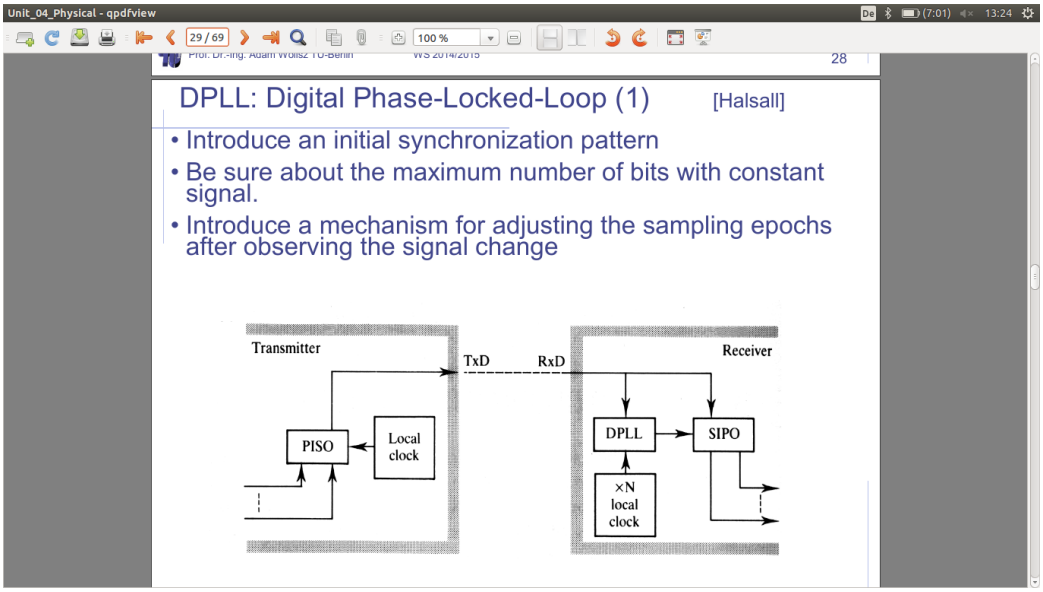
- discussed codes
- Spectral efficiency of codes
- Criteria for selection
  - lack of high freqeuncy and DC components lead to concentrated power mid-bandwith
  - clock reconstruction - pll
  - error detection feature
  - noise immunity
- Asynchronous Transmission:
  - good for long gaps between signals (keyboard)


Unit_04_Physical - qpdfview
6 / 69
100 %
The Discussed Codes..
0 1 0 0 1 1 0 0 0 1 1
NRZ-L
NRZI
Bipolar-AMI
Pseudoternary
Manchester
Differential Manchester


Unit_04_Physical - qpdfview
9 / 69
100 %
B8ZS, HDB3
NRZ-I, NRZI
AMI, pseudoternary
Manchester
differential Manchester
AMI = alternate mark inversion
B8ZS = bipolar with 8 zeros substitution
HDB3 = high-density bipolar—3 zeros
NRZ-L = nonreturn to zero level
NRZI = nonreturn to zero inverted
f = frequency
R = data rate
Mean square voltage per unit bandwidth
Normalized frequency (f/R)
Figure 5.3 Spectral Density of Various Signal Encoding Schemes
9

  - simple, cheap
  - overhead 2,3 Bit per character (20%) for synchronisation in case of Transmission
  - at least one signal change per symbol
  - clock at rxd considerably faster
  - clock rate equals Baudrate 2^N times n=[0..3,4,..]
  - the faster clock to bitrate the more centered is the sampling pulse

- Synchronous Transmission: fillers to hold connection
- PLL



- Bandwith for given Bitrate
  - multilevel schemes provoke e.g. 3 bits per symbol (=Baudrate)
  - Nyquist result: No noise: using M signal level with given Bandwith B

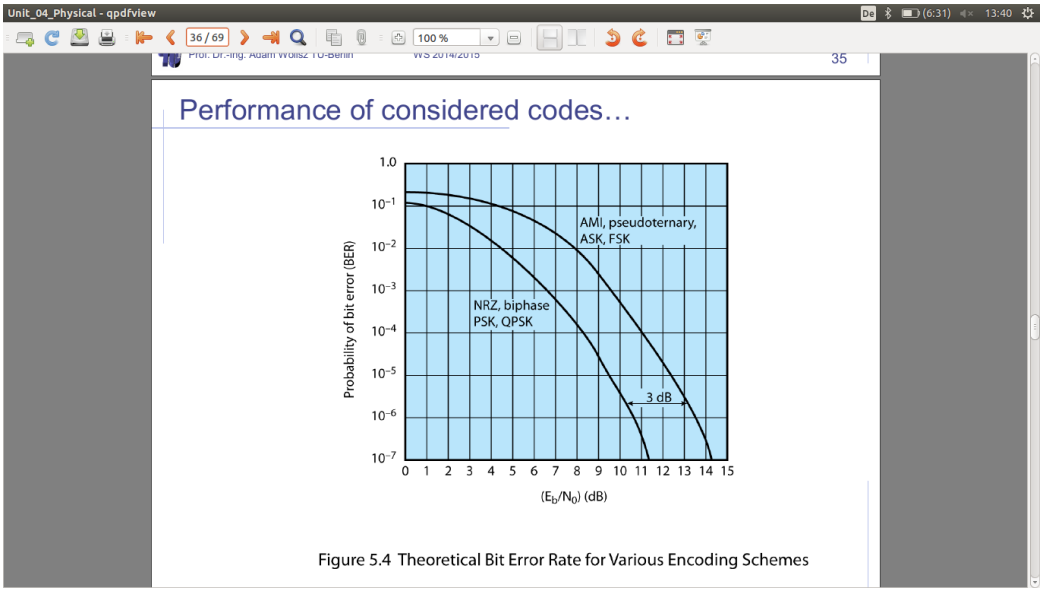
$$C[bps] = 2B \log_2 M$$

  - $\rightarrow$ donnot increase Signallevel due to noise $\rightarrow$ equals signallevel due to limited power ressource $\rightarrow$ errors
  - Shannon: Link capacity with noise (more real)

$$C[bps] = B \log_2 (1 + SNR)$$

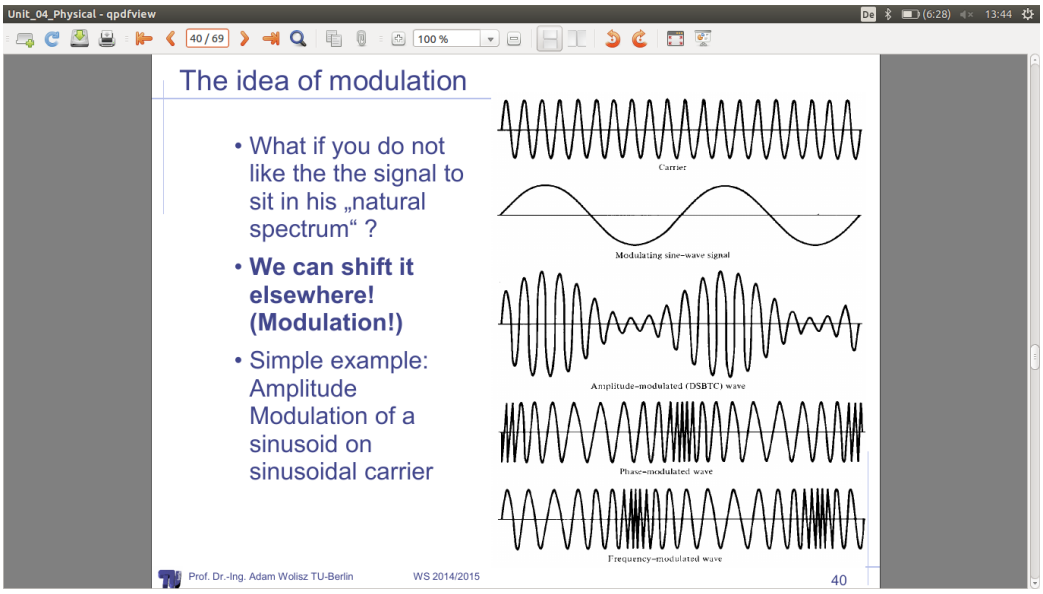
  - $\rightarrow$ achievable for error free transmission of porper code with no delay

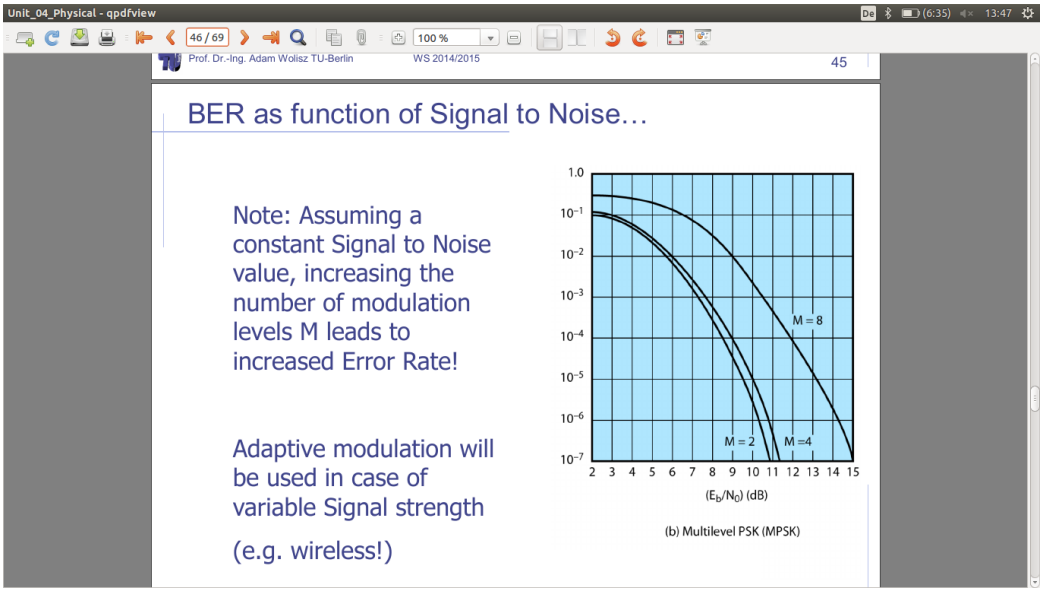
- way to analyse achievability
  - BER = Bit error Rate
  - $\frac{Eb}{N_0}$ Eb = ratio of Signal Energy per Bit; N_0 noise power density per Hz

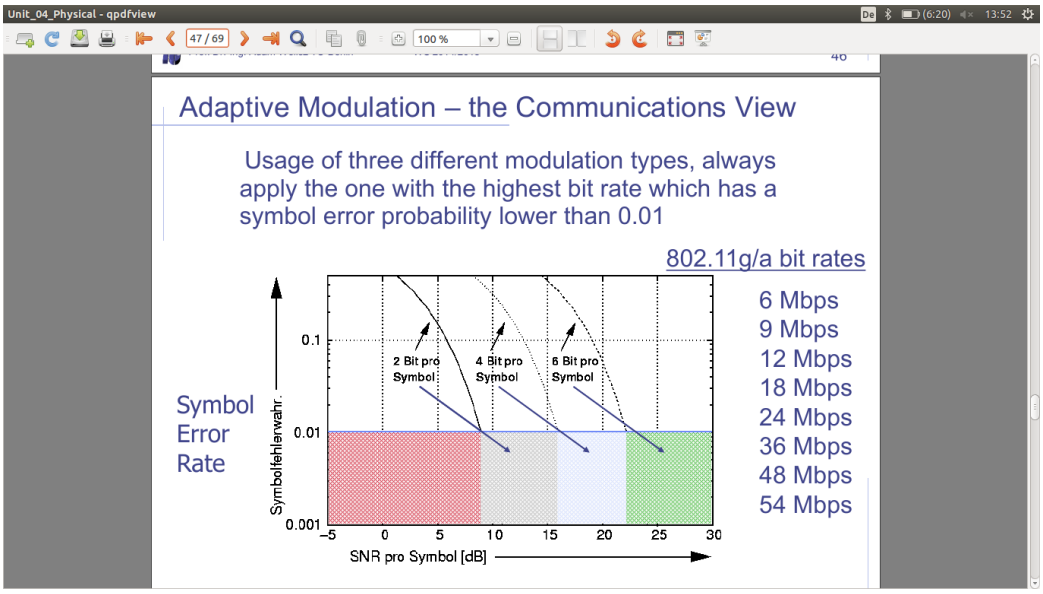


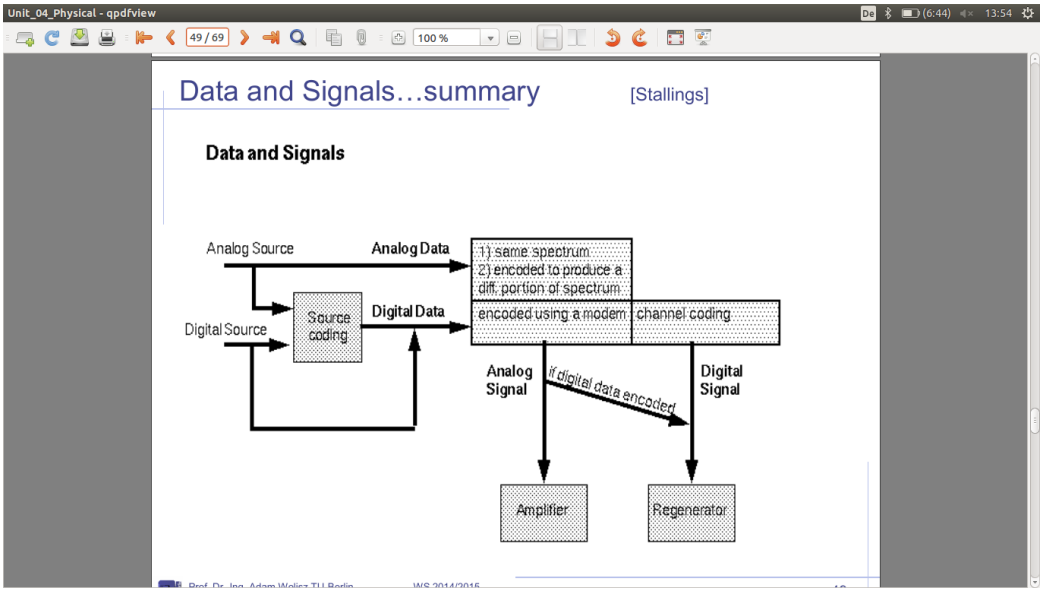
- passband communication with Modulation (shifting information to useful frequency band)
  - FM,AM, phase modulation
  - mixing AM and PM it leads to QAM (modem pictures) Quadrat Amplituden Multiplizierer
  - BER to SNR: konstant SNR $\frac{Eb}{N_0}$ increasing number M of achievable signal levels will lead to greater BER
  - adaptiv modulation

- signals transmission


Unit_04_Physical - qpdfview
40 / 69
100 %
The idea of modulation
• What if you do not like the the signal to sit in his „natural spectrum“ ?
• We can shift it elsewhere! (Modulation!)
• Simple example: Amplitude Modulation of a sinusoid on sinusoidal carrier
Carrier
Modulating sine-wave signal
Amplitude-modulated (DSBTC) wave
Phase-modulated wave
Frequency-modulated wave
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
40


Unit_04_Physical - qpdfview
46 / 69
100 %
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
45
BER as function of Signal to Noise…
Note: Assuming a constant Signal to Noise value, increasing the number of modulation levels M leads to increased Error Rate!
Adaptive modulation will be used in case of variable Signal strength
(e.g. wireless!)
1.0
10⁻¹
10⁻²
10⁻³
10⁻⁴
10⁻⁵
10⁻⁶
10⁻⁷
M = 8
M = 2
M =4
2 3 4 5 6 7 8 9 10 11 12 13 14 15
(Eb/N0) (dB)
(b) Multilevel PSK (MPSK)


Unit_04_Physical - qpdfview
47 / 69
100 %
Adaptive Modulation – the Communications View
Usage of three different modulation types, always apply the one with the highest bit rate which has a symbol error probability lower than 0.01
802.11g/a bit rates
6 Mbps
9 Mbps
12 Mbps
18 Mbps
24 Mbps
36 Mbps
48 Mbps
54 Mbps
Symbol Error Rate
Symbolfehlerwahr.
0.1
0.01
0.001
2 Bit pro Symbol
4 Bit pro Symbol
6 Bit pro Symbol
–5 0 5 10 15 20 25 30
SNR pro Symbol [dB]


Unit_04_Physical - qpdfview
49 / 69
100 %
Data and Signals…summary
[Stallings]
Data and Signals
Analog Source
Analog Data
1) same spectrum
2) encoded to produce a diff. portion of spectrum
Digital Source
Source coding
Digital Data
encoded using a modem
channel coding
Analog Signal
if digital data encoded
Digital Signal
Amplifier
Regenerator

1.1.1 Multiplexing

- Multiplexing is to avoid to short bit: gets difficult to reencode (ISI := previous bits have similar effect to noise/adaptive equalization fights against)
- mulitone Modulation (devide channel in subchannels with different coding in modulation sheme which change from time to time)
- Spread spectrum Technologies (avoid Frequency interference) makes System more complex but also safer in Transmission (Frequency hopping)
- FHSS - carrier Frequency changes periodicly according to pn sheme (Robustness because frequency selective fading and interference limited to short period of time)
- DSSS - every bit (duration T_b) is multiplied by a sequence of narrow pulses (chips) with time duration T_c
  - Spreading factor $N = \frac{T_b}{T_c}$
  - coded to appear as pseudonoise = PN
  - ADVANTAGE Users can share Bandwith by using spreading Codes which are orthogonal to each other CDMA
- Simplex (one direction - tele) , Half-Duplex (one at a time - police radio), Duplex (telephone)
- Transmission formulas:

$$\frac{\text{Bits}P}{\text{speed}R} = \text{TransmissionTime}$$

$$T = \text{PropagationTime} = \frac{L}{\text{speed}}$$

$$\text{InformationonMedia} = (\text{Transmissionspeed}) \cdot (\text{Propagationdelay})$$

- always remember trade-of between BER and Bit/rate
- Multiplexing
  - FDM: remember prisma and light: vary Carrier frequencies of modulated channels
  - OFDM: orthogonal... earn in spectrum efficiency

  - OFDM = inverse Multiplexing: transfer Stream higher bit rate to n Streams lower bitrate (each so narrow to be homogenous in its spectrum)
  - TDM = whole bandwith is used all the time, but by different users

- Time and Frequency Multiplexing
  - GSM cellular telephony FDM with TDD/FDD (Cellular Downlink/Uplink shared medium duplex transmission enable)
  - synchronous (specified time per client(if not used wasted)) and statisical (U-Bahn, use whenever/howmuch you want as available) TDM

- space division multiplexing - sectorised antenna (transmission in space 1 doesn't interfer with space 2)
- frequency planning: divide frequency spectrum in different cell (dynamically vs. statically)
- dimensions of muliplexing
  - time
  - frequency
  - code
  - sometimes space

- care for seperation... e.g. orthogonality
- synchronous allocation or statistical allocation

1.2 Switching

- Circuit Establishment follows Data Transfer follows Circuit Teardown
- Timing
  - Transmission Delay: Time to transmit data (packet)
  - Propagation Delay: Time packet is being transmitted

- Circuit Switching features

  - explicit setup and release requires proper signalling in which time no data can be sent: Connections may be refused if ressources are already blocked
  - after setup ressources are assured: bytes are only delayed by propagation; no packet losses because of constant sequencing

- Bursty Data: assembling the data into packets interspersing them onto one physical communication path
- Message Switching
  - permanent connection between each i/o of switch
  - user generated data has to carry information in its header uniquely defining which route to be chosen
  - multiplexing on different flows on a single path
  - routing information of the message has to be processed upon arrival of the data: (Store and forward(forwarding) vs. cut-though (difficult to implement))
  - buffering is needed: queuing delay + propagation delay
  - length of data should be quite constant
- Packet Switching: Avoids the problem of small data stuck behind big data $\rightarrow$ well defined blocks of bytes
- Datagram Packet Switching
  - header with routing information has to be processed in per-packet basis
  - statisical multiplexing on a single path recquires buffering
  - few packet might have priority
  - constant upper bound packet length eases control (e. g. Memory Management)
  - queuing delay and packet length determine the delay
- Store and Forward vs. Cut-through (forwarding is started as soon as the header is processed)
- bottleneck: sending packet with length n. Transmission time is inverse proportional to Transmission speed

- Routing vs. Forwarding
  - Routingtables: where will we send a packet with adress xxx done in advance
  - Forwarding: real processing; done upon arrival
- delay types
  - nodal processing: check bit errors, determine output
  - queing: time waiting for output allowence, depends on congestion
  - Transmission delay: R= link bandwith (bps), L= Packetlength (bits), time to send data into link: $\frac{L}{R}$
  - Propagation delay: d = length of the physical link; s = Propagationspeed in medium, Propagation delay: $\frac{d}{s}$
- Jitter: discrepancy between ideal and real, also variable Values/'Constants' might be significant; is often the reason for biterrors: bit is expected but hasn't arrived yet, so wrong value will be interpreted.
- Advantages of **Circuit Switching**
  - Guaranteed Bandwidth; no best effort
  - simple abstraction: reliable channel, no worries about packet-losses or out-of-order
  - simple forwarding
  - Low per-packet overhead
- Disadvantages of **Circuit Switching**
  - Wasted Bandwidth: unable to achieve advantages by statistical multiplexing
  - Blocked Connections, if ressource not available: no okay service to everybody
  - Connnection Set-up delay
  - Network state: network must store per connection information
- Datagram Packet Switching vs. Circuit Switching
  - explotation of statistical multiplexing at packet switching
  - connectionless: easy compensation if one link fails (pro ps)

  - deployability (pro ps)
  - congestion (con ps)

- Mix: Virtual Circuit: Packet Switching
  - connection has to be build and reserved and torn down
  - smaller routing table
  - quicker forward processing: convenient for high speed links
  - if switch crashed: reinitiate the connection
  - it can guarantee a 'quality of service'
- to survive failure make things as stateless as possible
- external and internal Operation - Packet switching: datagrams or virtual circuits
- connectionless:
  - build virtual circuit: status request
  - sequencially numbered
  - delivered packets in sequencial order
- connectionorientated
  - packets handled independently
  - external datagram service
- in packet switched networks connection oriented service can be offered on top of virtual circuit switching
- connection is established with datagram handshake
  - loss free operation
  - in sequence packet delivery will be provided by end to end mechanisms
  - e.g. TCP

1.3 Error Correction FEC

- Framing
  - delimiting PDU begin/end-marker
  - special control symbols
  - field length marker at the beginning of the field
  - frame markers can be bit or character oriented (J,K in IBM token ring)
- if marker of field length is corrupted a whole field would be lost
- delimiting characters in character based transmission: composed of ASCII-symbols (e.g. DLE STX ...data... DLE ETX)
  - a single bit error in 8bit character can cause misinterpretation
- bit oriented transmission-delimiting flags
  - transmitted information is represented as a string of bits (octets) e.g. 01 11 11 10
  - transmission transparacy is assured with bitstuffing
- Combination of them to increase framing power: bit-delimiting flags with character count
- Error Hypothesis
  - independent bit errors: each bit is an error with given probability p
  - for n bit packet: probability of error free transmisstion is $(1-p)^n$
  - probability for at least one bit error in the frame is $1-((1-p)^n)$
- error tends to appear in bursts; less frames are corrupted ar bursty error
- open loop error detection (without feedback to sender)
  - redundancy within in the frame
  - significant higher redundancy is needed to correct errors rather than detect
- **parity check**
  - column wise addition of parity bits (also two dimensional)

  - gerade anzahl 1 –>0 ungerade 1 –>1

- there are limited capability to each error detection algorithm
- **Block code:** m data to be sent, r bits of error controlbits are added, only odd numbers of bit errors can be detected
  - Hamming distance := number of bit (C) two following packet are a allowed to differ: you could try and guess what it was
  - code C can detect any combinations of x or fewer errors if

$$d_{min}(C) \geq x + 1$$

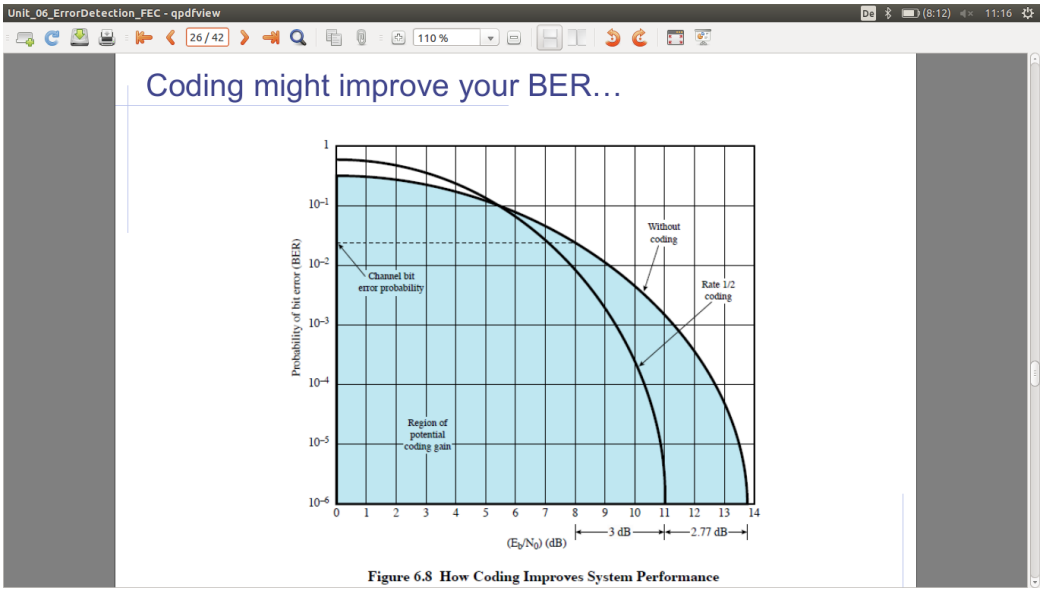
  - a code C can correct any combinations of y or fewer errors if

$$d_{min}(C) \geq 2y + 1$$

  - a code C can detect any combinations of x or fewer errors and correct any combination of y or fewer errors if

$$d_{min}(C) \geq x + y + 1$$

  - hamming distance of 3 can be achieved by n-m redundacy bits: e.g.: n=7, m=4, f1,f2,f3 = redundancy bits (siehe Ue-4 & Paper: Introduction in hamming coding)
  - operation of calcution is mod2 operation (XOR)
- effect of hamming coding: at good SNR significant lower BER: improved transmission
- **Polynomial coding: CRC**
  - error detection
  - overhead length **independent** of code word length, easy to computate, high probability of detection of multiple bit errors
  - Polynomial has to be agreed on sender and receiver side
  - e.g.: 11 00 01 represents $x^5 + x^4 + x^0$ k bits represent polynomial order k-1
  - k-1=r to make CRC work append r bits to original sequenz (siehe Ue3)
  - Prozess CRC



1. sender: extend bit by adding r zero
2. sender: scramble
3. receiver descramble
4. if added r bit = zero: positively no errors, negatively error invinsible for PN sheme

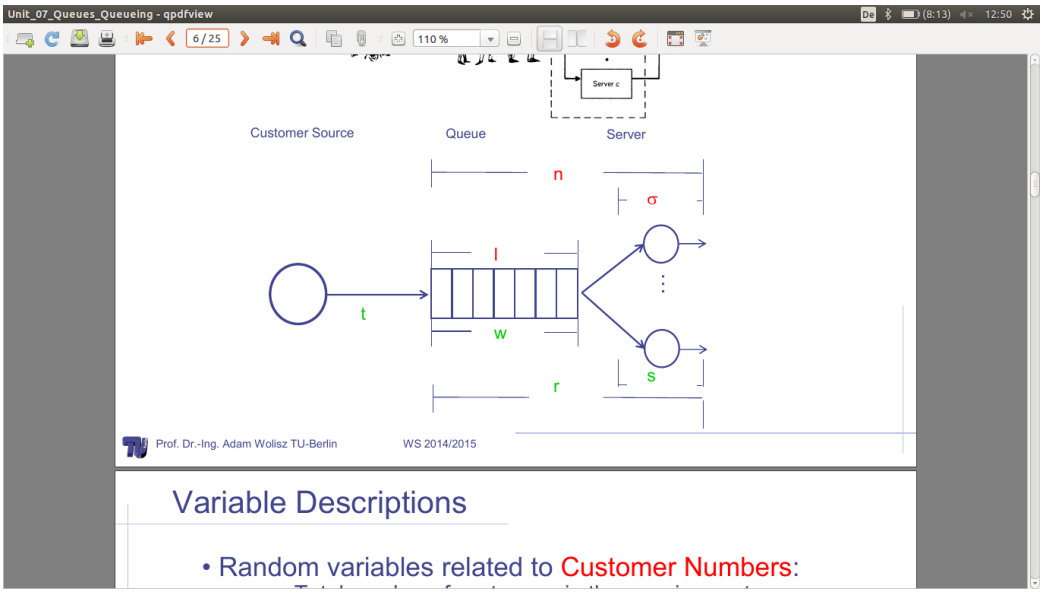
- will detect every single bit error if x+1 in sheme
- hardware implementation works fine (shemes as standartised) ip/tcp uses software implentation for the same (l/B-endian tolerant)
- additional reading 'Perfomance on Checksums and CRC's over real data'
- FEC: no additional delay in data transmission, significant increases of volume of data, nor necessarily simple processing

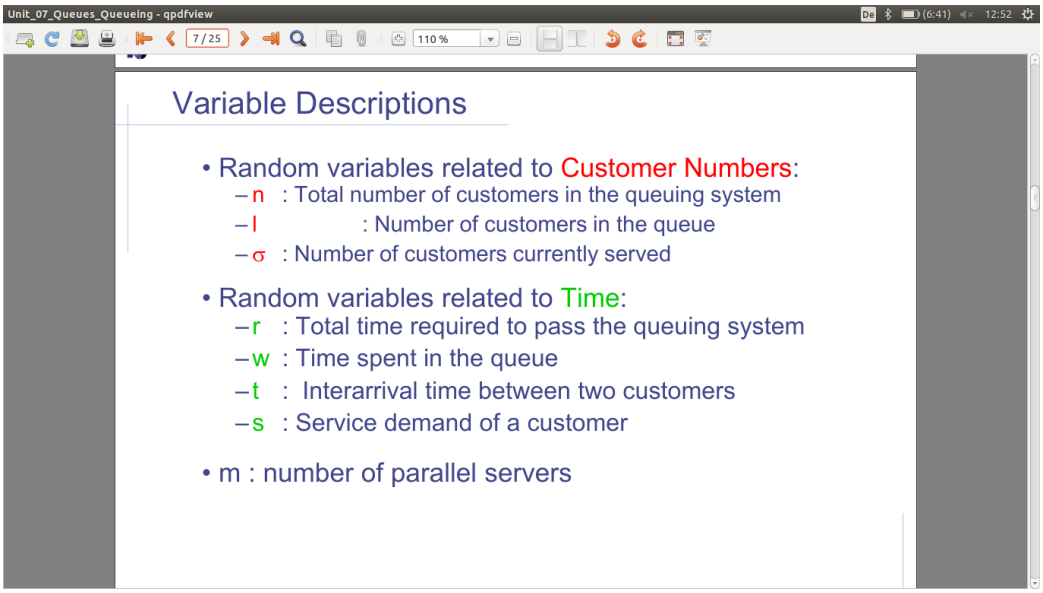
- closed loop error detection and retransmission at ARQ protocols

1.4 Queues and queueing

- queueing becomes necessary for service which are irregularyly accessed and take random time to proceed
- customers arrive individually in discrete, randomly distributed time intervals according to their inter arrival time distribution

- some service can be provided on a number of parallel servers
- each customer has a randomly distributed service time
- variable discriptions





- abstractions to understand queueing
  1. infinite amount of potential customers - the queue does not effect the arrival of new customers

2. infinite queue length - being long enough
3. selected distribution of random variable - trade-off between reality and computational designs

- classification system for queueing system by kendall: A/B/X/Y/Z (A: M:=exp)
  1. A: interarrival time pattern
  2. B: service time pattern
  3. X: number or parallel servers
  4. Y: the restriction of system capacity (total amount of slots in the system)
  5. Z: the queue discipline
- Formula:
  - arrival rate: average customers arriving in the que per time unit

$$\lambda = \frac{1}{E(t)}$$

  - Servce rate:

$$\mu = \frac{1}{E(s)}$$

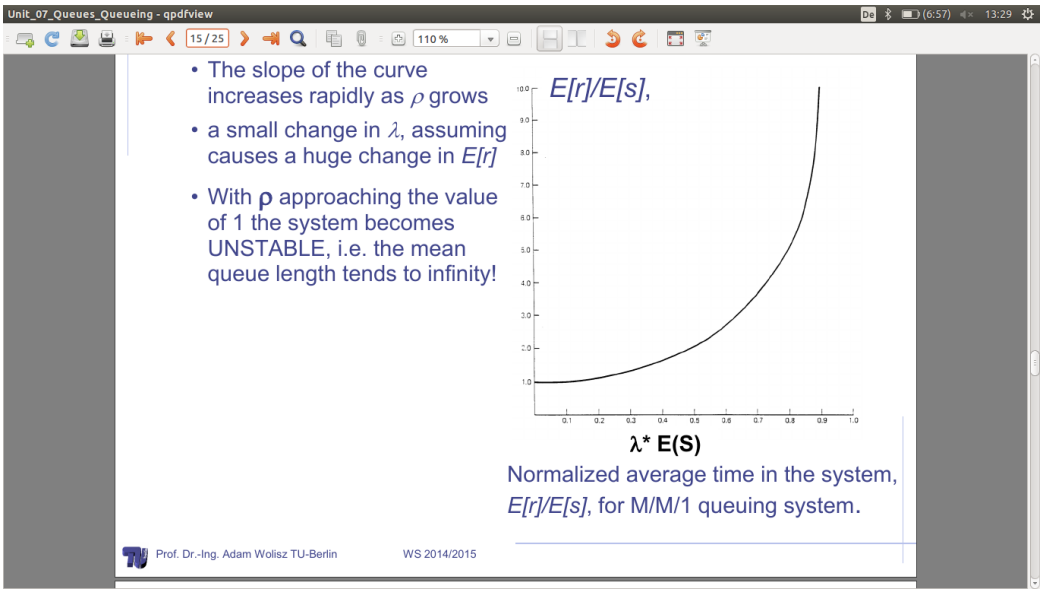
  - traffic intensity; m=# customer , $\rho < 1$ stabil

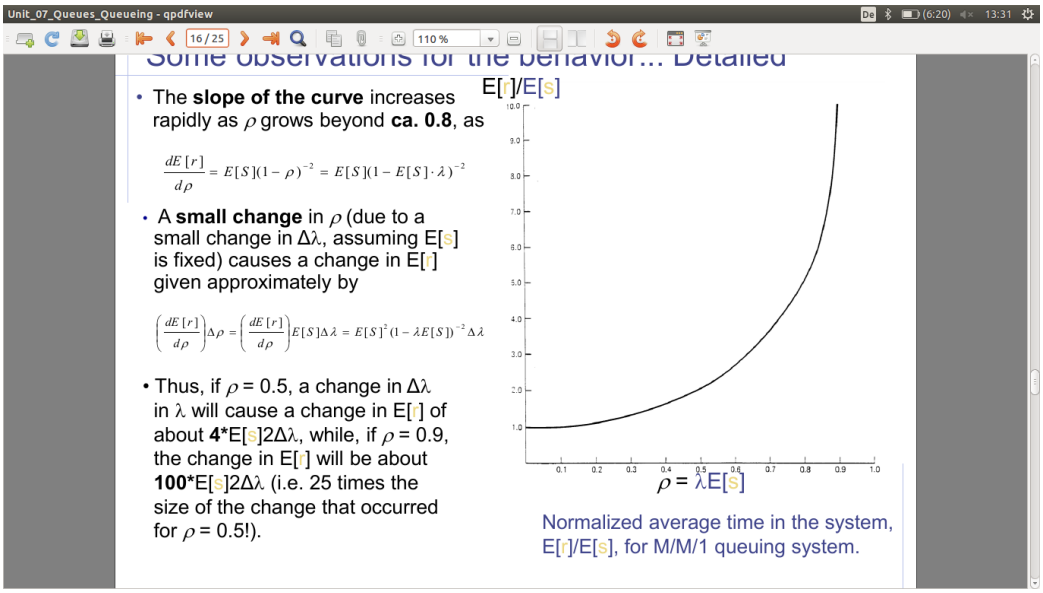
$$\rho = \frac{\lambda}{\mu \cdot m}$$

  - at stable systems is $\rho = U_k$ Server utilisation

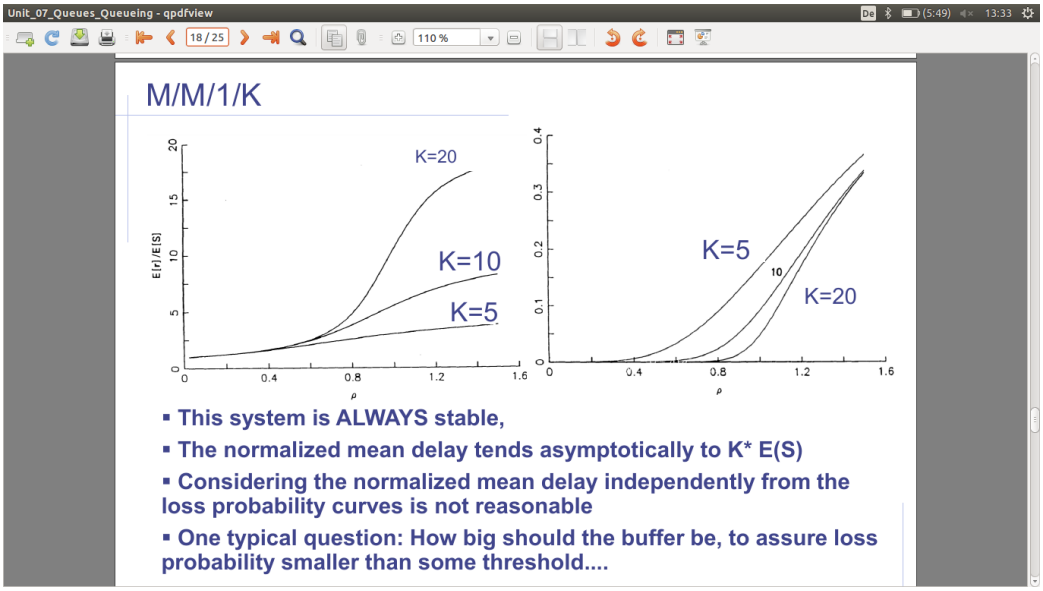
$$\rho = U_k = \frac{\lambda}{\mu \cdot m}$$

  - at long run throughput must equal to arrival rate
- usual simplification
  1. customers arrive individually in discrete, random, independent, identically distributed (i.i.d) time intervals
  2. parallel servers are identical and work independently
  3. packet transmission duration resulting out of packet length


Unit_07_Queues_Queueing - qpdfview
15 / 25
110 %
• The slope of the curve increases rapidly as ρ grows
• a small change in λ, assuming causes a huge change in E[r]
• With ρ approaching the value of 1 the system becomes UNSTABLE, i.e. the mean queue length tends to infinity!
E[r]/E[s],
λ* E(S)
Normalized average time in the system, E[r]/E[s], for M/M/1 queuing system.
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015


Unit_07_Queues_Queueing - qpdfview
16 / 25
110 %
E[r]/E[s]
• The slope of the curve increases rapidly as ρ grows beyond ca. 0.8, as
$\frac{dE[r]}{d\rho} = E[S](1-\rho)^{-2} = E[S](1-E[S]\cdot\lambda)^{-2}$
• A small change in ρ (due to a small change in Δλ, assuming E[s] is fixed) causes a change in E[r] given approximately by
$\left(\frac{dE[r]}{d\rho}\right)\Delta\rho = \left(\frac{dE[r]}{d\rho}\right)E[S]\Delta\lambda = E[S]^2(1-\lambda E[S])^{-2}\Delta\lambda$
• Thus, if ρ = 0.5, a change in Δλ in λ will cause a change in E[r] of about 4*E[s]2Δλ, while, if ρ = 0.9, the change in E[r] will be about 100*E[s]2Δλ (i.e. 25 times the size of the change that occurred for ρ = 0.5!).
ρ = λE[s]
Normalized average time in the system, E[r]/E[s], for M/M/1 queuing system.

- plot of the relations
- more realistic approach with finite queueing length (effects probable loss of customers)



- it is better to use a single queue with multiple servers, so servers are always in use as there are customers.

1.5 Examples of Transmission Systems

TOBE checked again

- telephpone Backbone: FDM Carrier Standard: old solution FDM transmission in trunk transmission
- FDM transmission standard in hierachy: American Digital Hierachy
- American TDM Carrier Standard: e.g.: D4 frame (digitized voice)

$$8000 \cdot (24(\text{Channels}) \cdot 8(\text{Bi})\text{t} + 1)\,[\text{bps}] = 1.544 Mbps$$

- Control channels are used for signalling
- pathlength changes, so number of store bits in the transmission
- pulse stuffing concept: mulitplexer showing equal data rates for each input channel
  - not needed since both data streams have exactly the same rate
  - additional information for each channel must be allowed to adjust to compensate (stuff bits)
  - output datarate should be equal to double (legally upper bound) input rate
- a real multiplexer 1.1
- Problems of PDH
  1. each part of the world has its own transmission standards
  2. bit stuffing spreads data over the frame
  3. too hard to interoperate
- **SDH**
  - avoid problems of PDH
  - achieve higher bitrates
  - assumes high quality clock synchronization
- SONET/SDH - basic components
  1. Path

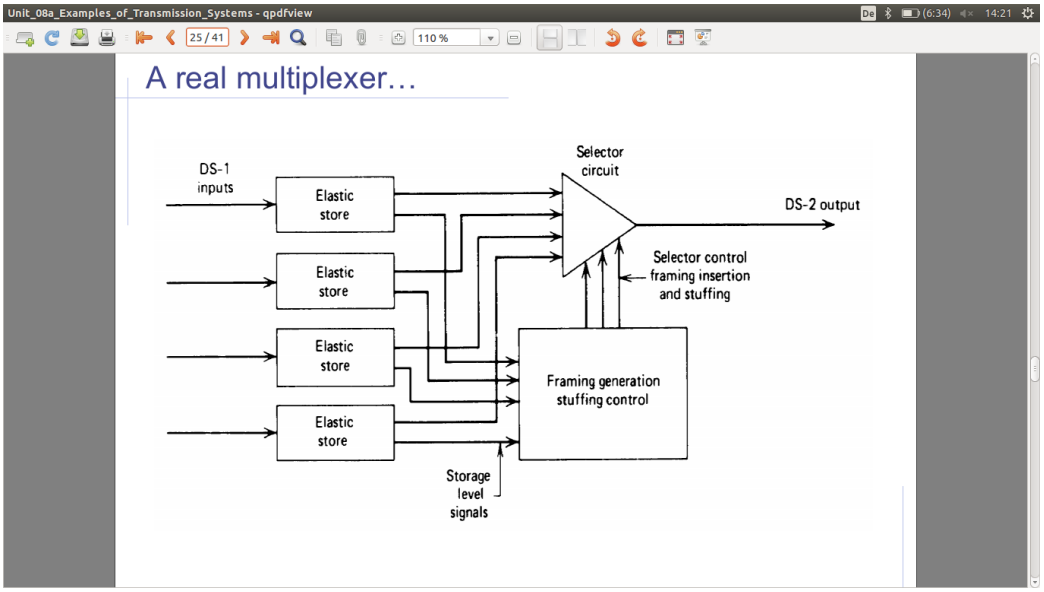


Bild 1.1: a real multiplexer

2. Line
3. Section
4. a repeater binds sections to line
5. a multiplexer bind lines to path

- SDH clocking: although the network elements are totally synchronous there are delays in the network: lower level synchronises to higher levels
- each frame is always (SONET) 125μs long STS
- frame has fixed size: 9 columns x 87 columns, Consits of Header and Payload
- payload start right at the end of header, it is fixed by pointer in the header: Delay correction of arriving Packets to be multiplexed
- Why SONET/SDH does better in general
  - interconnection is easy
  - Justification if needed, is performed by pointer (no stuffing management)
  - Ease extraction of tributaries: any STS-1 line can be found and extracted form the frame

1.6 POTS

- components
  1. subscriber
  2. local loop (TP, analog transmission duplex)
  3. exchange (switching centers, end office support subscriber)
  4. Trunks (digital, TDM, Unidirectional)
- Hybrids match impedance from (local) two wire loop to four wire Toll circuit: avoids echo
- channel bandwith devided into several channels: voice(1), uplink(few), downlink(lots)
- Signaling is needed in networks to control their operation and indicate status
- Subscriber Loop signalling (dialing, make the phone ring), Interoffice signalling (set-up call, indicating that a call is established, billing purpose, diagnosis)
- Tone dialing vs. Frequency dialing
- Tone dialing signals in the voice channel
- subscriber loop signalling (trunk busy, dial tone)
- blocking vs. non blocking (blocking can result from limited line capacity)

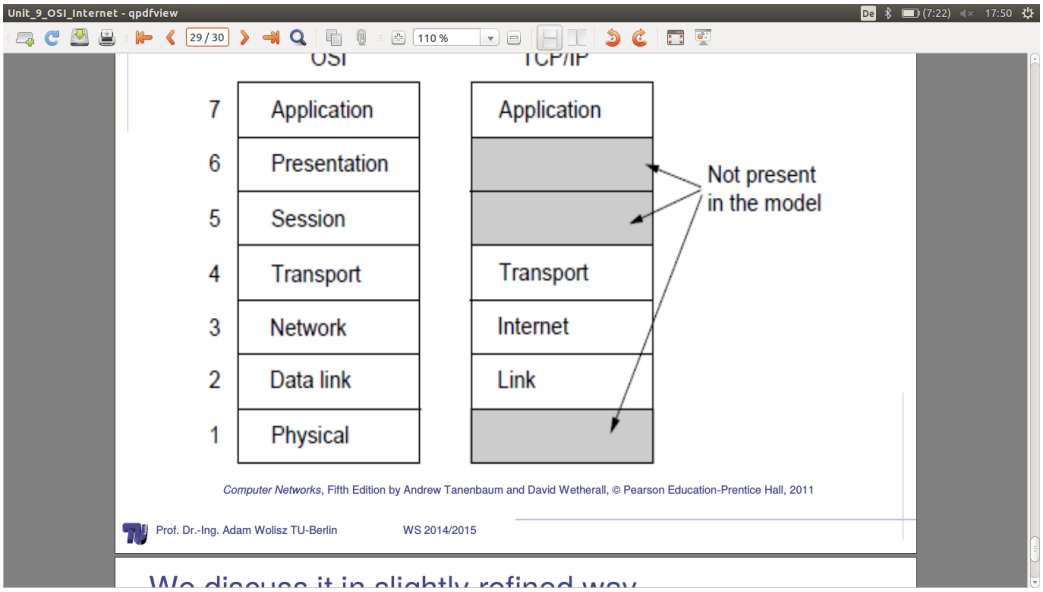
1.7 OSI Internet

- OSI unified view
  - communication between peer processes is virtually and indirect
  - layer n+1 transfers information provided by layer n
  - service are available at SAP
  - each layer passes DATA and control information to the layer below until PHY and transfer occurs
  - The data passed to the layer below is called a SDU

  - SDU's are encapsulated in PDU
  - layer n in machineA interacts with layer n in machineB: these entities are called peer process
  - machine uses a set of rule and conventions called layer n Protocol (e.g.: TCP)
  - layer n processes communicate by exhanging PDU's
- Layering
  - simplifies design, implementation and testing by partioning
  - each layer can use seperate protocol
  - protocols make calls from layers below
  - Layering provides Flexibility for modifying and evolving protcol and services without change of lower levels
- Multiplexing tag or ID required in each PDU to determine which user an SDU belongs to
- every protocol adds header (in link layer: + trailer: CRC checksum)
- SDU is segmentated in PDUs and then reassemblied in SDUs at the receiver
- Physical layer
  1. transfers bit across link
  2. Definition and Specification of the physical aspects of a communication link
- Data Link Layer
  1. Transfer frames across direct connections
  2. goups bits into frames
  3. detection of bit errors, retransmission of frames
  4. activation, maintenance, deactivation of data link connections
  5. MAC for LAN
  6. Flow control (vs. Congestion Control, siehe UE4)
- Network layer (IP)
  1. creates a logical path between open system and individual subnetworks

2. supports routing
3. network address - end system address

- Transport Layer (TCP)
  1. transfers data end-to-end (connection orientated)
  2. reliable stream or simple block-by-block transfer
  3. Port numbers enable multiplexing
  4. message segmentation and reassmbly
  5. connection setup, maintenance and release
- Application layer provides service which are frequently required for e.g. file transfer, web access
- Presentation, Session layer are incorporated into the Application layer
- Internetworking to provide distributed applications
  - required for internet-communication based Applications
  - email, www. ... , peer to peer
  - independence of network topology
- IP is the bottleneck: freedom in layer n+x and n-x but IP highly standartised
- in the IP/TCP model three layers are integrated in other and therefore not present in the analysis



1.8 physical interfaces

- DTE - DCE - DCE - DTE (DCE-DCE:=Point-to-Point)
- Aspects of **Interfacing**
  1. mechanical: Type of connector/pins, male/female
  2. Electrical V.28 RS 232 unbalanced on-board interface
  3. functional: data(synchronous, asynchronous, simplex, duplex etc.), timing, control, ground
  4. procedural
- Loopback interface: Communication between DTE and DCE: Looped RXD to TXD pins
- USART: Status register for quick traffic control for on-board circuits

1.9 ARQ-Approach

- The concept of ARQ is retransmission of errornous frames by using a feedback orientated approach
- various kinds of errors to be dealt with: duplicates, disorders, lost, corruption of packets

- only possible for connection oriented transmission (except: acknowledged datagrams)
- sender and reciever create their local context expressing their local view
- neither receiver nor sender has complete information on the state of transmission: control information exchange is needed
- Sequence numbers are used to detect lost, disordered or duplicated packets and to relate ACK to data packets
- **timers** are needed to avoid **deadlocks**
- **acknowledgments** and timers are used to **provide the sender** with information on state of the reciever
- **Sequencenumbers** of PDU's are used to **provide the receiver** with information on state of the transmission
- Send-and-Wait (SnW)
  - inifinite loop possible, if connection fails (counter to limit number of retransmissions)
  - Packets are duplicated if ack is lost: an additional one bit identifier is used: called ABP
  - performance problem: one packet per round trip delay: ideally if packet length is ∞ (vgl. UE4)
- Alternating Bit Protocol (wichtig in VL)
  - Specifications Sender $(s, e, a)^4$; states, events, actions: each 2 bits
  - Specifications Receiver $(s, e, a)^2$ sequence number 0 oder 1
  - check unit 11 page 8ff (Final State Machines)
- requirements for correctness
  - sequence of states lead to deterministic execution state
  - execution state is terminating if no events can occur in the last state
  - the execution sequence is acyclic if no state occurs more than once
  - the execution sequence is cyclic if all states but the last one is distinct and the last state is equal to the first one. The execution may include non progress cycles

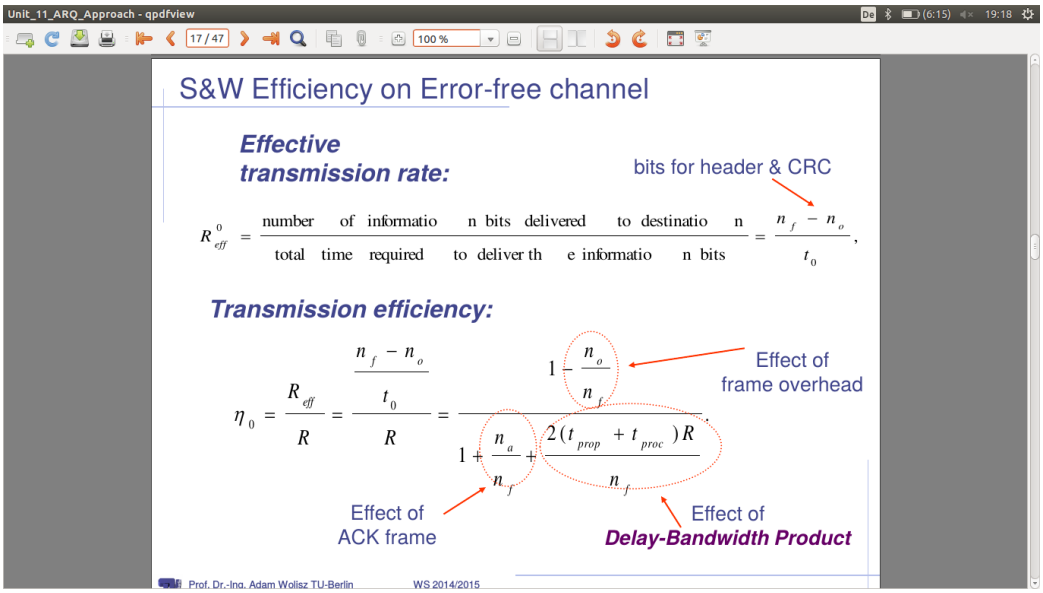
  - in a finite system all executions sequences terminate after a finite number of state transions or the cycle back to some previous state
  - termination in proper end states or deadlock
  - Unspecified receive - there appears an unexpected message
  - a state will occur after the another or immediatly after (temporal claim)

- full state space search: s byte of memory for one state. M = memory available, needed memory for S states $= \frac{M}{S}$
- Performance Analysis Snw (n bits, R transmission rate)

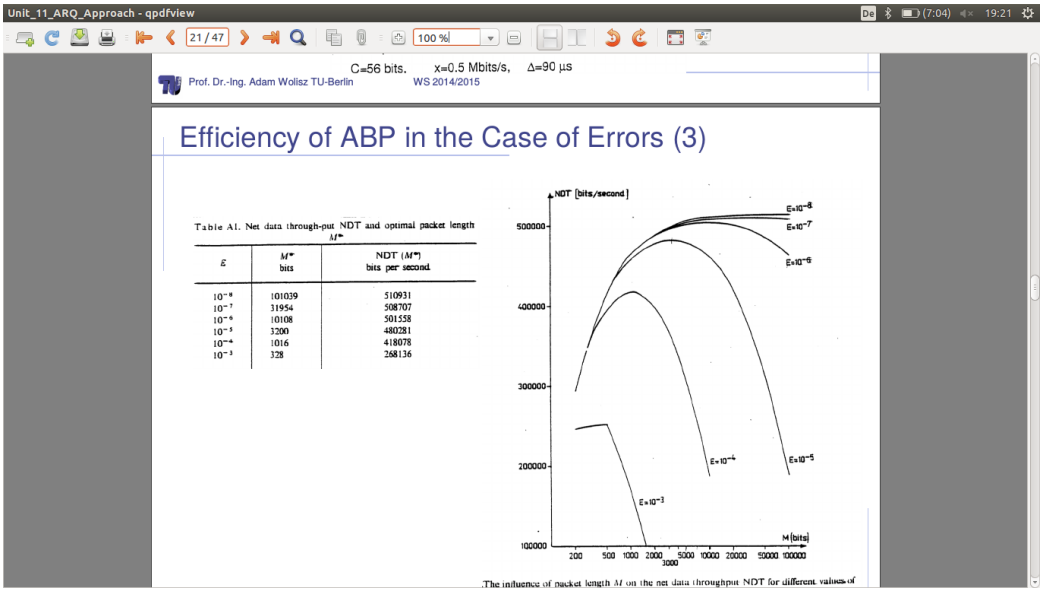
$$t_0 = 2t_{prop} + 2t_{proc(cessing)} + t_{f(rame)} + t_{ack}$$

$$t_f = \frac{n}{R}$$

$$t_{ack} = \frac{n}{R}$$

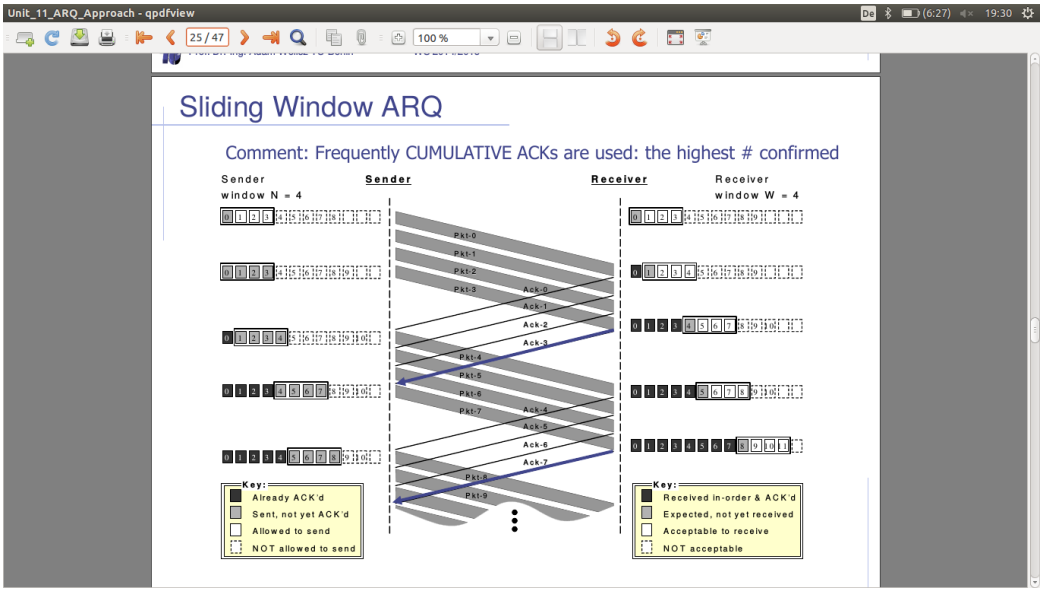


- ABP efficency in case of Errors: optimum packet length can be calculated
- Continuous ARQ
  - ARQ protocols which allow continous transmission of PDUs (GbN, SR)



  - sending packets without waiting for ack of every packet (window sequencing)
  - finite number of unacknowledged packets

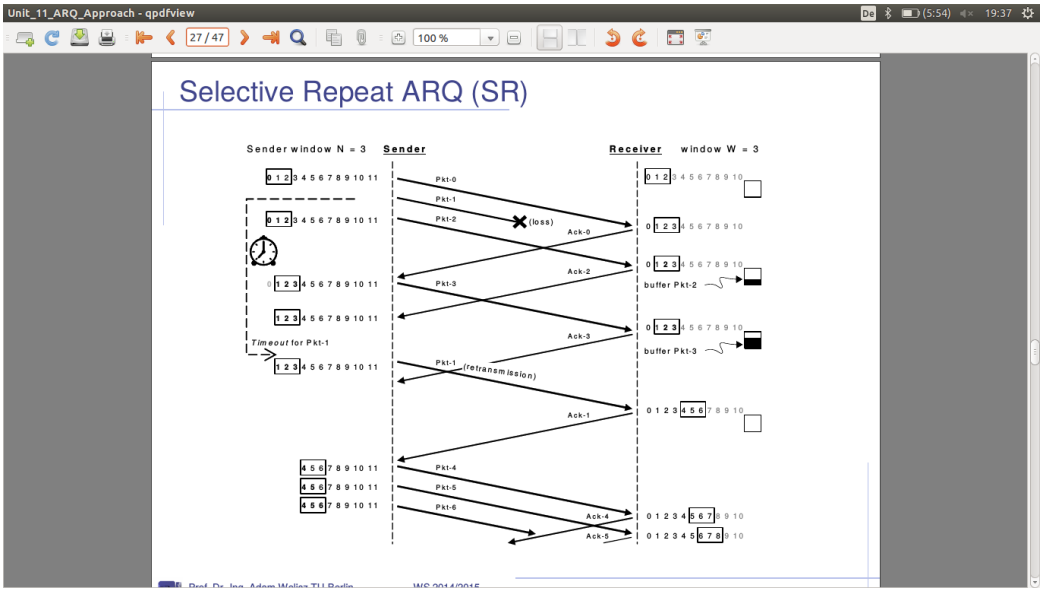
- window sequencing - sliding Window in ARQ (vgl. screenshot)



- Selective Repeat principle
  - ARQ protocol which only retransmits errornous packets

  - **pure positve feedback: sending side detect error throug time-outs**
  - each ack acknowledges only one packet
  - control packets are used as ack
  - indivdual timer per PDU is used
  - e.g.: w=4 window size

- ARQ(SR) schematic: look carefully!



- limits of sequence numbering: there are cases in which it appear for the sender to be the start of a new sequence although it was a repetition for packet, which ack had been lost and was because of that retransmitted. so the arriving packet is no duplicated at the receiver side and can not be identified as such.

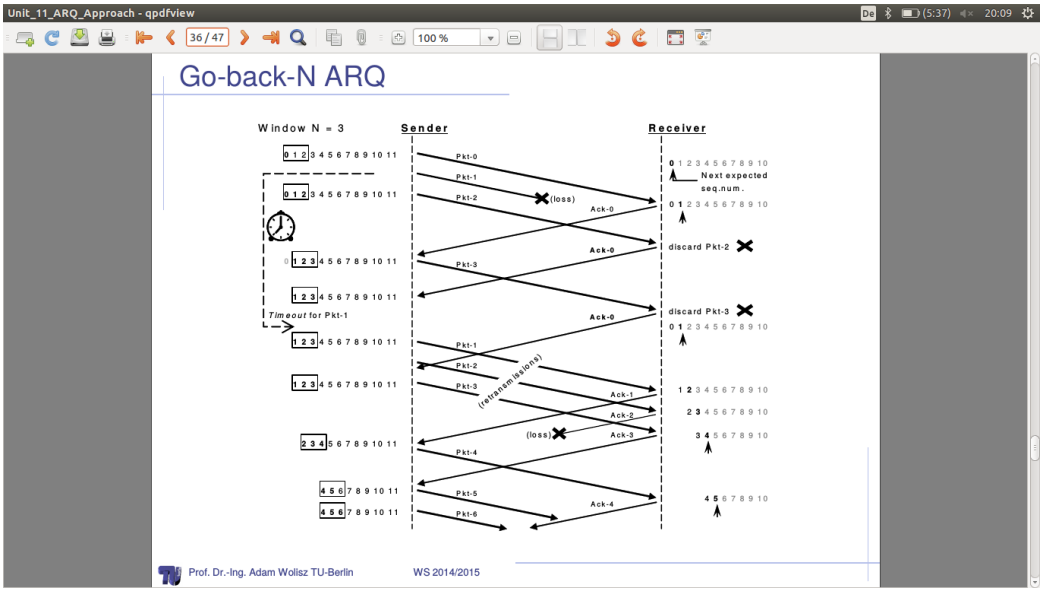
- seqeunce number range R must be more or equal to window size w

$$R \geq 2 \cdot w$$

- Selective Repeat efficency
  - depending on the window size
  - vulnerable: target to keep flow of packets continous <-> contradicts with Flow Control / Congestion Control

  - in error free case: SR is the most efficient protocol

- Selective Repeat summary
  - packets can be sent continously, as long as not more than w packets are unacknowledged
  - packet are numbered consecutively in a cyclic way
  - sender retransmit copy of packet which hasn't been acknowledged
  - ack from receiver has the same sequence number as set from the receiver
  - receiver stores packets in received order and deliveres them in order to the user
  - SRP is more efficient to ABP, because the sender must not wait on every sent packet to be acknowledged
  - required buffering at the receiver
- **Go-Back-N**
  - GBN works like SRP in absence of errors
  - if sender detects an error through time out, it retransmits the packet and all subsequent packets
  - GBN allows sender to have multiple unacknowledged packet without the necessity of the receiver storing out-of-sequence packets
  - configuration properties
    1. pure positve ack
    2. an ack acknowledges the corresponding data packet plus all previous ones
    3. control packets are used
    4. individual timer per PDU
- GBN ARQ schematic
- GBN becomes inefficient if Error Rate increases significantly
- piggybacked ACK vs. control packets (out of order question: be aware of the difference)
- ARQ - possible variants
  - Transmission strategies: continous vs. seperately



- ack strategies:
  1. pure positve feedback is sufficient
  2. pure negative feedback is insufficient (set-up the connection: packet failed or dead trunk)
  3. mix is used for opitmization
- singel ack per packet vs. cumulative ack
- control packet vs. piggybacked acknowlegment
- retransmission strategies
  1. only erronous packets
  2. erronous packets and all subsequent (avoid buffer at receiver for out of sequence packets)
- timer:
  1. timer per PDU vs. timer per connection
  2. individual global timer at Sender and Receiver each

1.10 Introduction to SDL

- stands for Specification and Description Language
  - SDL provides a graphical Representation (SDL/GR) as well as a textual Phrase Representation (SDL/PR)

  - SDL contains concepts such as: inheritance, abstract generic, types for block processes, service parametrization for Block instances

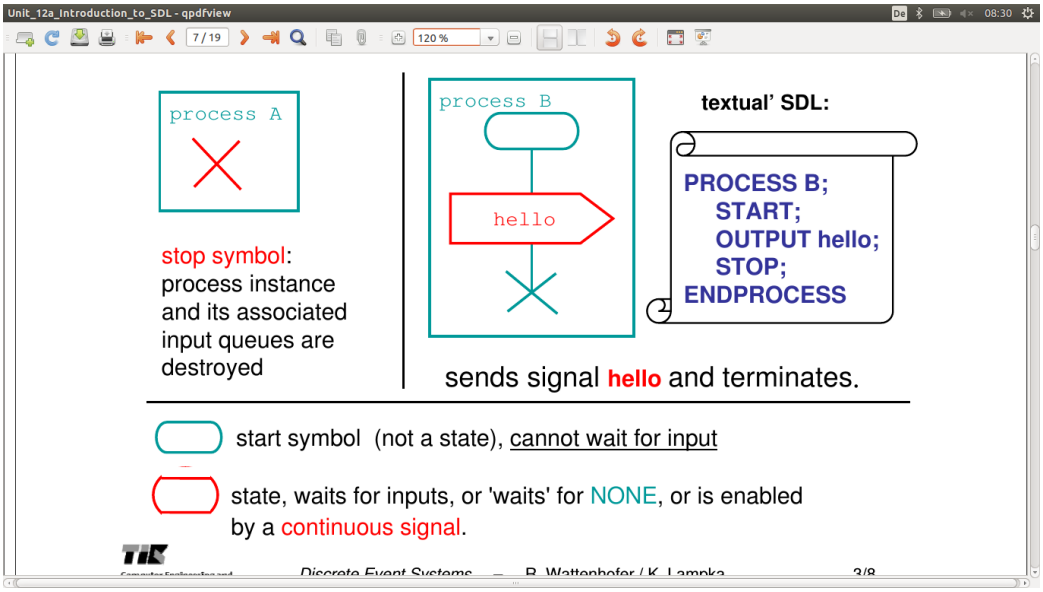
- Structure of SDL specification
  - hierarchical levels
    1. System (the root is always considered System)
    2. Blocks (in the middle)
    3. Processes (always the leaves)
    4. Procedure
  - Processes are entities which define actual behavior
  - Procedures can be viewed as some mechanism for encapsulation (sub-)behavior to be used multiple times (function/method)
  - some tools allow calling externel components in C/JAVAs

- signaling in SDL (idle=inaktiv, (faul))
  - **Processes** interact via signals (=messages) for output/input
  - outputs are send non blocking
  - each signal is **buffered** at the receiving process side (FIFO - queues)
  - signal transmission is buffered: therefore not immediate
  - routing is defined by channels (system level), signal routes (block level)

- SDL - Processes
  - processes are kind of automata (extended final state machines)
  - each process has its own input queue (infinite storage available... theoretically)
  - each state of a process is determined by
    1. current state of process
    2. value of current variable
    3. contents of input queue
  - state changes can be triggered by input messages: this may lead to
    1. send output
    2. call procedure
    3. execute loop

4. manipulate local variables
5. create new instance of process

– Hello World in SDL



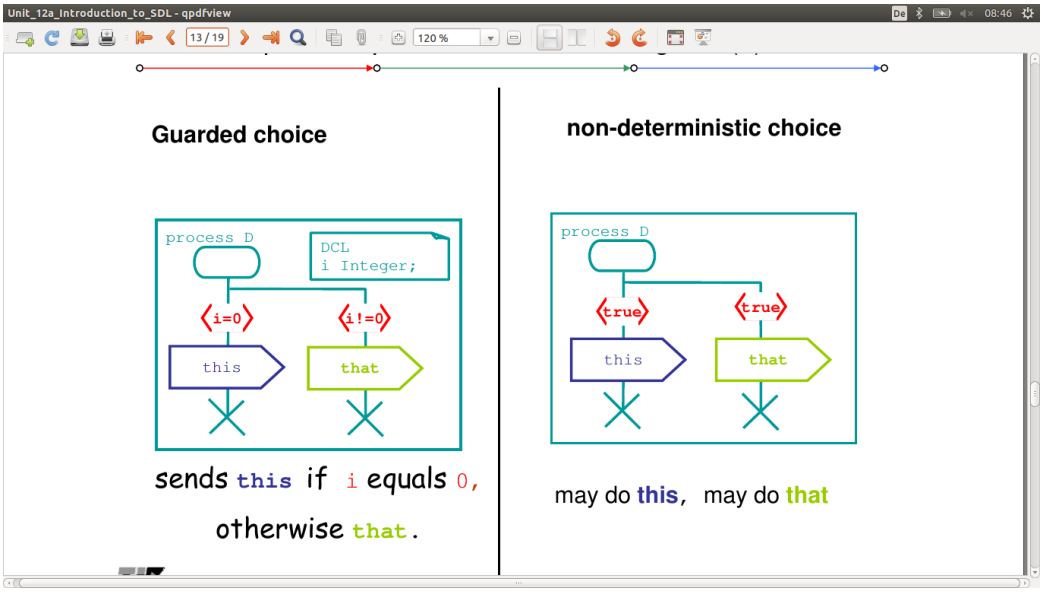
– signals which aren't expected are ignored and dumped unless not explicitly saved
– inputs are non-persistent: top elements in the queue are discarded (verworfen) if not expected: use save
– Asteriks in Inputs behave like wild cards: consume any signal form queue

- Process interaction: Output

1. output signal triggers sending of a signal to a connected process
2. format: name-of-signal (list of parameters(optional))
3. with VIA or TO route or receiver can be addressed specifically; otherwise non-deterministic choice of receiver possible

- Signaling on routes and channels

– signal routes/channels are non delaying
– implemented by synchronisation: but still delay from input buffer at receiver side

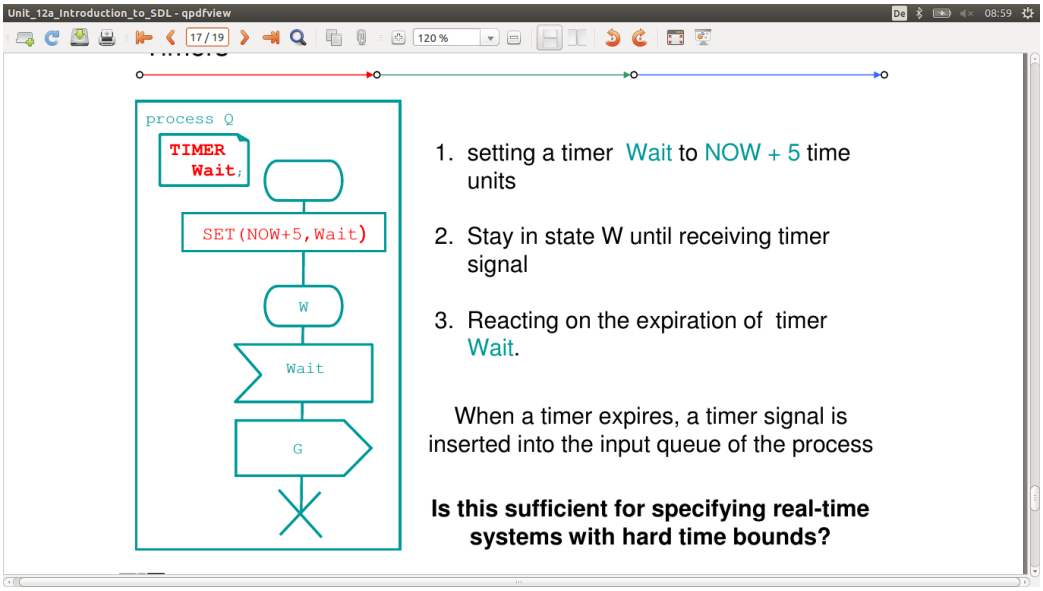


  - delaying can be implemented by another unbounded FIFO buffer in the middle; which makes delay unpredictable
  - channels and signal routes may be unidirectional of bidirectional

- continous signals
  - guards the branches of inputless alternatives
  - guards a boolean conditions on process data
  - if available input signals are always processed first
  - non determinism among various enabled continous signals can be resolved by assigning priorities to the conditions
  - sheduling
    1. check for input signal
    2. check continous signal with Prio 1

- Enabling condition
  - idea: block consumption of a signal if not ready
  - guards are an input command
  - signal can only be consumed if guard evaluates to TRUE
  - guard is a boolean expression
  - guard cannotcontain parameter of signal to be consumed

- Time in SDL
  - problem: time consuption of a transmission can no be modeled explicitly
  - SDL: allows to
    1. set timer
    2. react to timeouts
    3. refer to the current time
  - set time in SDL: timer expiration send signal to a local buffer queue; timeouts may not be immediate...



  - FSM, FIFO buffering, offers specification, (test/simulation (MSC-log)), validation; automatic code generation (to then implement in a real runtime environment for performance analysis)

1.11 Flow Control Link Protocols

- motivation: mismatch between the sender processing speed and receiver processing speed $\lambda \geq \mu$
- Flow control is specificly used to insure, that a source does not overwhelm a destination with more traffic than it can handle (sender side to protect receiver)

- each layer has to take the received information and prepare a new one
- sliding window flow control
  - after reception the receiving side provides a permit to indicate willingness on next frame
  - **ack** are send **as soon as possible**
  - **permits** are triggered by **buffer release**
  - ack can be permit but usually this is seperated
- Seperation of flow control and error control e.g. HDLC (link layer)
  - RNR and RR
  - window will be reopened by explicit permit or timer expiration
- Link utilization: $a = \frac{\text{PropagationTime}}{\text{TransmissionTime}}$

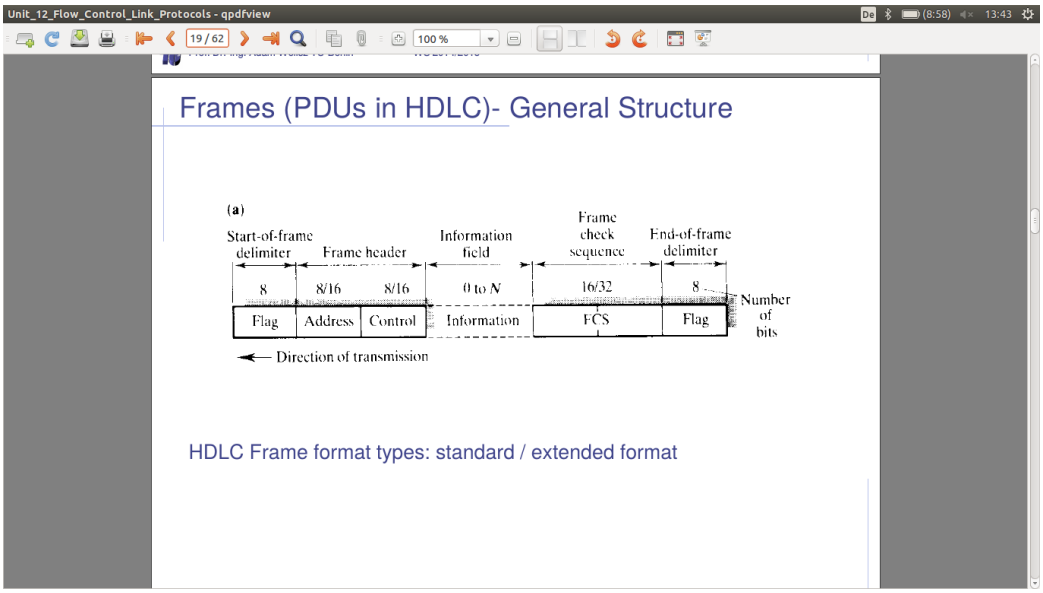
$$U = \frac{1}{1+2a}$$

- window size should be large enough to keep the transmission Pipeline filled
- buffer size could be set to maximum buffer length: $\rightarrow$ bad for long round trip delays
- buffers must have double size of round trip delay to avoid buffer overflow and keep pipeline filled
- the more buffer there is available the more decoupled is the operation between flow control and user (higher layers)
- Rate control (traffic shaping or network acces control)
  - control the amount of data per time unit
  - Token Bucket: Queue of packets without permit wait for permit from permit queue, which reflectes the limited space. transformation to sent packet with permit
  - pros: open loop approach (no influence on round trip delays)
  - pros: fewer packets in the network
- Link Protocol HDLC
  - Connection oriented reliable Transmission

1. Connection establishment/release
2. Error Control (ARQ, complex)
3. Flow control (window-based)

- connectionless mode also supported
- multiplexing/demultiplexing (several logical connections top of a single physical channel)
- splitting/recombinig (single SDUs <-> severa PDUs)
- severeal different operation modes

- frames PDUs in HDLC - general stucture



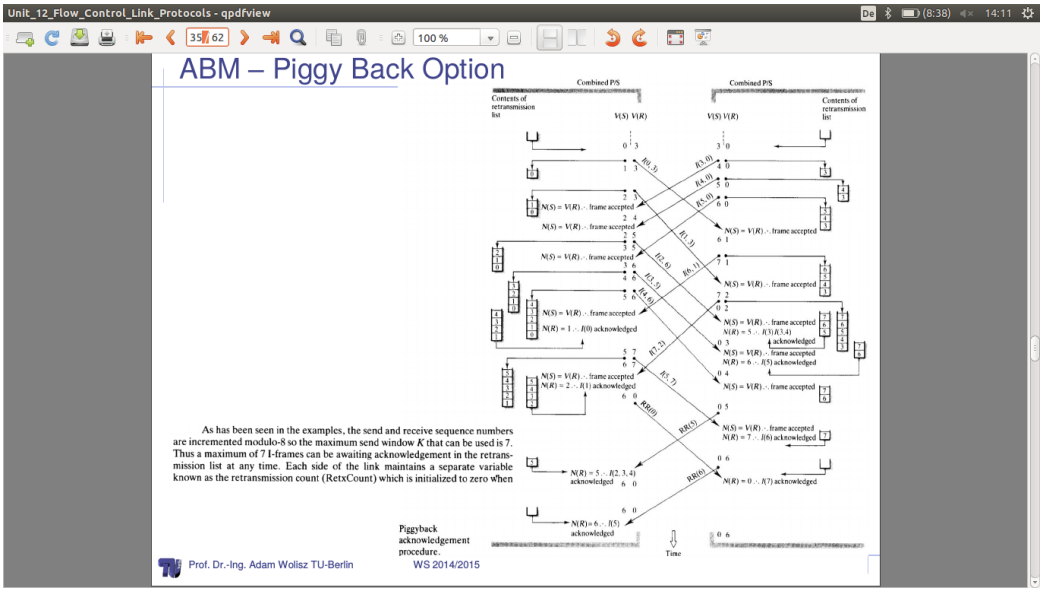
- Flag fields and bit stuffing
  - delimiting frame at both end with marker 0111 1110
  - receiver hunts for marker to synchronize
  - bit stuffing is used to avoid data containing the exact flag
  - bit stuffing: insert 0 after 5 1' and delete it at receiver without information analysis
- frames - addresses: single address field identifies next SAP
- Control field in HDLC
  - use Poll/Final (P/F) Bit which depends on context

  - in command Frame P bit set to 1 to solicit (poll) respnse from Peer
  - in response frame F bit is set 1 to indicate response to soliciting command
  - Usage
    1. sequence numbers of information frames: 8 or 128 (extended version)
    2. cumulative ack plus next expected number is transmitted
    3. positive ack can be sent in seperate supervisory frame or in an information frame (piggy backed)
    4. negative acks can be used for the sake of performance increase

- ABM Asynchronous Balanced Mode
  - balanced Mode: point to point
  - both can initiate connection and release
  - each station can both issue command and responds on an open connection: full duplex operation is supported

- ABM - Error detection using FRMR
  - error in decoding of the frame type (also CRC is correct)
  - excessive length of the information field
  - in error case: primary (initiating) station sets back the dialog to some initial point and uses the reset frame for requesting the secondary to do the same
  - ABM piggy backed schematic:

- ATM - Asynchronous Transfer Mode
  - reminder: virtual cicuits 1.2
  - ATM follows the virtual cicuit packet switched networks
  - important concepts:
    1. virtual cicuits
    2. fixed sized packets
    3. small cell size
    4. statistical multiplexing
    5. integrated services



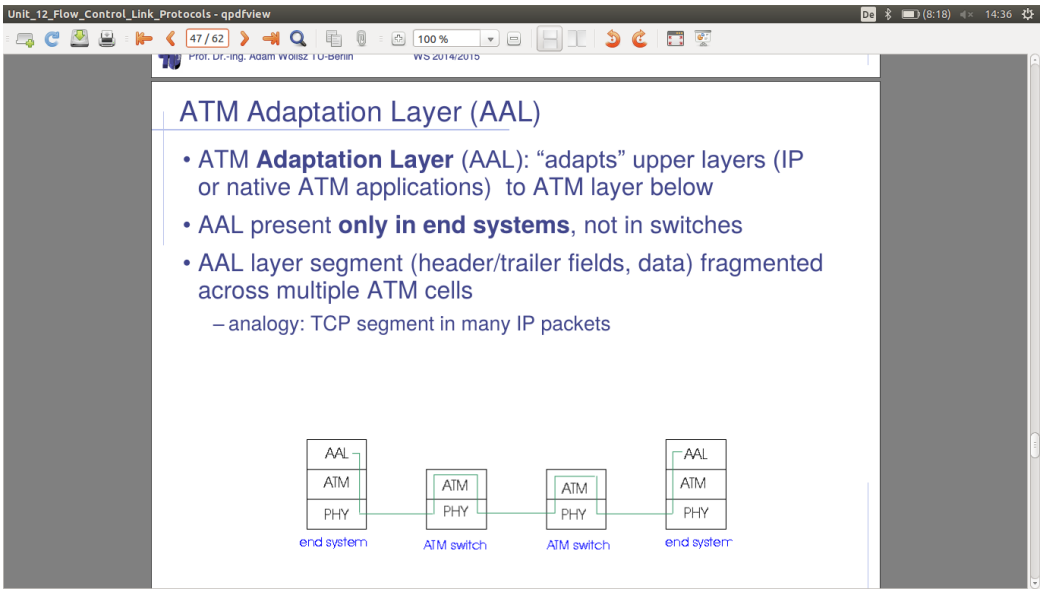
  - small fixed size packets simplifies the processing inside a switch and thus enables high data rates
  - Protocol reference model as three seperate planes (user plane, control plane, management plane)
  - second sublayer in which as a VPC several VCCs are combined

- Advantages of virtual paths
  1. simplified network architecture
  2. increased network performance and reliability
  3. reduced processing and short connection set-up time
  4. enhanced network services

- ATM concepts: fixed sized packets
  - pro
    1. simpler buffer hardware
    2. simpler line sheduling
    3. easier build large parallel packet switches
  - cons
    1. overhead for sending small amounts of data
    2. segmentation and reassembly cost

3. last unfilled cell after segmentation wastes bandwith

- AAL is only needed in end systems not in swiches



- PPP, ADSL - ADSL is PPP over ATM
  - dial up - ppp over async. serial or modem
  - allows framing, efficiency, authentification and address negotiation
- Point to Point data link Control: one sender - one receiver - one link: easier than broadcast link: no MAC, no explicit MAC addressing, e.g. dial-up link, ISDN

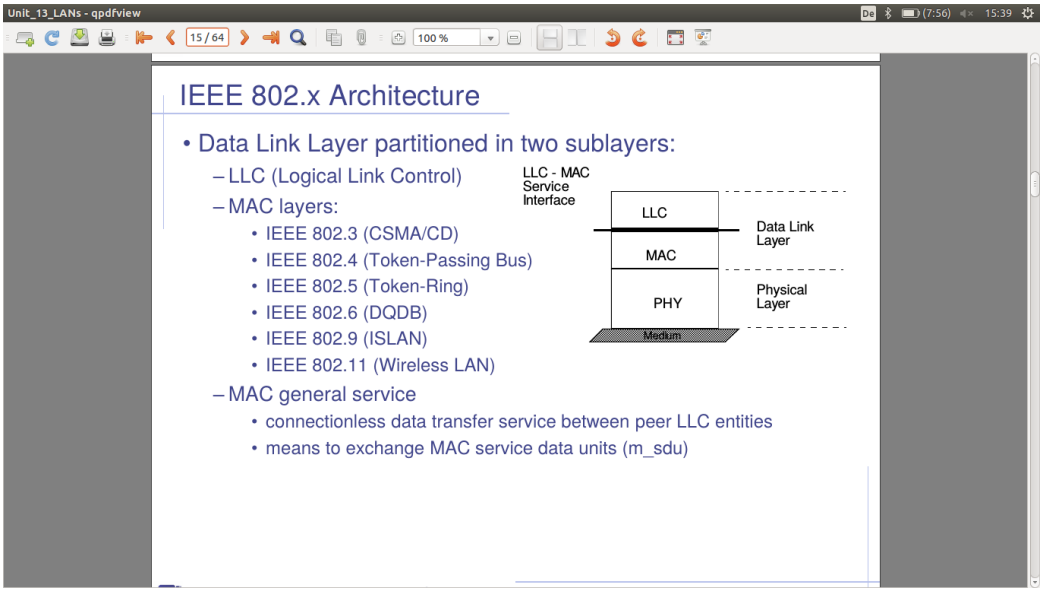
1.12 LANs

- LAN Topologies
  - birectional bus
  - active star
  - Unidrectional
  - IEEE 802 reference Model
  - MAC Service Primitves

- PPP vs. Rings, Stars, Busses
- Bidirection Bus
  - support for multpoint transmission needed
  - collision resistant at receiver needed
  - minimum distance between station is exactly defined
  - Propagation delay depends on the segment length NOT the number of stations
  - purely passive signal propagation (no repeating)
  - a collapse of the one station shouldn't block the whole system: jabber(constant talk into channel) prevention
  - difficult error diagnosis
- Active Star
  - only point to point transmission equipment is needed
  - fairly large total length of cabling
  - distance between end station and hub is limited
  - limitations in hierachy architecture if needed
  - maximal propagation time is independent to the number of stations
  - failure of single link is not critical
  - fairly easy error diagnosis
- Unidirection Ring
  - only point to point transmission facilities needed
  - mixed media in a ring is possible TP and Coax
  - small total cabeling length for large area covered
  - normally total number of segments has to be limited because of a limitation of the joint jitter
  - signals are repeated at each station: single failure, entire system fails
  - information has to be specificly removed from the ring (phantom prevention)
  - simpel diagnosis of failed station
- Local Link control (Sublayer) integrated in HDLC

  - LLC layer operates on peer basis
  - supports connection orientation and connectionless transmission

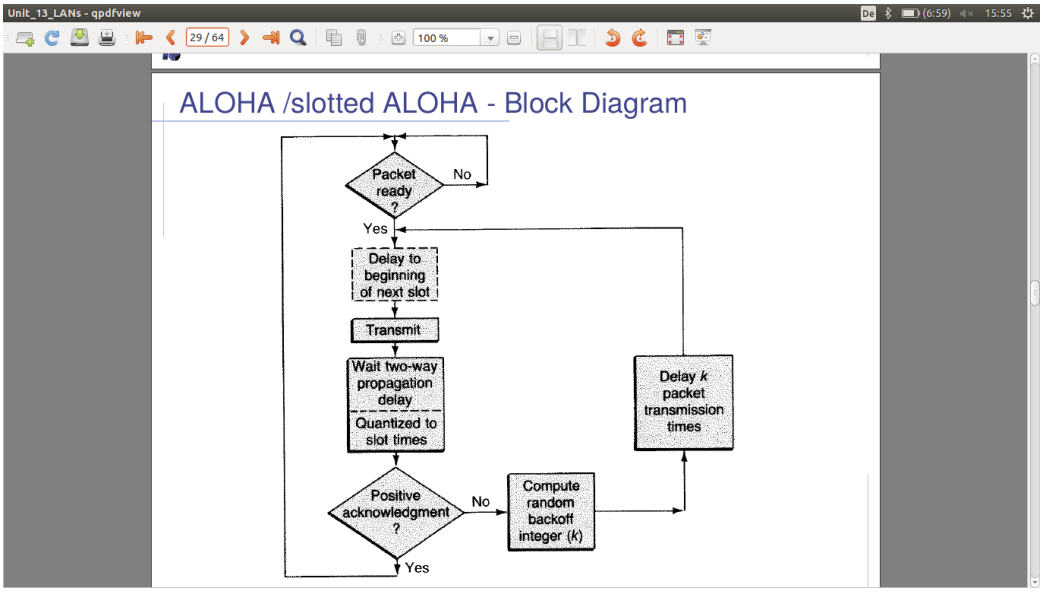
- IEEE 802.x Architecture: explains integration of LLC sublayer to Link Layer and its variations



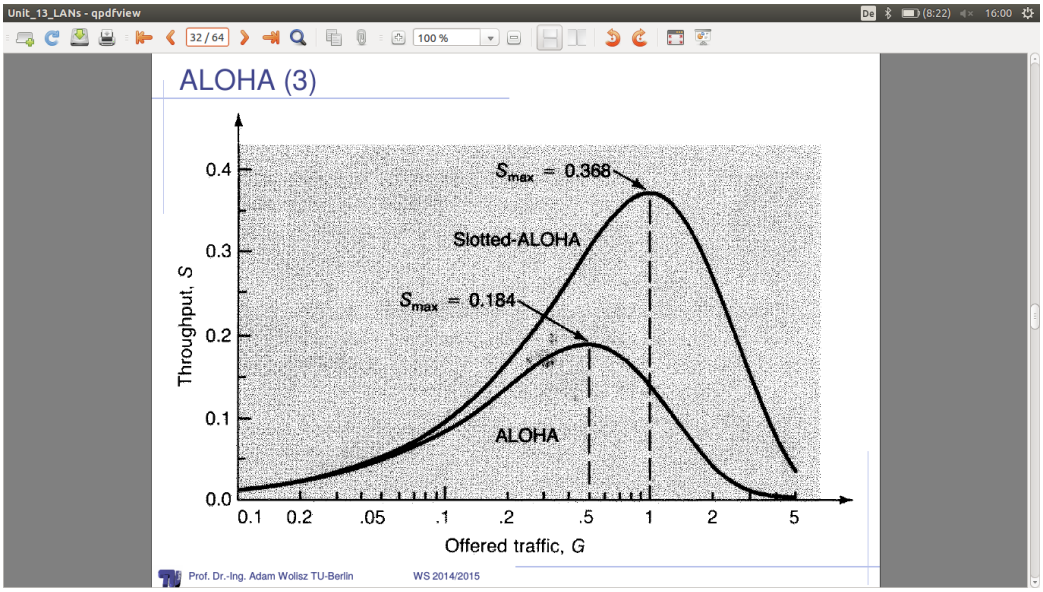
- MAC for Bus/Passive Star - Multiple Access
  - TDMA
  - Polling based approach
  - ALOHA
  - CSMA
- Why mulitple access?
  - distributed stations are sharing a medium: only one can sent at a time
  - how to arrange sharing of the medium
- TDMA: packet ready - wait for assigned slot - transmit packet
- taking turns - MAC - Protocols
  - Polling: Master node invites slave nodes to transmit and turn (polling overhead, latency, single point of failure (master))

  - logical token passing: control token is passed around in a hypothetical ring

- Random Access Approach: talk whenever you like: collision only in the uplink: base station sends one by one. Stochastic delay (ALOHA: random backoff integer) before retrial prevents all senders to collide right again after collision has occured
- ALOHA is such protocol; slotted ALOHA (has sort of collision detection and avoidens)
- blockdiagram of ALOHA



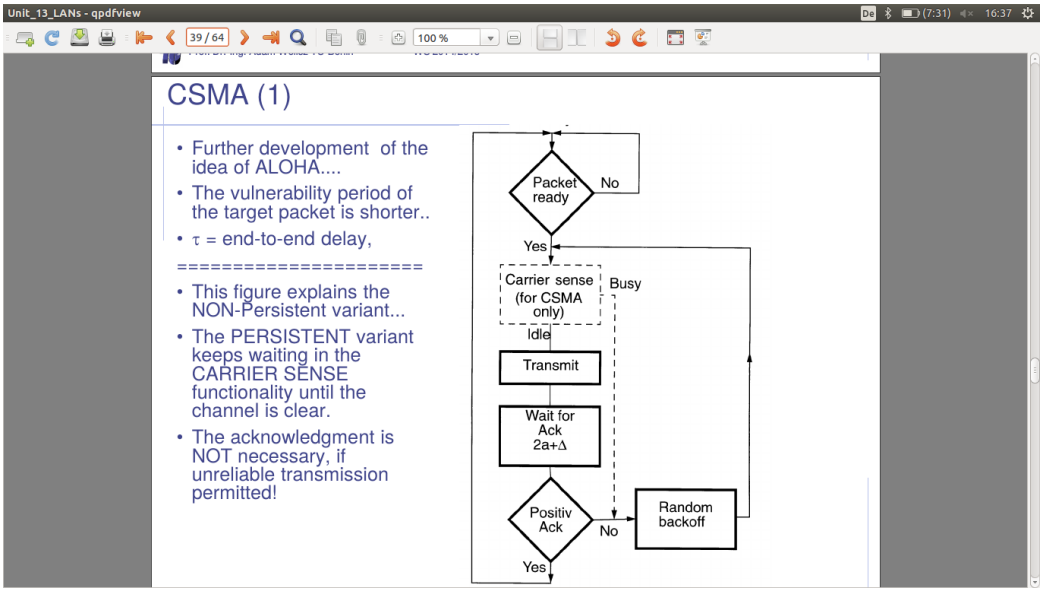
- efficiency of ALOHA and slotted ALOHA
- Stability issues of ALOHA
  - increase of Load can after a certain level decrease throughput up to system collapse
  - has to be avoided by reducing the load
  - the usual measure is to increase the back-off parameter (doubled or exp - raised)
  - check additional reading
- Back-off tuning: increase back off after unsuccesful transmission, decrease after succesful transimission



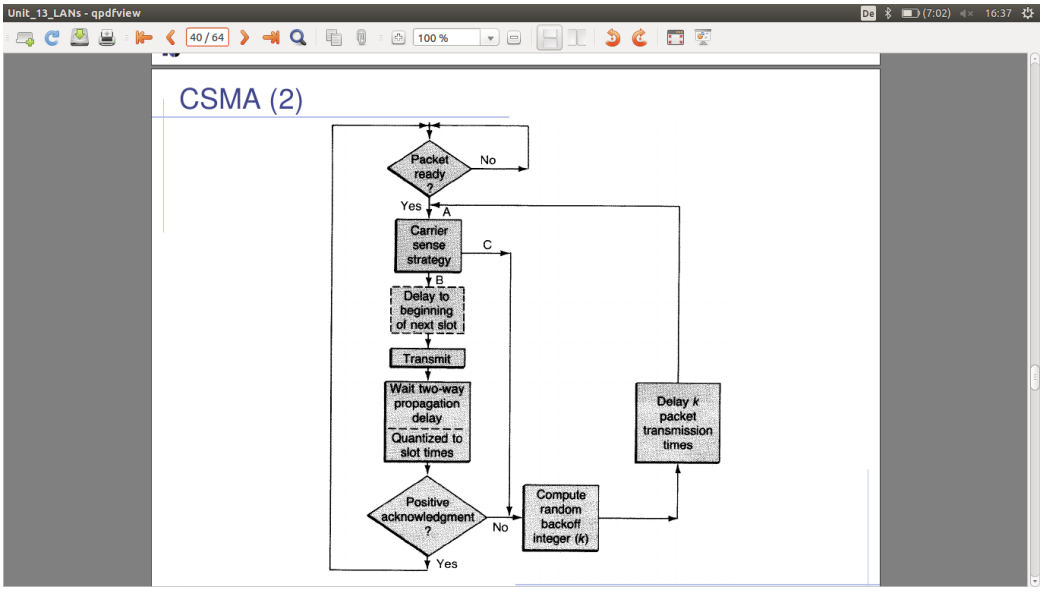
- Pros and Cons of Slotted ALOHA
  - Pros
    1. single active node can continously transmit at full rate of channel
    2. highly decentralized: only need of slot synchronization
    3. simple
  - Cons
    1. wasted slot: idle and collisions
    2. nodes should detect collision in less time than it takes to transmit the package
    3. clock synchronization
- improving on slotted ALOHA
  - fewer wasted slots
  - doesn't waste full slots on collision
  - avoid need of synchronisation (ALOHA-token)
- CSMA - Carrier Sense Multiple Access
  - listen before start of transmission
  - collision can still occur, if node are too far apart and due to propagation delay cannot hear each other

  - if collision occurs: entire transmission time is wasted

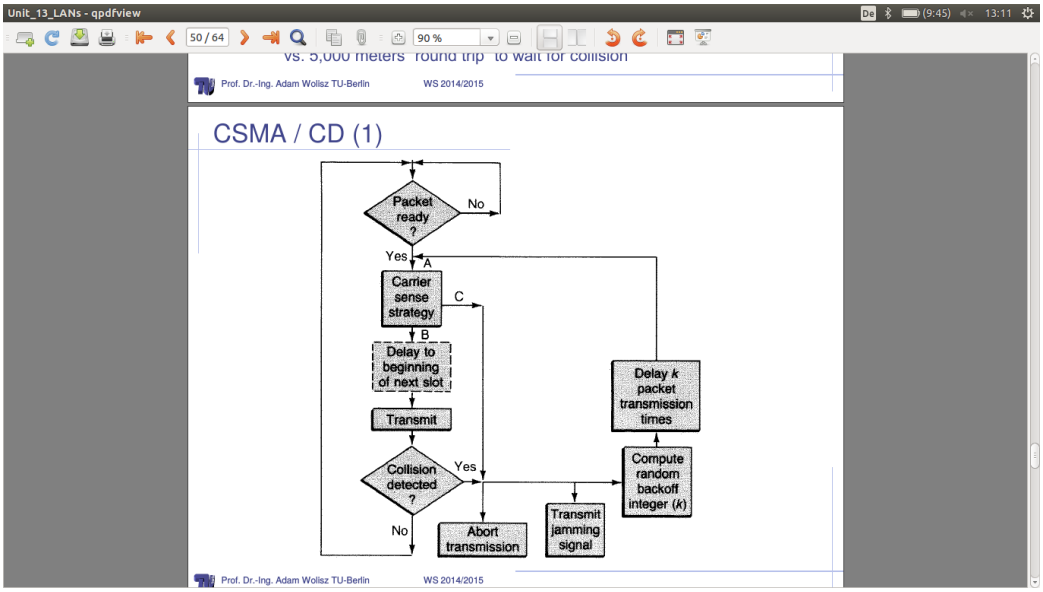
- CSMA as a further development of ALOHA - (would slots ease the problem)
  - non persistent CSMA (figure 1)



  - CSMA with priority Access (slot delay) (figure 2)

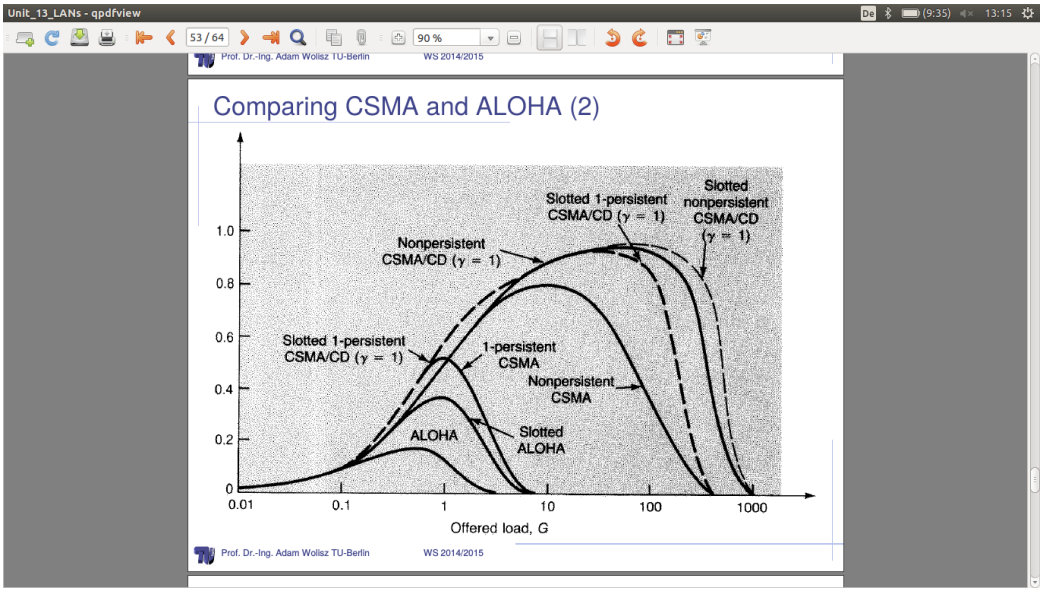


- make sure of difference: persistent (increase back off parameter until channel is clear), non-persistent (random backoff), p-persistent (parameter: enable priority Access) (vgl. UE4)
- CSMA/CD: cut off after collision $2t - e + \text{detectiontime}$
- usually CSMA/CD better than slotted ALOHA, execept in case of the transmission time being shorter than Propagation time (very long channels) (to be savily checked : Tobagi 1 : 1411(12))
- wire circuits: collision detection via comparation input/ouput signals; wireless circuits more difficult, because transmission fails only while transmission
- for collision detection restrictions for minimum frame size and maximum distance are needed
- latency d (propagation time difference between two senders): collision jammer can be seen by other after t + 2d
- so A needs 2d to detect collision with packet B and should therefore KEEP transmitting
- procedure of CSMA/CD if collision occures, pay detailled attention on the closed loop circuit



- Efficiency note: much better than ALOHA and still decentralized: check figure!

$$\text{efficiency} = \frac{1}{1 + \frac{5t_{prop}}{t_{trans}}}$$



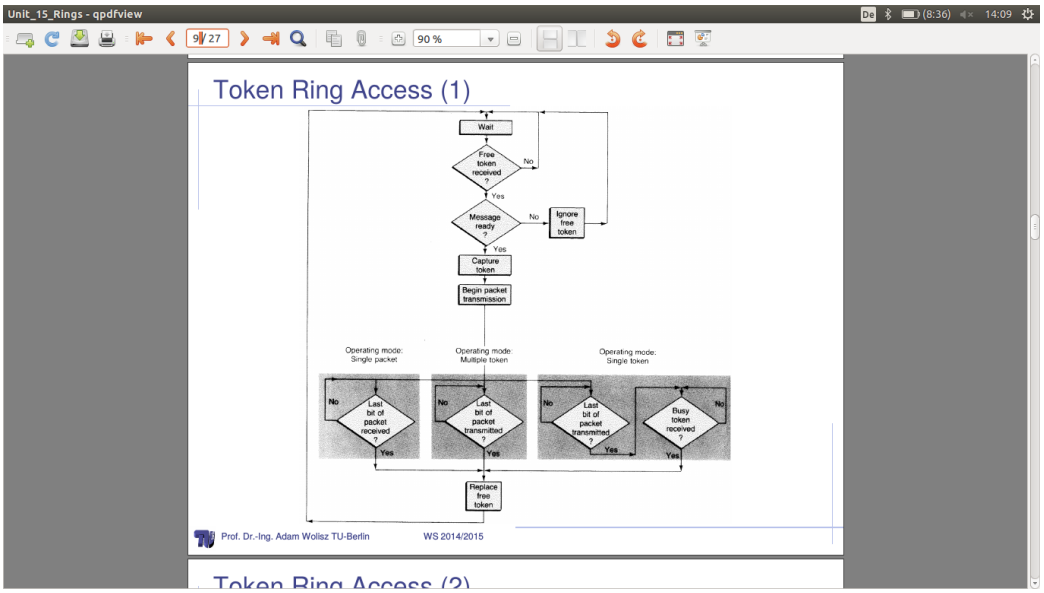
- note CSMA/CD, -/CA, -/CR
- basic problems with wireless communication
  - hidden terminal szenario: a sends to b, c sends to b, a cannot see c, a and c interfer at b
  - exposed terminal szenario: c attemps send to d, while b in range of c, sends to a: communication delayed for C,D delayed without reason
  - forwarding of the signal needed if a wants to talk to d which can only be reached over b,c
- CTS, RTS at aloha to build up communication
- centralised MAC (e.g. polling)
  - pro: simple, efficient,
  - con: complexity of central controller
- usually distributed (decentralised approach)

  - shedule based MAC (highly regular traffic)
    1. shedule exits like TDMA
    2. ressource can be bandwith in physical bandwith (CDMA)
    3. shedule is computed on demand
    4. usally collision, idle and overhearin
  - contention based (irregular traffic)
    1. risk of colliding packets is deliberatly taken
    2. coordination overhead can be saved, resulting in overall transmission
    3. mechanism to reduce impact of collisions

1.13 Rings

- Overview
  - access to ring topologies
  - slotted rings
  - register insertion rings
  - 802.5 IEEE token ring
- removing packet from the ring
  - receiver? no adress has to be processed before forwarding –> delay
  - sender? counting packets to determine which one was sent by you, so no address proceding: smaller delay
- Slotted Ring
  - contains fixed number of bits cycling around the ring
  - each station reads and passes it on to the next station
  - arranged to contain fixed number of slots S containing M bits (bits per slot $= \frac{M}{S}$)
  - slot can be marked as empty or in use (like storage status bit on ROMs)
  - station waits for empty slot, fills it in order to send, after one circulation: marker as empty

- Cambridge Ring
  - fixed number of packets on the ring allows sender to just count passing ones and take theirs of the ring
  - frames are minipackets 38 bit with only 2 byte data
  - largest Cambridge Ring circulates 4 minipackets
  - maximum length 10 Mbit/sec - 100 m between stations
  - response bit; sender can determine if retransmission is needed
  - initialization and maintenance is done by special monitor station
  - monitor passed bits enables the maintenance at the monitor station if sender fails to delete the packet from the ring
  - Cambridge ring is parallely developped with Ethernet
  - main advantages: predictable delay: good support for short messaging
  - disadvantages: monitor sation, higher layer protocol need more than one slot
- Token Ring Access
  - Schematic - wait for token, use it to transmit



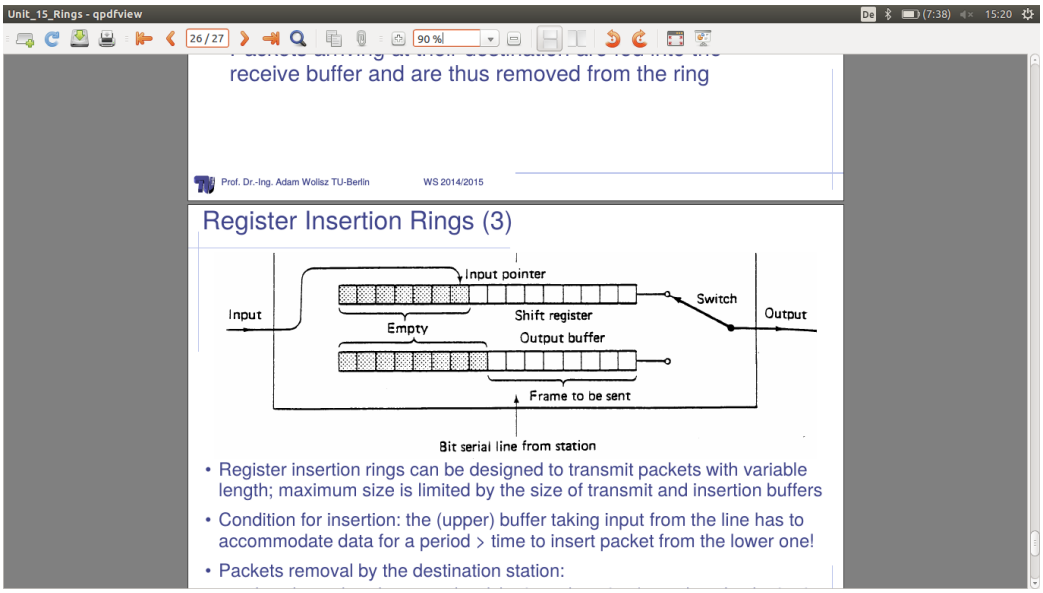
  - multiple token: free token immediatly after last bit of transmitted packet

  - single token: receiving station uses incoming token to retransmit information
  - single packet: sending station re-receives its own packet and the releases the token

- Cabeling
  - station failure prevention by hubs to bypass failed station and keep the ring alive
  - rings started the development of structured cabeling systems (Ethernet followed)

- IEEE 802.5 - Token Ring
  - twisted pairs are the medium
  - every station has a 1 bit buffer: adding a bit delay to the network
  - token is a special 3 byte pattern
  - the right to transmit is obtained by changing one bit in the pattern
  - change of that bit converts token to header and data follow immediatly aferwards
  - SD/ED contain Manchester-code violations to seperate them from data
  - Symbols J,K constant level for the length of one bit: J polarity equals to previous symbols, K opposite Polarity to preceding Bit
  - I-bit (intermediate frame of a frame sequence), E-bit (error detection)
  - the receiving station, copies the message and retransmit it into the ring
  - bit change to indicate succesful transmission
  - A and C bit to indicate Frame Status
  - sender removes packet from the ring an generates new token
  - next station can hold token for a maximum time and does then have to release it
  - Priority Handling via P and C Bit (8 Priortiy Levels for Access Control)
  - Priorities can be used to reserver token: token with higher priority as the station are forwarded

- Ring Maintenance
  - Monitor station is needed
    1. detecting token loss
    2. removing packetfragments after break-down
    3. if needed insert 1 bit delay to keep token on the ring
  - if it fails an other station has to become Monitor
  - maintaing buffer for jiitter alignment
- Register Insertion Rings
  - registers are used to increase capacity: refinement of a slotted ring to increase packet size
  - if station wants to transmit packet: it is loaded into transmit buffer (register), and are then injected into the ring (Receiver-Switch and Sender Switch closed datta travels from point A to B)
  - if ring breaks: registers can be used to store bits...

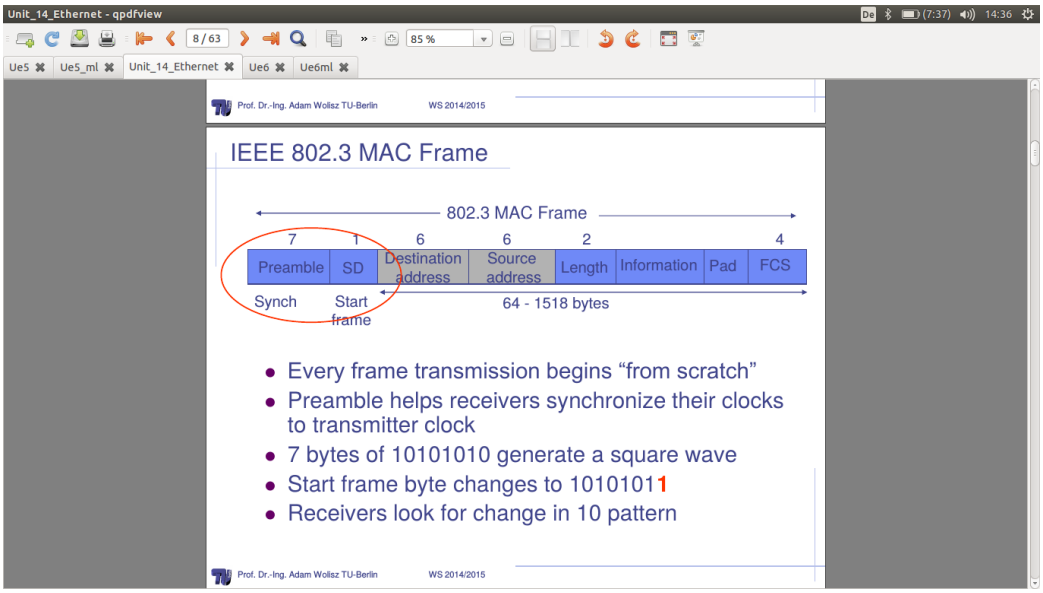


- Packet Insertion Rings - theoretical posibility (RPR for MANs)

1.14 Ethernet

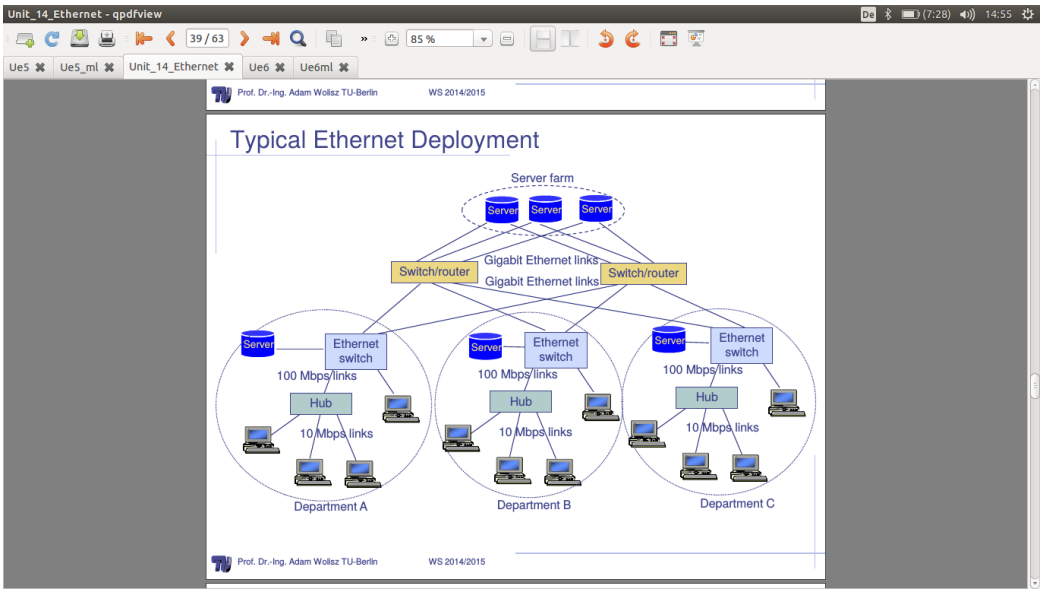
- the yellow wire

- IEEE 802.3 MAC: Ethernet - mac protocol
  - CSMA/CD
  - slot time is the critical system parameter
    - upper bound on time to detect collision
    - upper bound on time to acquire channel
    - upper bound on length of frame segment generated by collision
    - quantum of retransmission scheduling
    - roundtrip propagation, MAC jam time
  - truncated binary exponential backoff
    - for retransmission $n < r < 2^k$
    - give up after 16 retransmissions
- *each decade increase in Bitrate is accompanied by a decade decrease in distance*
- **Ethernet IEEE 802.3 MAC Frame**

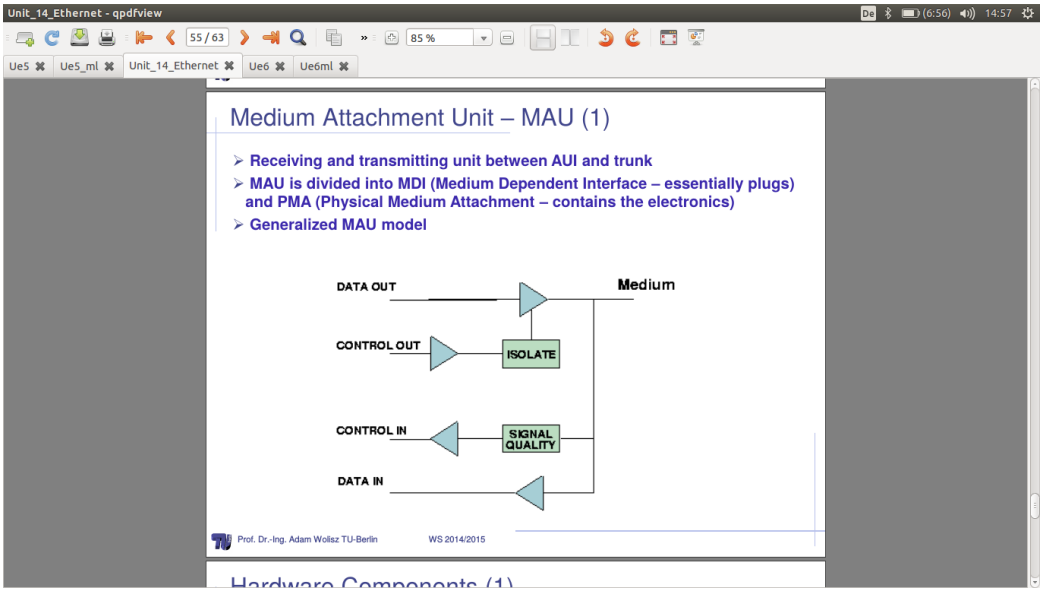


- padding ensures min frame size of 64 Bytes
- connector standards: look up the praktikum skripte
- access interference and recovery: like CSMA/CD (extra jamming signal to indicate collision, which can be generated by any station)

- collision detection on medium, monitoring by every station: normalised signal amplitude, if collision occurs voltage on medium is twice than if it was generated by a single sation
- Media Access Control - Basic Functions
  - data encapsulation/decapsulation
    - framing
    - addressing
    - error detection
  - Media Access Management
    - medium allocation (collision avoidance)
    - contention resolution (collision handling)
- Bridging improves scalability: seperate collision domains
- typical ethernet deployement



- medium attachment unit - MAU

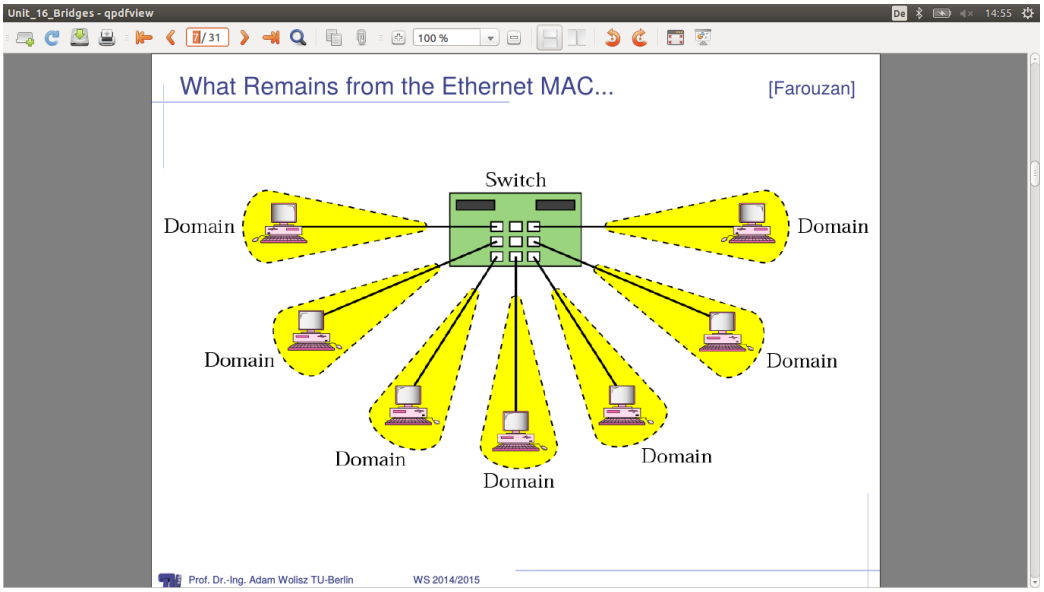


1.15 Bridges

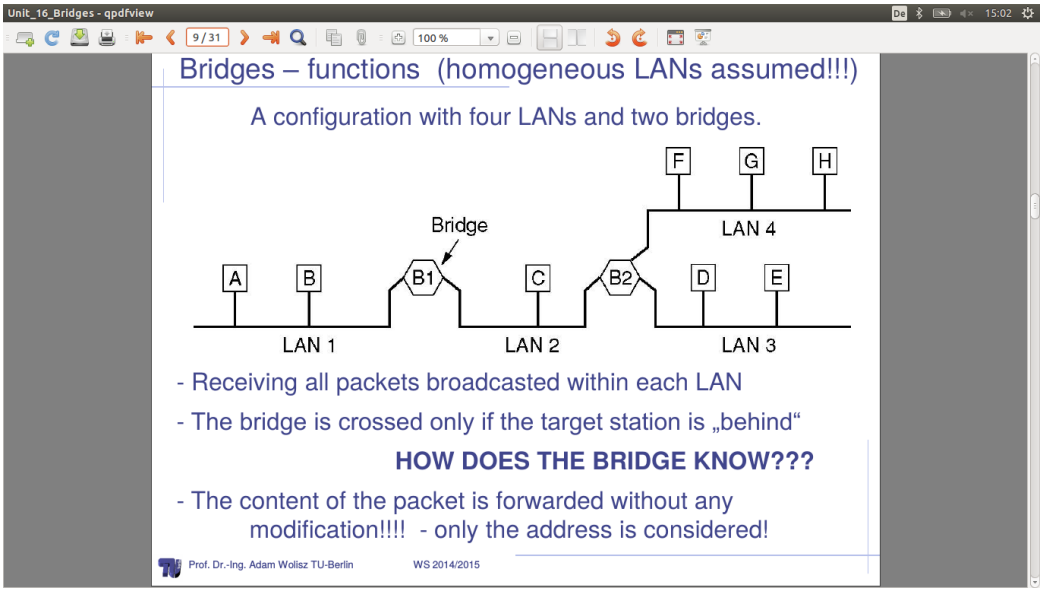
- Ethernet: minimum frame size
  - a minimum frame size equal to number of bits transmitted during one round trip is required to detect all collisions
  - slot size: number of bits transmitted by a source during the maximum round trip time for any Ethernet network
  - rule in Ehternet: all frames must be as large as slot size
- Physical Propagation Limits
  - end to end delay isn't achievalble: use of **repeaters**
  - the total delay: propagation + delay on all repeaters must not exceed the limiting value
  - higher bit rate is problematic for efficiency: (either shorter distance or larger packets)
  - **Errors result usually from colliding Packets**
- Star topology with a centered switcher is better than a hub
- Hub vs. Switch:
  - parallelism in transmission holds only with unicast traffic

  - broadcast goes over all traffic
  - multicast goes over all ports: each reciever makes the decision for himself

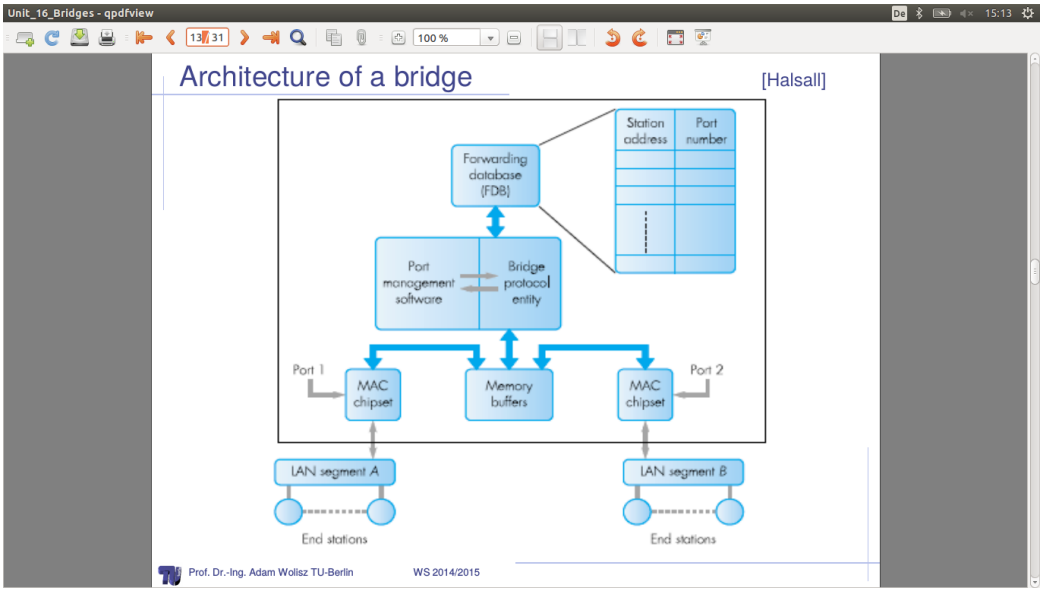
- remains of Ethernet-MAC



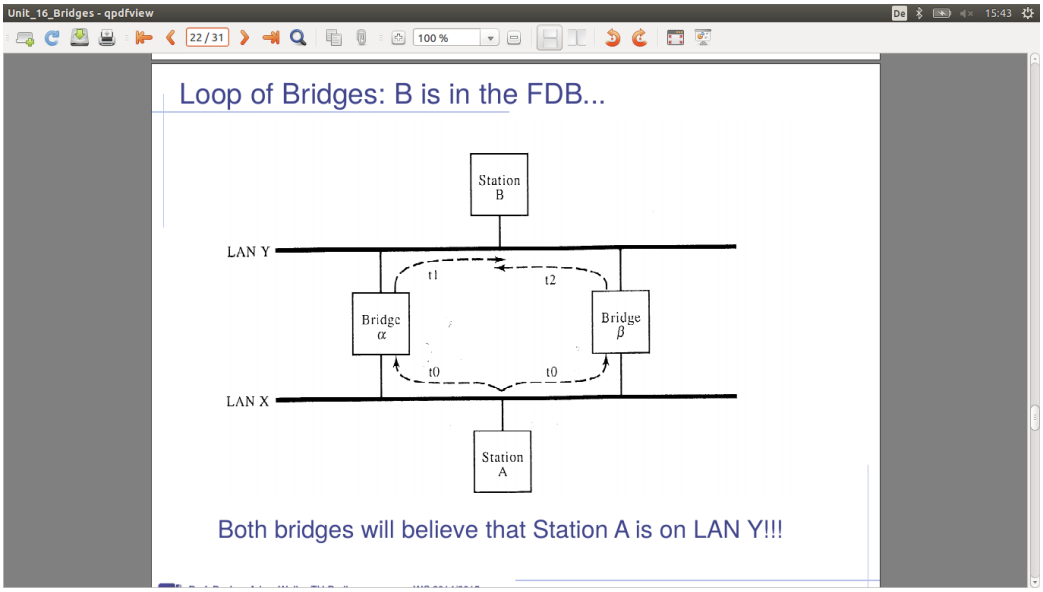
- Switchapproach vs. Shared Medium
  - on a shared medium the sender is aware of losses and restarts transmission
  - this means no losses but variable delay because of various retransmission
  - in a switch topology there is need for buffering and forwarding because of losses without awareness of the original sender
  - buffers can overflow: –> task for higher prootcols
  - broadcast: for builiding routing tables
- bridges function: connecting LANs (forwarding only if target is behind)
- bridges work on protocol layer PHY and MAC(2/3)
- features of bridges
  - decrease of traffic on a LAN segment vs. broadcast with repeaters



  - removal of physical restrains: total number of attaches stations and segments can be readily increased
  - due to their MAC subaddress relaying function they are transparent for higher protocol layers: so they can be used connecting LANs with different protocol stacks
  - - additional store and forward delay (repeaters don't)
  - - bridges might be overloaded
- architecture of a bridge
- bridges is invincible for the stations
- Bridges have to learn the topology by building up a Forwarding Data Base (FDB)
  - an entry for each host who is origin of any received packet contains the senderID, port on which packet has been received and time of receive
  - if sender is already in the FDB: update
  - if sender isn't include
  - topology of the network **change** so **this information is soft state**
  - this timer is set to 15 seconds default



- assumption: most station will send something thus generating an entry in FDB: if entry isn't there **Broadcast!**
- learning process of the FDB takes time initially but: transmission consists of more than one packet: so all furhter and yet ACK can go back directly
- Flooding : Broadcast process to learn topology
  - good and quickes way for distributing of information: independent of the topology:
  - general more packets send than needed
  - hop counter (to calculate quickest route)
  - implies multiple arrival of one package
  - constrains are needed
    1. lifetime of a packet can be limited
    2. nodes can store information (packetID, source address): and deny retransmission
    3. looking into packet is necessary
- Loop of Bridges: Important
- **Preventing Loops** - Spanning Tree Protocol
  - network as a graph

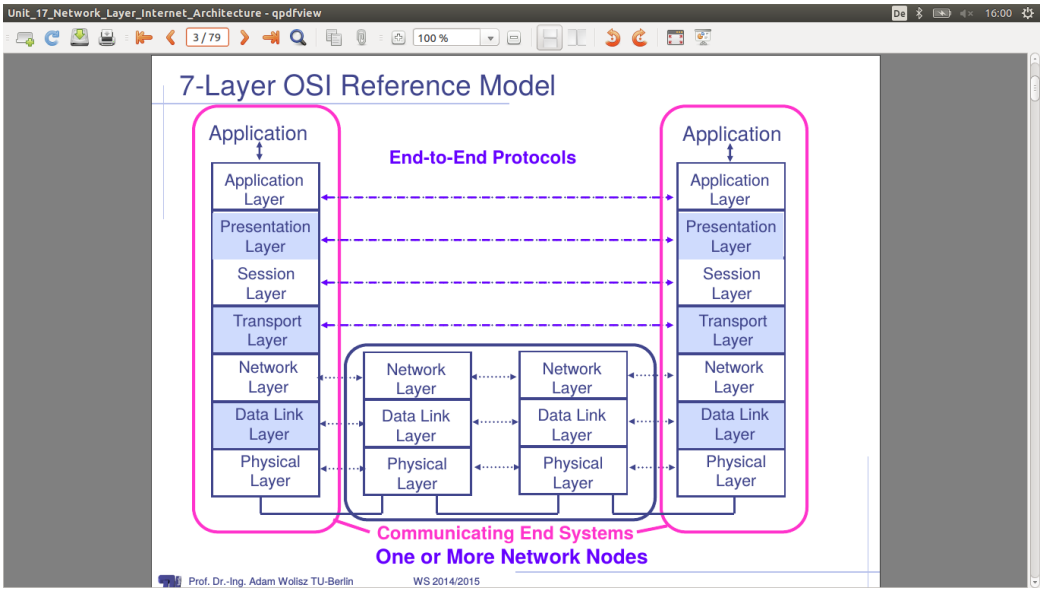


  - Spanning (all briges are included) Tree (no loops) Protocol finds a Subgraph that spans all the vertices without loops
  - determine which bridge is the root of the tree (lowest MAC-address: ID): each bridges turns of ports which aren't part of the tree

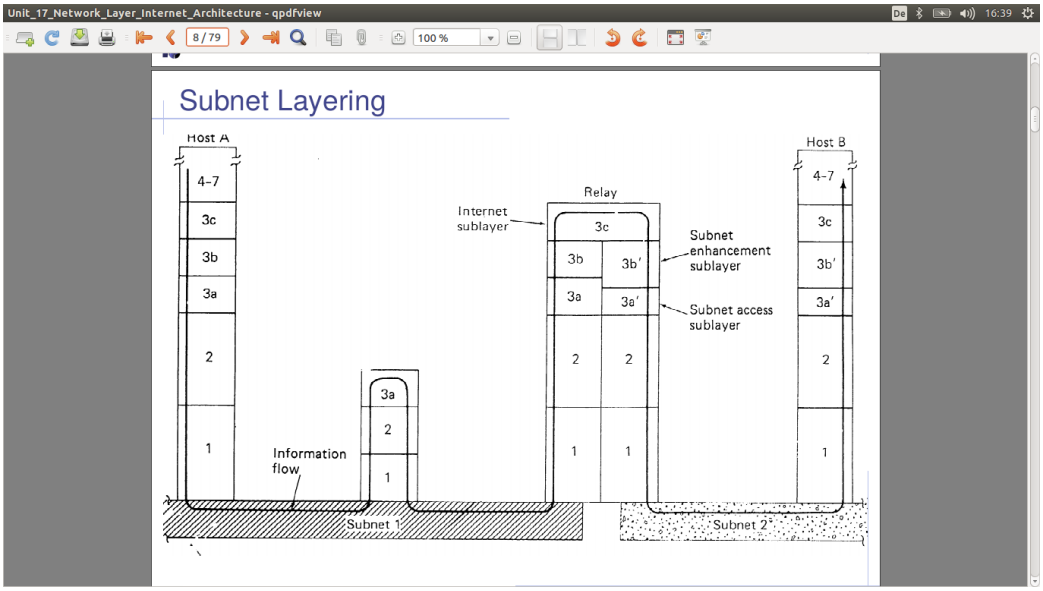
- bridges can adapt Ethernet from different standards and combine them: (e.g. form 802.11(Wlan) to 802.3(Ethernet))

1.16 Network Layer - Internet Architecture

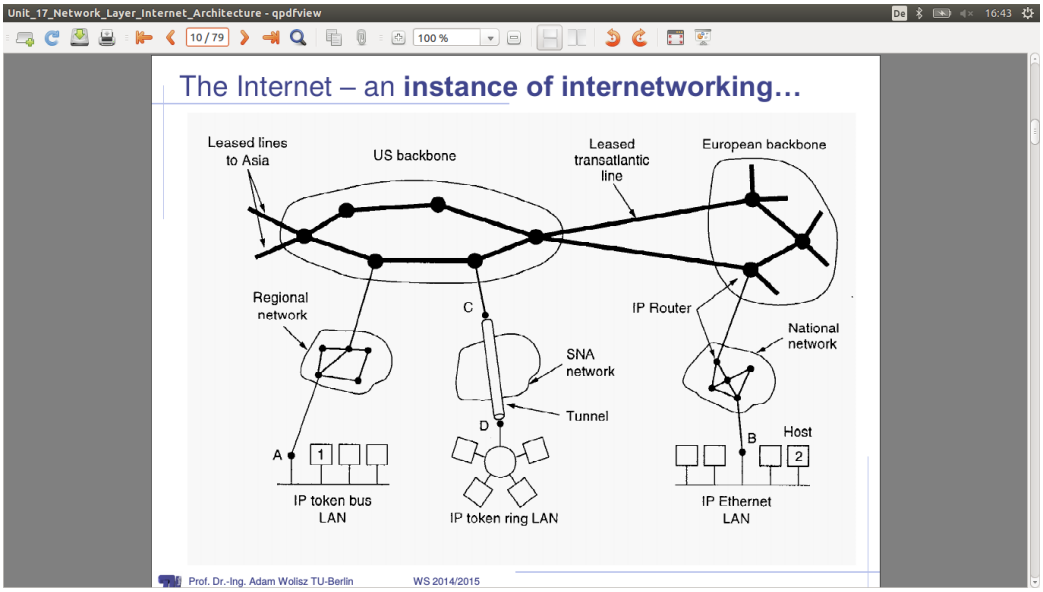
- 7 Layer OSI - reference Model
- Function of Network Layer:
  - deliver data from network adapter at source to network adapter at the destination: independently from the connection in between
  - includes function like: Addressing, Routing and Forwarding, Sheduling in the network elements
  - includes optional features like: congestion control (overload control), segementation and reassembly (packet length adjustment), security
- Problems on connecting individual networks: 1. scalability (flat addressing) and 2. heterogenity of networks aren't supported



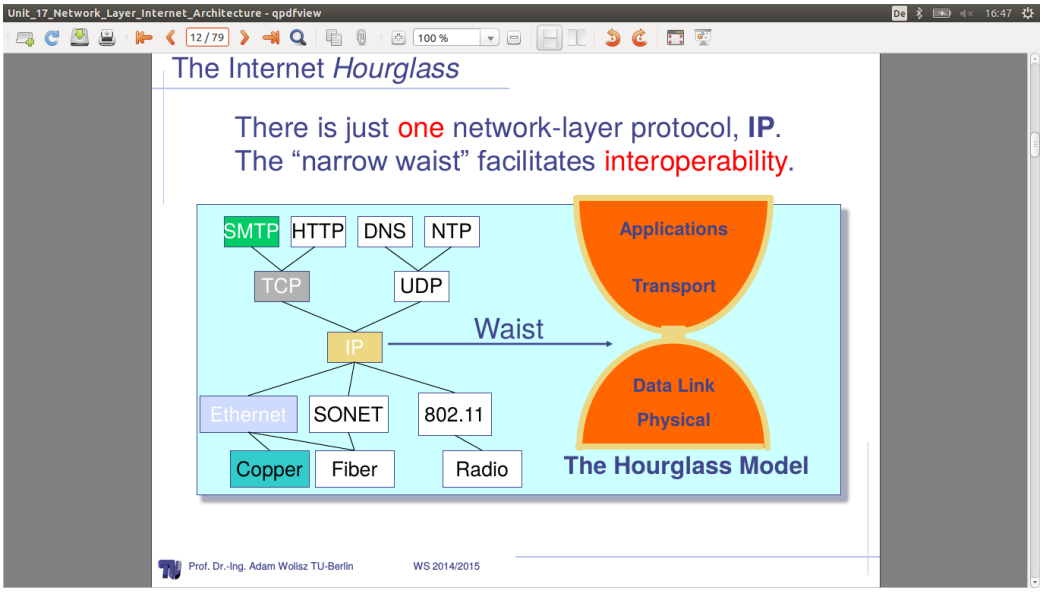
- a service model for the interconnected network is needed
  - how to: address, route
- bridging eases the Problem for flat addressing but **not** scalable for very large networks
- subnet layering: Adaption is needed if both have to work on the connection: done by relays



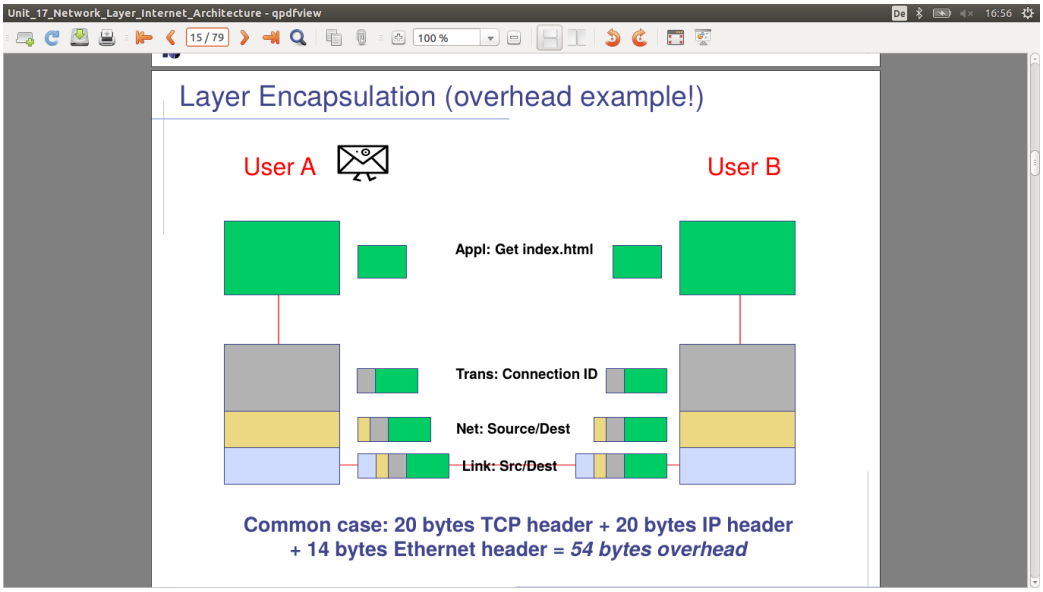
- internet: instance of networking: notice: IP router, token ring, token bus, Ethernet



- Elements of internet philosophy
  - type of layering
  - placing the functionality
  - fate sharing
- The internet-hour-glass: just one network layer protocol: IP. the narrow waist facilitates interoperability
- IP is the highest protocol which is implemented at both: host and router (bridges end at link layer)
- layering in the internet
  - interconnection over networks rather than nodes: different technologies could be included
  - any network that supports IP can exchange packets
  - allows application to function on all networks (seperation of APPLICATION and COMMUNICATION) - socket interface as principle transport service interface
  - supports innovation itself above and below IP: but IP is holy



- header grows with depth into layer model: header bigger bigger ... transmission ... header smaller smaller



- internet philosophy: placing network functionality
  - some types of network functionality can only be implemented end-to-end

  - end host must statisfy the need of a connection by himself without the networks help

- reliable file transfer
  - solution 1: make each step reliable and the concatenate them
  - solution 2: end-to-end check and try again if necessary (complete, because of **NO** need of reliability from lower layers)

- DON'T implent a function at the lower levels of the system, unless it can be completly implemented at this level: –> so keep the network layer as simple as possible

- E2E principles relied on fate-sharing
  1. invariants only break if end-points themselves break
  2. minimize dependence on other network elements
  3. this should dictate placement of storage

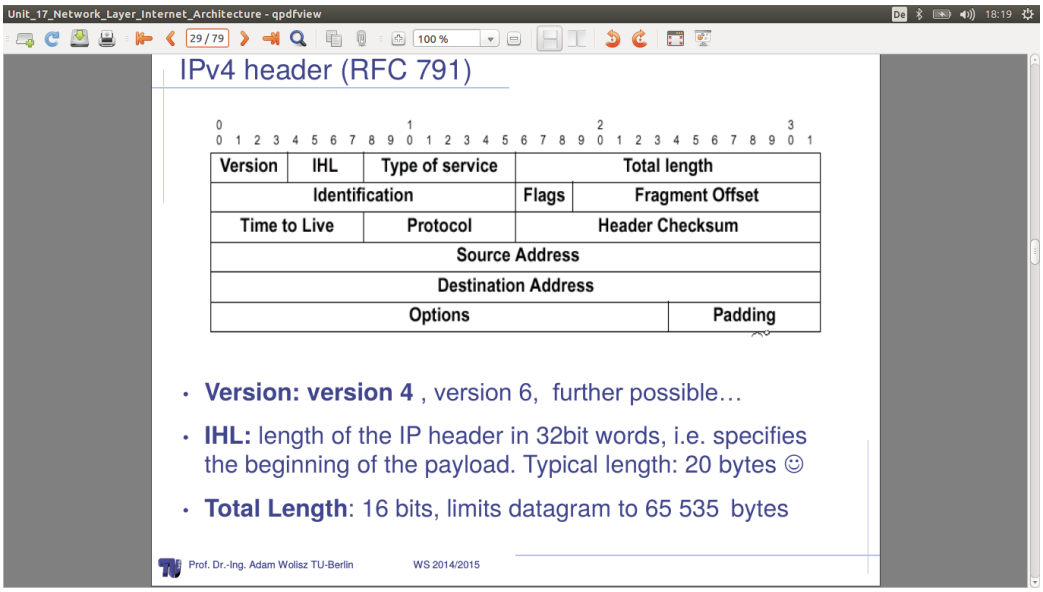
- **Fate-sharing**
  information fails - link fails - doesn't matter because everyone knows
  ZITIERT AUS WIKIPEDIA: Fate-sharing is an engineering design philosophy where related parts of a system are yoked together, so that they either fail together or not at all. Fate-sharing is an example of the end-to-end principle. The term "fate-sharing"was defined by David D. Clark in his 1988 paper "The Design Philosophy of the DARPA Internet Protocolsäs follows:

  The fate-sharing model suggests that it is acceptable to lose the state information associated with an entity if, at the same time, the entity itself is lost. Specifically, information about transport level synchronization is stored in the host which is attached to the net and using its communication service.
  A good example of fate-sharing is the transmission of routing messages in routing protocols such as BGP, where the failure of a link or link interface automatically has the effect of terminating routing announcements through that interface, ultimately resulting in the tearing down of the state for that route at each end of the link. Similar considerations apply to TCP.

- Internet vs. POTS
  - everyone can develop application and provide funtionality vs. operator had to introduce functionality
  - application on IP: www, e-mail, ip-telephony, –> multi-Service network

Basic Internet Protocol IPv4

- basics - limited scope of IP
  - adressing
  - forwarding
  - framentation
  - NO: end-to-end reliability, overload (packets are just dropped), sequencing
- IP supports unicast, multicast, broadcast
- IPv4 header RFC 791

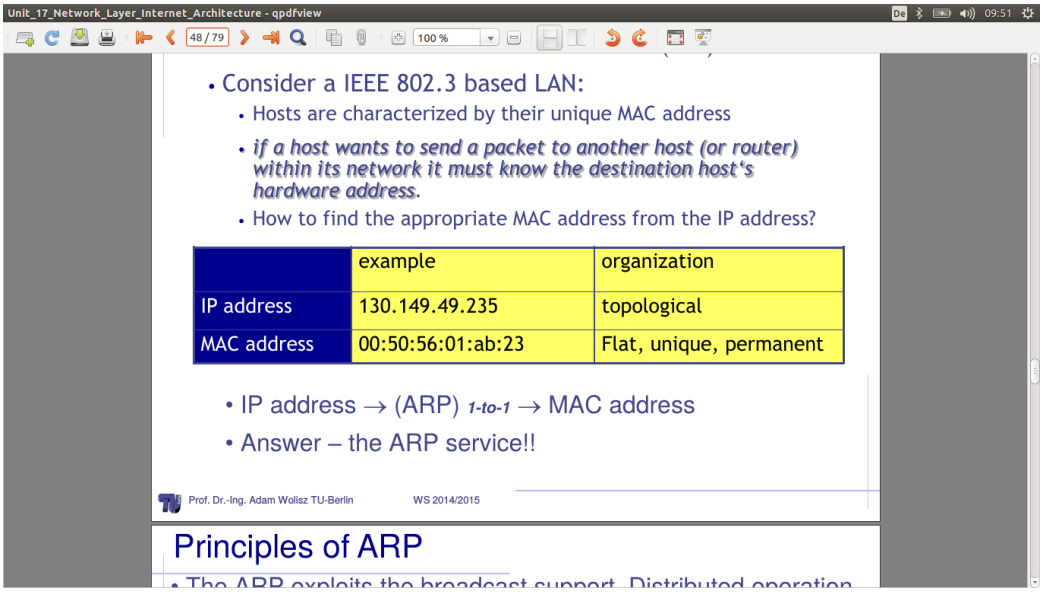


- IP header - further fields
  - identification: used by sender to mark inidvidual packets
  - TTL: lifetime is decremented by each node: if TTL == 0 packet is destroyed and a notification is send to the sender by ICMP
  - header checksum: 16 bits: (not the same as CRC which is added in the trailer at level PHY)
  - source routing; timestamp information
- IP addressing scheme

  - it identifies an INTERFACE not a host
  - IP uses 32 bit address (4 Byte)
  - special address: 127.0.0.1 loopback. and 255.255.255.255 local broadcast
  - hierachical addressing: network <-> host : network address for large scale routing and host address for local routing

- Generating Addresses
  - ISP gets address block from its own provider or from one of the 3 routing registers
  - Subnetting in order to provide better routing performane
  - subnets lead to multiple LANs with single IP - Network Address

- Each subnet needs a Subnet number and a Subnet Mask to define which bits are relevent
- bitwise AND operation eases to determine if destination is on my subnet
- CIDR = Classless InterDomain Routing
  - CIDR allows networks to be assigned on arbirary bit boundries
  - use **aggregation** - provide routing for one a large number of networks with one common prefix
  - reduces routing tables and maintains connectivity
  - subnet part first - then host address (eases routing: look into it and forward it as quickly as possible)
- ARP (-cache) (Address resolution Protocol) - Using IP in a LAN
  - host are characterised with unique MAC-address (IEEE 802.3)
  -
- ARP - principles
  - exploits broadcast support; distributed operation per host (a special server only if no broadcast)
  - all host can send ARP requests and ARP replies



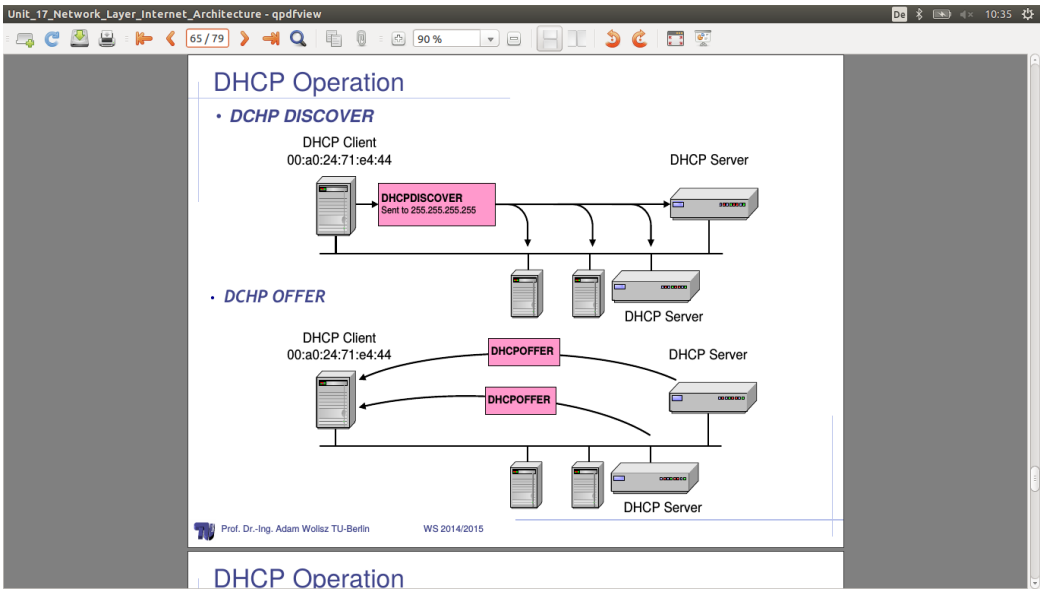
  - already resolved addresses are stored in an ARP cache: Using Soft state principle (ARP cache: timer expiration resets to default (20 minutes standard time to expiration: prevents ARP request from happing to often))

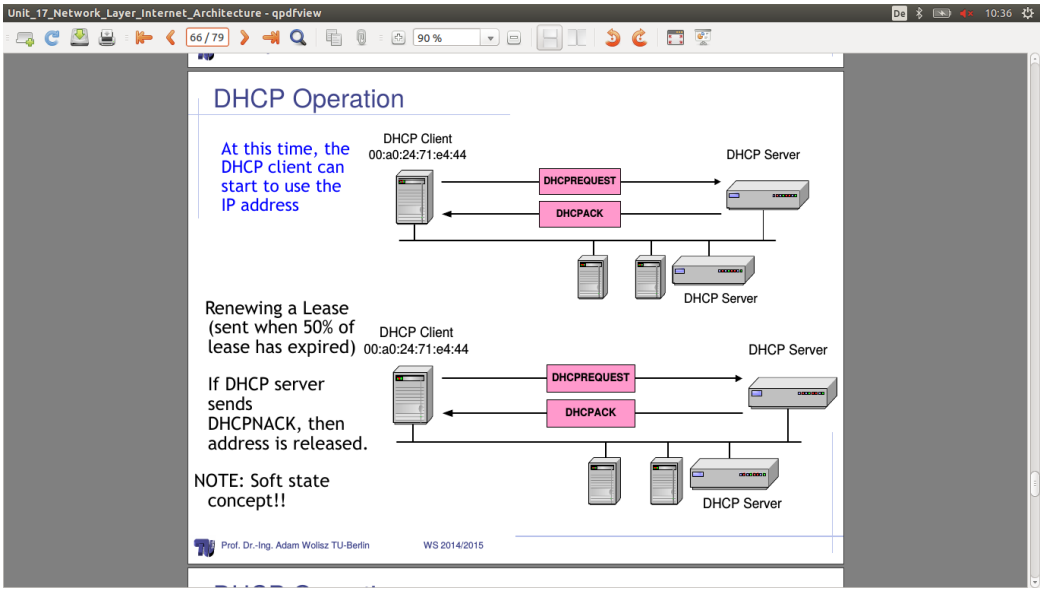
- Address Translation with ARP
  - server broadcasts an ARP request to all stations on the network: what is the hardware address of router 137?
  - arp reply: contains hardware address of router 137
- proxy - ARP
  - Host or router responds to ARP request that arrives from one of its connected networks for a host that is on another of its connected networks
- ARP gives up eventually if there is a ARP Request for a non existing host
- on some operating systems (linux) a host periodically sends ARP requests for all addresses listed in the ARP cache. this refreshes ARP cache content but also introduces traffic
- Vulerabilities of ARP
  - ARP doesn't authenticate requests or replies. ARP request and ARP replies can be forged

  - ARP is stateless: replies can be send without corresponding request
  - ARP must be updated if request/reply comes in
  - Exploitation : a forged ARP request leat to updated ARP -cache with forged entry (ARP poisoning)
  - this can be used to redirect IP traffic to other hosts

DHCP - Dynamic Host Configuration Protocol

- How to get an IP address
  - a laptop has a ethernet interface (with MAC address) and WLAN interface with own MAC address
  - IP address is dependent on who is the Service Provider
- Dynamic assignment of IP addresses
  - IP addresses are assigned on demand
  - avoid manual configuration
  - support mobility of Laptops
- BOOTstrap Protocol (BOOTP)
  1. host can configure its IP parameters at boot time
  2. 3 services: IP address assignment, detection of IP address for a serving machine, boot file name which is then executed by the client machine
- DHCP (since 1993) extends BOOTP: Extensions are
  - support temporary allocation ('leases' IP addresses)
  - DHCP client can acquire all IP configuration parameters
- DHCP is the preferred mechanism for dynamic assignment of IP addresses
- DHCP can interoperate with BOOTP clients
- DHCP operation
- DHCP release: DHCP client releases IP address
- DHCP server does not have to be available in each network: (relays are sufficient)
- Requirements for DHCP

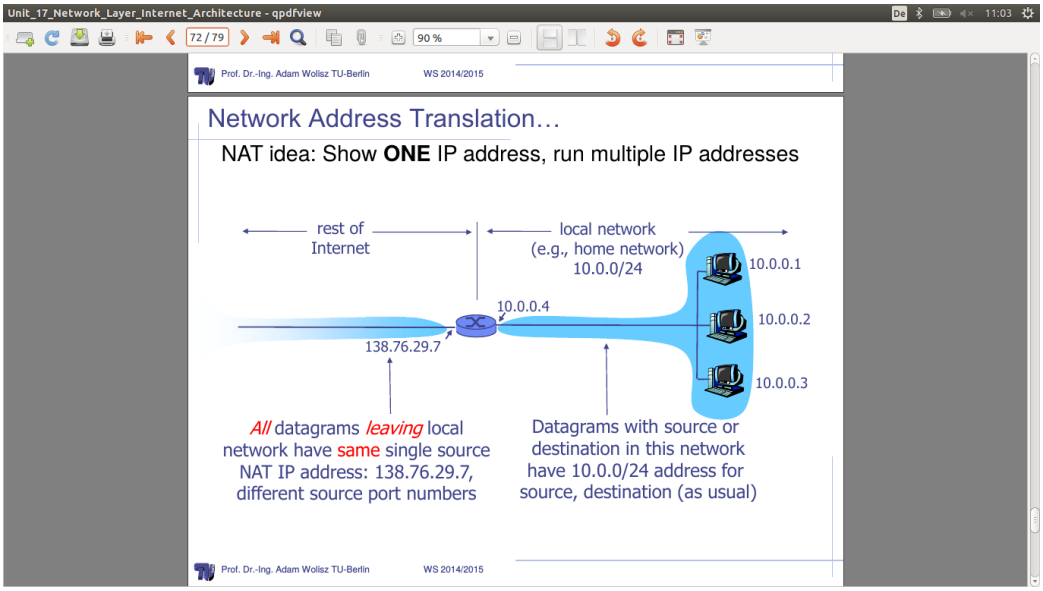

Unit_17_Network_Layer_Internet_Architecture - qpdfview
65 / 79
90 %
DHCP Operation
• DCHP DISCOVER
DHCP Client
00:a0:24:71:e4:44
DHCP Server
DHCPDISCOVER
Sent to 255.255.255.255
DHCP Server
• DCHP OFFER
DHCP Client
00:a0:24:71:e4:44
DHCPOFFER
DHCP Server
DHCPOFFER
DHCP Server
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
DHCP Operation


Unit_17_Network_Layer_Internet_Architecture - qpdfview
66 / 79
90 %
DHCP Operation
At this time, the DHCP client can start to use the IP address
DHCP Client
00:a0:24:71:e4:44
DHCP Server
DHCPREQUEST
DHCPACK
DHCP Server
Renewing a Lease (sent when 50% of lease has expired)
DHCP Client
00:a0:24:71:e4:44
DHCP Server
If DHCP server sends DHCPNACK, then address is released.
DHCPREQUEST
DHCPACK
DHCP Server
NOTE: Soft state concept!!
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015

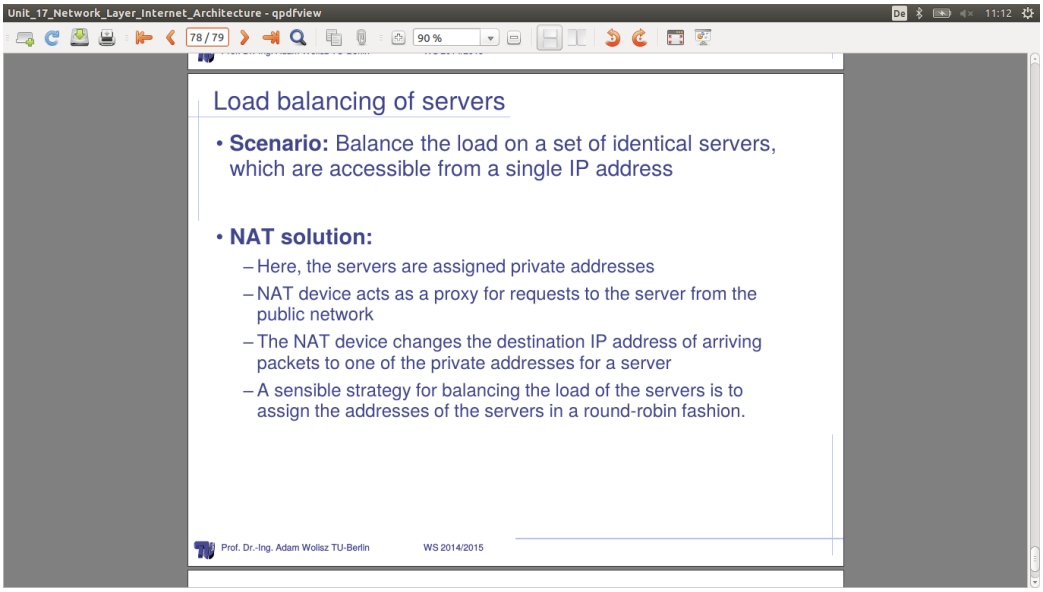
- guarantee that any specific network address will not be in use by more than one host at a time
- retain host configuration across reboot: host will be assigned the same configuration parameters (e.g. network address) in response to each request
- retain host configuration across server reboots. same configuration parameters thus restart of DHCP mechanism
- allow automatic assignments of configuration to new hosts
- support fixed or permanent allocation of configuration parameters to specific hosts (server functions)

NAT - fighting limits of address space

- enlarge IPv4 address space: IPv6 because of address shortage
- prevent home users from running servers at home
- each IP with ISP cost money
- hide internal topolog to outside world (security)
- NAT idea: show one IP address and run multiple IPs

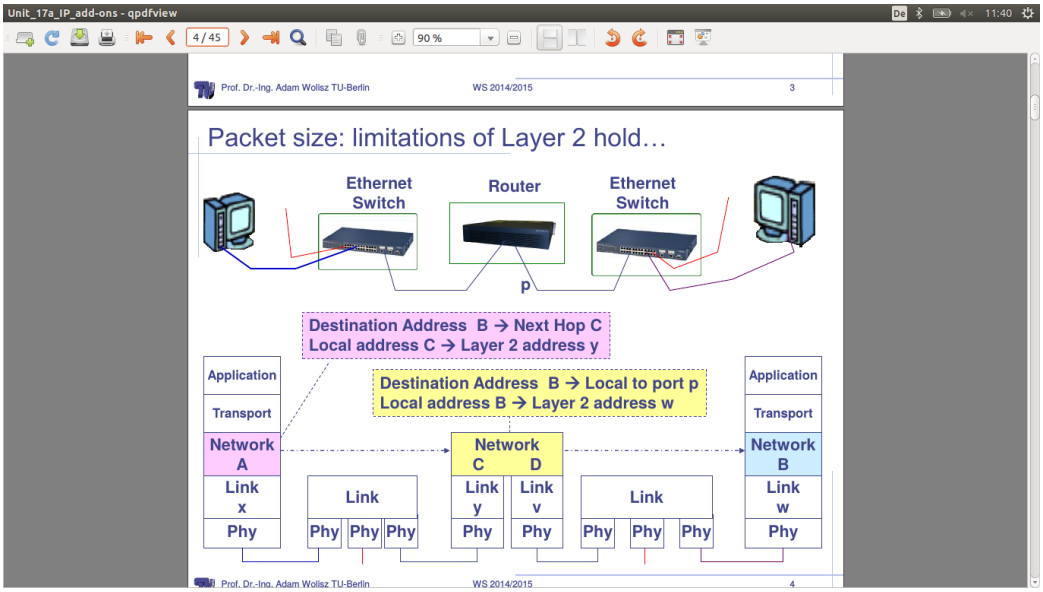


- NAT: home network via NAT hub: IPb and IPc invisible outside only IPa visible
- NAT: trick use TCP port to distinguish computers
- NAT device can serve as a proxy to balance load on servers accessed by the same IP



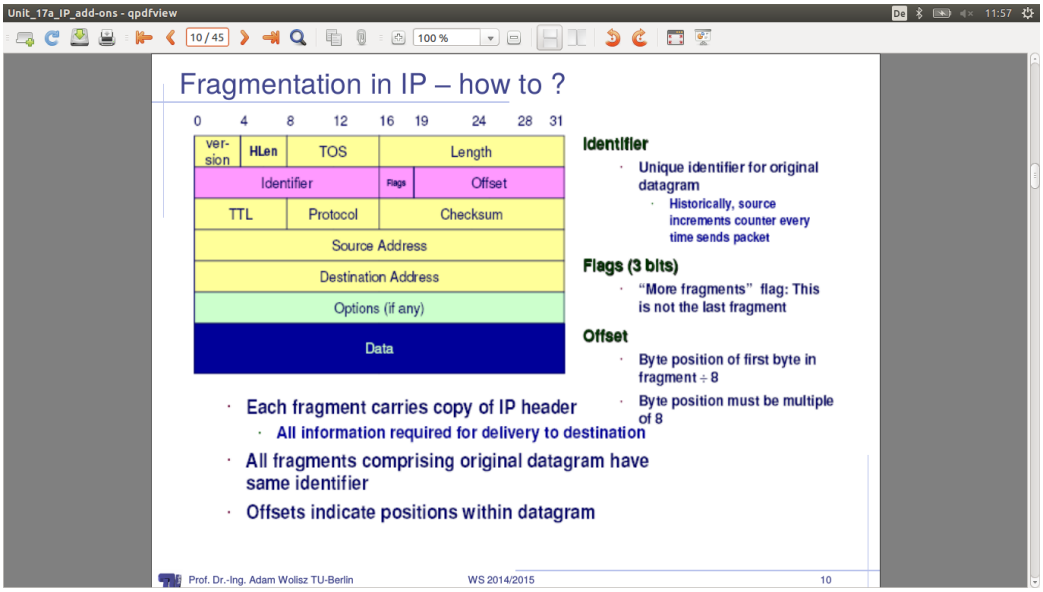
1.17 IP add-ons, ICMP, Mobile IP, IPv6

- Fragmentation - dealing with different Packetlength on different Networks
  - Adaption (e.g. Packet length) in connecting routers
  - Ethernetpacket: Payload maximum length 1500 Bytes
  - Transfering a packet on a network:
    1. every internet module must be able to forward 68 octets
    2. every internet destination must be able to receice 576 octets
  - **fragmentation**: division of a long packet in pieces
  - alternatively:
    1. use path featured discovery
    2. short enough packet



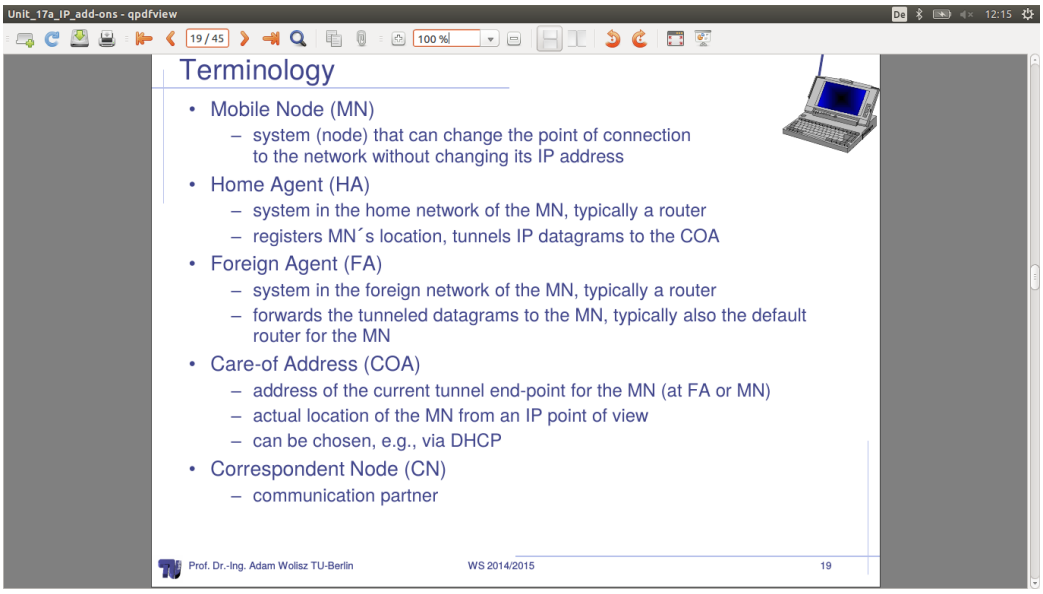
  - REassembly at the destination and not on the path, because this isn't fixed/constant

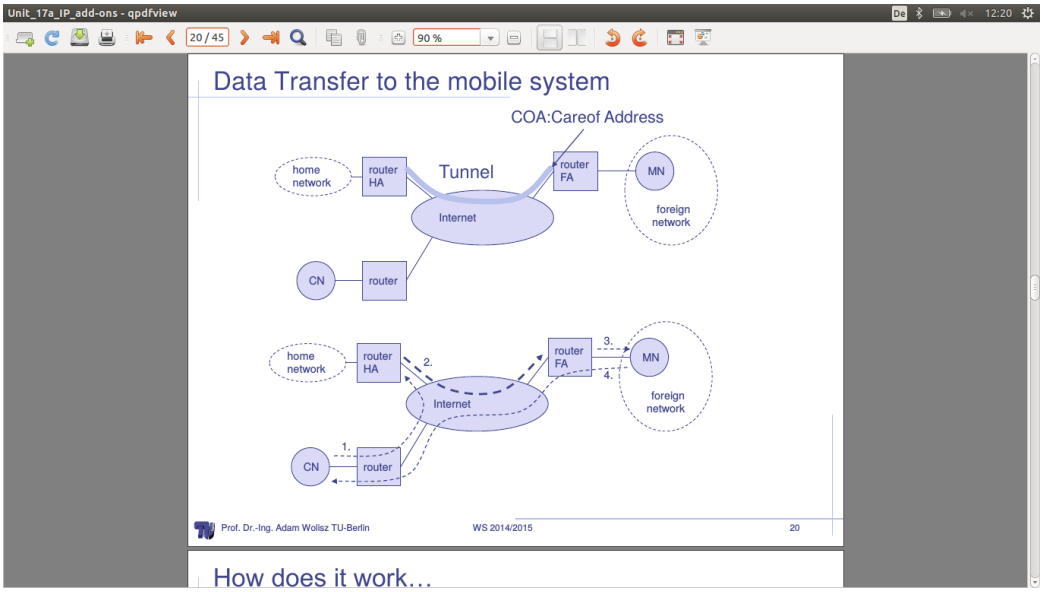
- Fragmentation and Reassembly Options
  - reassembly is done by destination not by router
  - in case of non fragmented option: Packets are dropped at the router
- Fragmentation in IP: sequence numbering in otherwise same header allows reassembly at the destination
- characteristics on internet fragmentation
  - decentralized (independent choice of MTU)
  - connectionless
  - fail by dropping packet, destination can give up reassembly, no need to indicate failure
  - keeps most of the work at Endpoints
- fragmentation is harmful
  - uses ressources poorly: forwarding costs per packet; worst case if packetlength is just over MTU
  - poor end to end performance: if loss of fragment
  - Path MTU discover protocol: uses ICMP error messages
  - engineering principle: make the comon case fast



Mobility

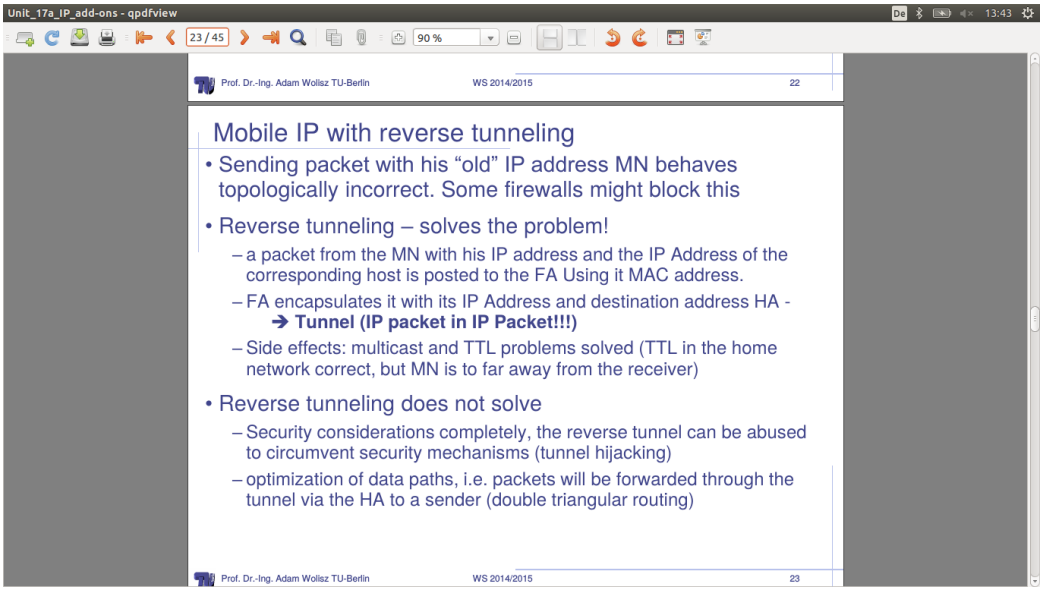
- Mobile IP
  - Routing is based on IP destination address, network prefix, which determines physical subnet
  - change of physical subnet, implies change of IP, to have topological correctness
  - specific routes to end- system: not scalable for destination changes and large number of customers (interfers with routing tables)
  - changing the IP? - imposibility to find mobile system, security issues
- requirment to mobile IP
  - Transperancy
  - compatibility: use same layer 2 protocols as IP
  - Security: authenification of all registration messages
  - efficiency and scalability (link is usually over low bandwith radio link)
- Terminology - notice Tunnel to change IP via foreign agent
- data transfer to the mobile system
- mobile IP with reverse tunneling

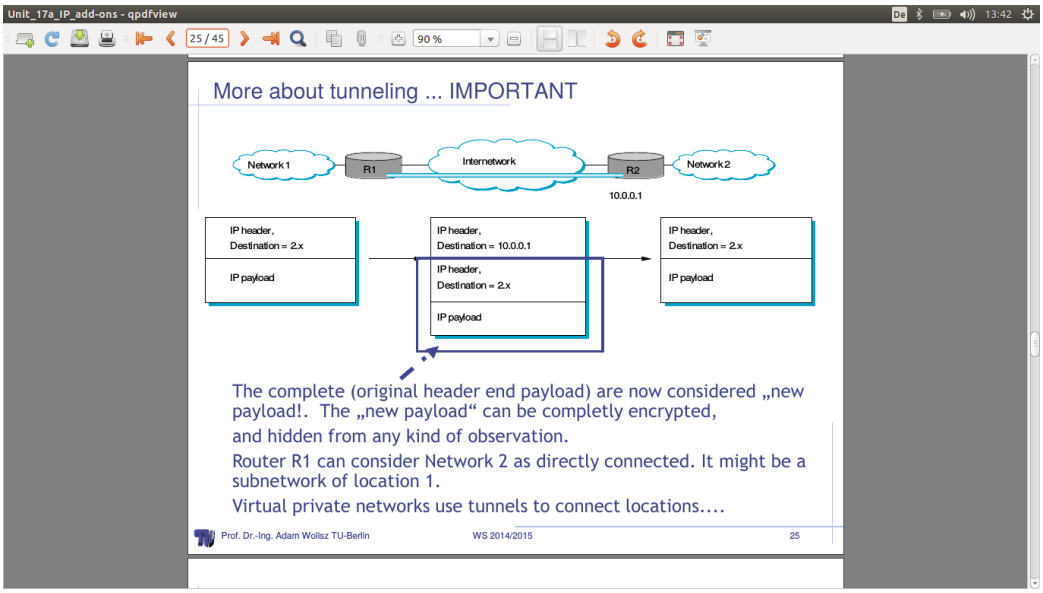

Unit_17a_IP_add-ons - qpdfview
19 / 45
100 %
Terminology
• Mobile Node (MN)
– system (node) that can change the point of connection to the network without changing its IP address
• Home Agent (HA)
– system in the home network of the MN, typically a router
– registers MN´s location, tunnels IP datagrams to the COA
• Foreign Agent (FA)
– system in the foreign network of the MN, typically a router
– forwards the tunneled datagrams to the MN, typically also the default router for the MN
• Care-of Address (COA)
– address of the current tunnel end-point for the MN (at FA or MN)
– actual location of the MN from an IP point of view
– can be chosen, e.g., via DHCP
• Correspondent Node (CN)
– communication partner
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
19


Unit_17a_IP_add-ons - qpdfview
20 / 45
90 %
Data Transfer to the mobile system
COA:Careof Address
home network
router HA
Tunnel
router FA
MN
foreign network
Internet
CN
router
home network
router HA
2.
router FA
3.
4.
MN
foreign network
Internet
1.
CN
router
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
20
How does it work…

  - sending packet with its old IP behaves topologically correct although a few firewalls might block this
  - reverse Tunneling: IP packet in IP packet solves this problem

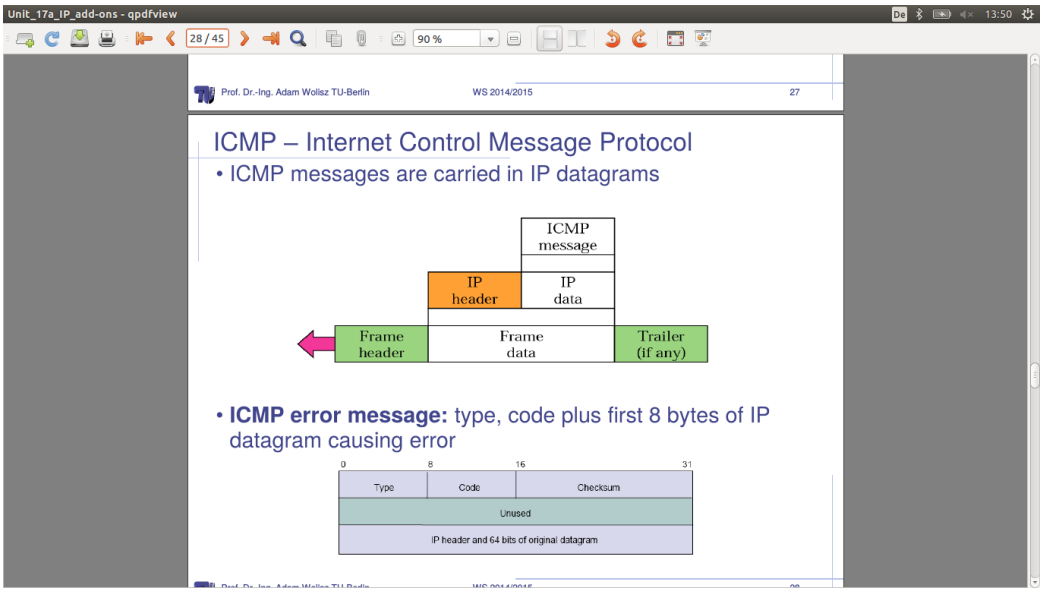
- check carefully following slides about tunneling: TOBE checked again





ICMP

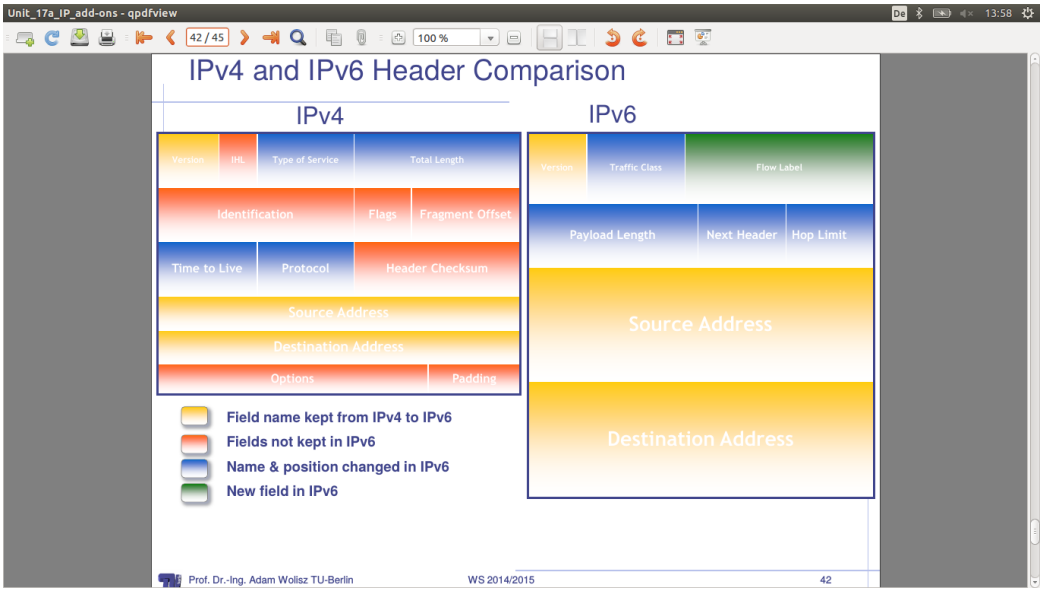
- ICMP is in the network layer BUT ABOVE IP
- ICMP is used by hosts, routers, gateways to communicate network level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping) PING
    1. pings are handled directly by the kernel
    2. Ping is translated in ICMP echo request
    3. ping'ed host responds with an ICMP echo reply
    4. executed three times: you get; answer and time of the Ping delivered to judge connectivity
  - other functions like: reachability testing, congestion control, route change information, performance measuring, subnet addressing
- ICMP packet topology



- Traceroute - in order to show routing of a packet: stays the same for roughly 20 minutes: good for system adminsistration

IPv6

- major features
  - 128 bit address
  - auto configuration
  - multicast
  - better QoS support
  - better authentification and security
  - end-to-end fragmentation
  - enhanced routing functionality, including support for mobile hosts
- IPv6 - IPv4 - header comparision



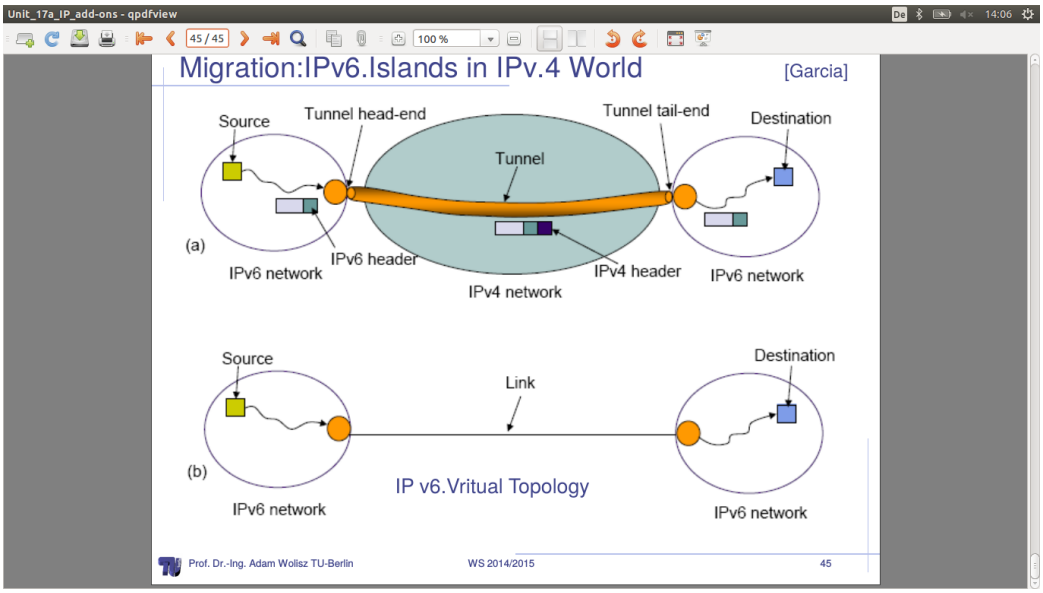
- Philosophy of changes
  - leave all problems to the ends
    1. elimination of fragmentation on routers
    2. eliminate chechsum
    3. but leave TTL for ICMP - traceroute
  - simplify handling
    1. new option mechanism: NEXT HEADER approach

2. eliminate Header length: no padding in header
3. eliminate checksum: failure occurs very very little: isn't effecient to provide checksum for analysis

– provide general flow label for packet

1. not tied to semantics(inhalt)
2. provides great flexibility

- migration of IPv6 island in an IPv4 World via Tunneling through IPv4 from IPv6 to IPv6



1.18 Routing Algorithms

- Routing Theory
  - finding the way from a given source to a given address
  - in multiple Layers (remember: learning bridges, network Layer, Application Layer)
  - graph problem: nodes are Components, verges are links
  - **Forwarding:** processing of a packet in a node assuming the routing information is known
- Source Routing vs. Hop by Hop Routing

  - source routing: source node defines the whole route -> Features
    1. no state information is needed in the individual node
    2. link breakdowns causes glibal consequences
  - hop by hop routing: each node guides via tables to through the net

- Hot potato (deflection) routing: vs. Target routing
  - target routing
    1. tables with useful information used for forwarding
    2. datagram case: there has always been the prot defined for each destination address
    3. using tables: **forwarding** vs. preparing Tables: **routing**
  - hot potato routing
    1. assume a packet arrives and no entry for the destination in tables
    2. -> forward somewhere at any Port: (same strategy is used for full queue)
    3. intuition: if I don't know maybe someone else knows
    4. used in optical networks

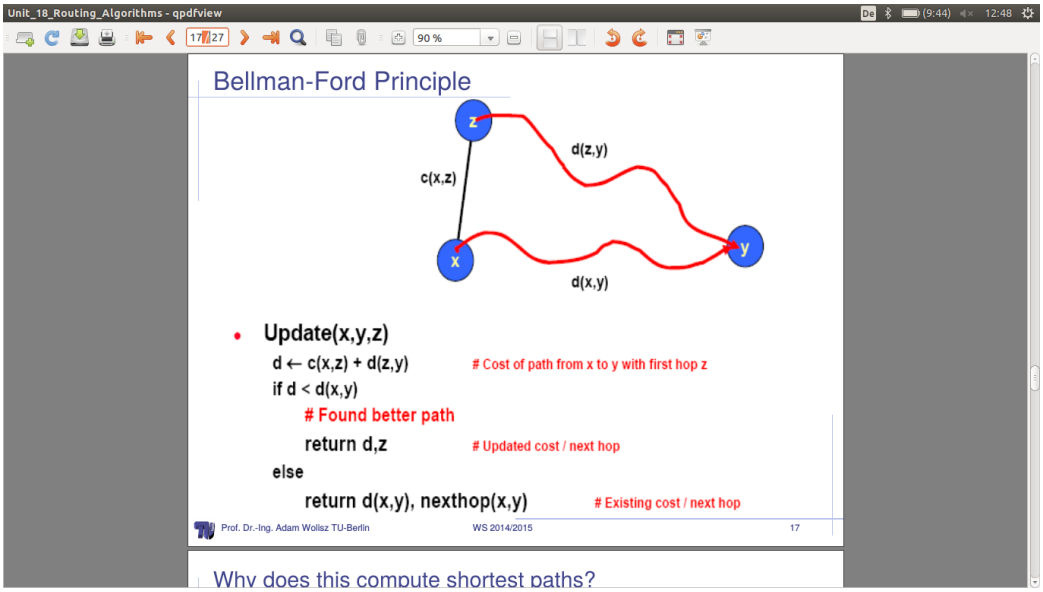
- Flooding
  - remember: filled out at previous chapters
  - flooding is used to construct a tree rooted at A:
    1. each nodes sends packets to its neighboors
    2. all nodes should mark the transmitter of each packet they receive as their parent on the tree
    3. nodes should relay packets to their neighbours only once - subsequent receptions of the same packet are ignored
    4. $\rightarrow$ source routing of tables important

- beyond trees: source routing is quite inefficient considering an increasing hierachy

- classes of routing algorithms
  - centralized (not really scalable, problem of central provider fails)
    1. collect one graph structure in one place
    2. use standard graph algorithm
    3. disseminate routing tables

  - partially distributed
    1. every node collects complet graph structure
    2. each locally computes shortest path from it
    3. each generates own routing tables
    4. Link State Algorithm
  - fully distributed
    1. no one has a copy of the entire graph
    2. nodes construct their own tables iteratively
    3. each sends information about its table to its neighbors
    4. distance routing

- The three solution option: Between Graph Theory and Computer Networking
  - Graph theory: Computer Science
  - Bellman-Ford: distance routing
  - Dijkstra: Link-state
- Dijkstra Algorithm
  - every node knows the graph: -> all link weights are >= 0
  - goal at node 1: find the shortest path from node one to all the other nodes
  - each node computes the same shortest path so they all agree on the routes
- Bellman-Ford-Algorithm (pic)
- Why does this computes the shortest path
  - suppose in each tick each node sends its distance vector
  - assume that initial distances are ∞
  - at time h, node i has an estimate of the shortest path to node j that has <= h+1 hops
  - $$D^{h+1}(i,j) = min\left(D^h(k) + c(i,k)\right)$$
- asynchronous Bellman Ford



  - in general nodes are using different an usually inconsistent estimates
  - if no link changes: the algorithm will converge to shortest(depends on cost function (time, distance, hops)) path
  - no synchronisation required at all

- comparision
  - Bellman-Ford
    1. calculation for node n involves knowledge of link cost to all neighbor nodes plus total cost to each neighbor from s
    2. each node can maintain set of costs and paths for every other node
    3. can exchange information with direct neighbors
    4. can update costs and paths based on information from neighbors and knowledge of link costs
  - Dijkstra
    * each node needs complete topology
    * must know all link costs of all link in the network
    * must exchange information with all other nodes
  - Dijkstra is robust since each nodes computates its own route independently

1. suffers from weaknesses of the topology update protocol: inconsitence etc
2. excellent choice: for a well engineered network within one administrative domain

– Bellman-Ford works well when the network is large, since it requires no synchronisation and has a trivial topology update algorithm

1. suffers from convergence delays
2. very simple to computate at each node
3. excellent choice for large networks

1.19 Global Internet

- hierachy of internet service Providers (ISPs)
- How are ISPs connected: Peering and Transit
  - Peering: business relationship whereby ISPs reziprocally provide to each other connectivity to each others customers
  - Transit: the business relationship, whereby ISPs provides usually sells access to all destinations in its routing table
- hierachical routing:
  - the internet has many administative domains
  - e.g. border routers (BGP)
  - **Interdomain (larger: e.g. BGP) & Intradomain (Germany, France,...)**
- natural way to scale routing
  - size, network or governance
  - allows multiple metrics at different levels of the hierachy
  - exploits address aggregation and allocation
- internet is organized at a two level hierachy: some Intradomain routing create their own hierachy: (see OSPF (link state routing protocol))
- autonomous System (AS's) - AS region of networks which run under a single adminitration domain

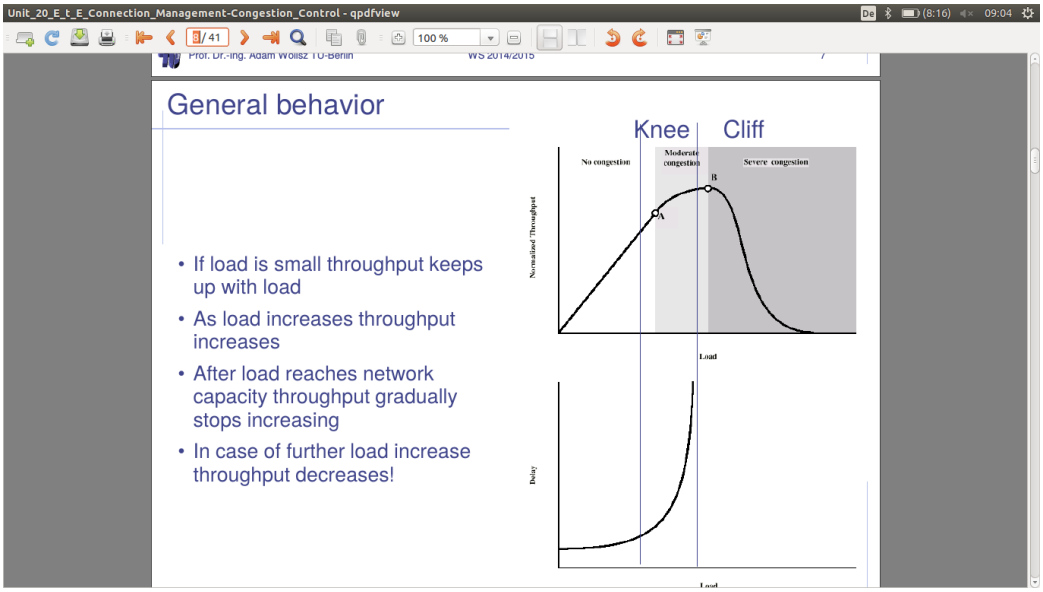
- AS's run an intradomain routing protocol
- be aware of term: Link State('FSM - whats possible'), Distance Vector (dijkstra approach), Path Vector('dont like B')
- Routing sub-functions
  - Topology Update: charakterize and maintain connectivity (discover neigbors, measure distance, disseminate)
  - Route Computation: Kind of path, multicast, unicast, centralized or distributed algorithm, policy, hierachy
  - route information distribution: if not computed locally in each relevant router
- OSPF - topology update via flooding
- OSPF - link state advertisement via packets (Link State Packets: LSP)
- OSPF - topology update - processing
  - entries in routing table with specific sequence numbering
  - update: periodically - process entries with actual data
  - keep date up to date while forwarding to make routing more efficient
- Issues with OSPF updating
  - what if some routers are much faster at transmitting LSPs than others
  - sequence number wrap
  - partitioned web is re constituted
  - security?
- BGP connects ASs
- Hierachical Addressing helps in Routing Structuring
- Forwarding table is configured by inter and intra AS routing algorithm
- Advertising a route means readiness to carry traffic you can reach A via me!
- Border routers have to communicate constantly: routers of the same AS speak IGBP and from different ASs speak EGBP

- Border gateway Protocol
  - obtain subnet reachability information from neighboring ASs
  - propagate reachability information to all routers in the AS
  - determine good routes to subnet based on reachability information and routing policy
  - allows subnet to advertise its existence to rest of the internet: 'I am here'
- BGP peers exchange routing information over semipermanent TCP connection: called BGP session
- note: BGP sessions do not correspond to a physical links
- BGP is a path vector protocol with **extra** Information
- prefix + attributes = 'ROUTE'
- when gateway router receives advertisment uses import policy to accept/-decline (decision makin, avoiding loops)
- The philosophy: Reachability
  - interdomain routing is about implementing policies of reachability
  - ISPs could be competitors and do not want share internal network statistics such as load and topology
  - routers have to select which one is the best route
- Why different Intra- and Inter AS-routing
  - Policy
    1. Inter AS admin wants to control over how its traffic is routed, who routes through its net
       - local preference value attributes: policy decisions
       - shortest AS path: minimal AS along the way
       - Best MED: multiexit discriminator: announced prefered entry router
       - closes NEXT-Hop router: hot potato routing: 'Out of my AS'
       - IP address of Peer-Router
    2. Intra-AS: single admin so no policy decisions necessary
  - scale: hierachical routing eases routing tables

  - Performance:
    1. intra-AS: can focus on performance
    2. inter-AS: policy may dominate over performance

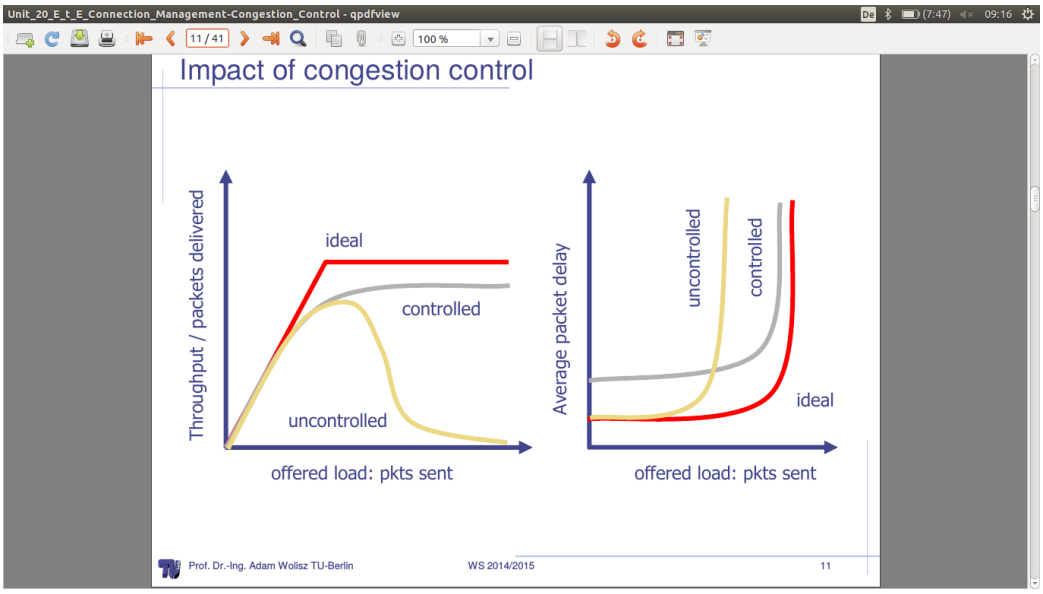
1.20 connection Management and Congestion Control

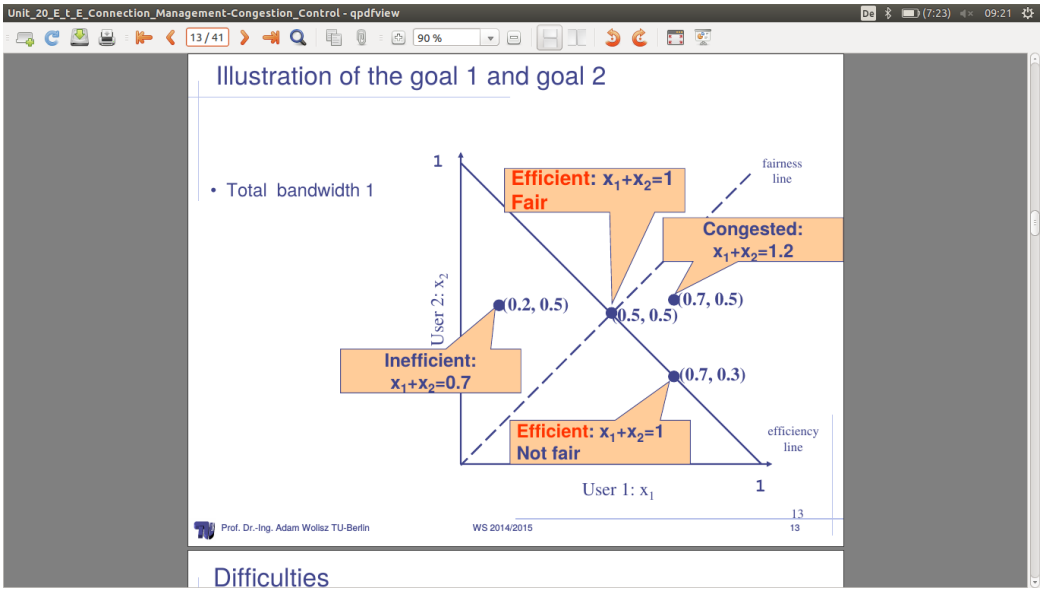
- Overview
  - Congestion Control
  - Connection: - Management, - Establishment, - Releasae
  - Timer based Connection Establishment: - Timeout estimation
- **Congestion Control**
  - informally: to many sources sending data to fast for network to handle → Demands exceeds Capacity
  - manifestation of congestion: - lost Packets (buffer overflow at routers(->Flow control)), long delays: queueing in router buffers)
- **Problems solving congestion:** Source are unaware of current state of the ressources and unaware of each other. A retransmission timer based source might retransmit extensivly delayed packets so generates even more Congestion.
- Flow control is used to insure that a source does not inundate a destination
- Congestion Control is used to insure that the network as a whole is not asked to carry more packets than it can handle (under givern flow distributions)
- Congestion Collapse Def: Increase of network load results in decrease of useful work done
- General behavior of congestion in network in dependance of Load refering to Throughput (see pic)
- end host can prevent congestion if the source adjust amount of data to put in the network according to a detected congestion
- routers can help avoiding congestion by:



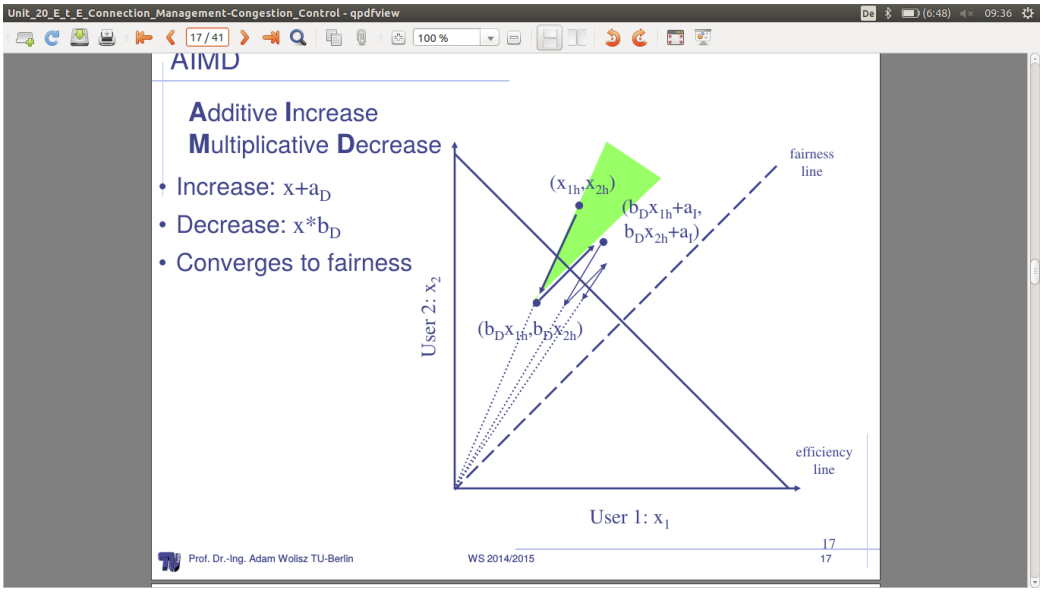
  - sending accurate congestion signals
  - isolating well behaved sources from ill-behaved sources
  - deciding which packet to drop
  - re-routing flows (shuffle flows to less congested links)

- congestion control and avoidance (mechanisms to use networks ressources efficiently)
  - avoidance: keeps the system running at the 'knee'
  - control: attempts to keep it left from the cliff
- impact on congestion control (pic)
- Goal fo congestion control and congestion avoidance
  - efficiency: utilize available bandwith to optimum (use the goal bandwith)
  - Fairness: equal access to bandwith for all hosts
  - Convergence: constant load => single solution for sharing/using Bandwith
  - Distributed Implementation: handle without a centralized decission maker
- goals decide on which you prefer


Unit_20_E_t_E_Connection_Management-Congestion_Control - qpdfview
11 / 41
100 %
Impact of congestion control
Throughput / packets delivered
ideal
controlled
uncontrolled
offered load: pkts sent
Average packet delay
uncontrolled
controlled
ideal
offered load: pkts sent
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
11


Unit_20_E_t_E_Connection_Management-Congestion_Control - qpdfview
13 / 41
90 %
Illustration of the goal 1 and goal 2
• Total bandwidth 1
1
Efficient: x1+x2=1
Fair
fairness
line
Congested:
x1+x2=1.2
User 2: x2
(0.2, 0.5)
(0.5, 0.5)
(0.7, 0.5)
Inefficient:
x1+x2=0.7
(0.7, 0.3)
Efficient: x1+x2=1
Not fair
efficiency
line
User 1: x1
1
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
13
13
Difficulties

- Difficulties
  - where is the knee? increase till congestion then decrease
  - stay at the knee: assuming you have a rough idea of bandwidth (put packet 2 in if packet 1 has left)
  - adjust total sending rate to match bandwith changes (increase till congestion and the decrease)
  - share the capacity among flows
- recognize Congestion: explicit network signal or implicit network signal(-> easier to distribute in machine: lost packets, long round trip delay)
- Approaches towards congestion control
  - End-to-End congestion control
    1. no explicit feedback from network
    2. observed by the end hosts
    3. e.g. in TCP
  - network assisted congestion Control
    1. intermediate system provides feedback to end system (via threshold system)
    2. choke packet router to sender
    3. indicate congestion via single bit solution
    4. explicit rate sender should send it
- AIMD - Additive Increase Multiplicative Decrease (pic)
- limiting sending rate: rate based & window based (packets allowed in further allowed in the network: useful in connection oriented versions)
- Isarithmic Congestion Control
  - token pool principle
    1. limit the total number of packets in a packet switched network
    2. packet must capture and destroy permit before entering the network
    3. permit is regenerated if packet leaves the network
    4. total number of packets in network will never exceed the number of permits initially present in the network
  - difficulties
    1. how to reach equal distribution of permits in the network



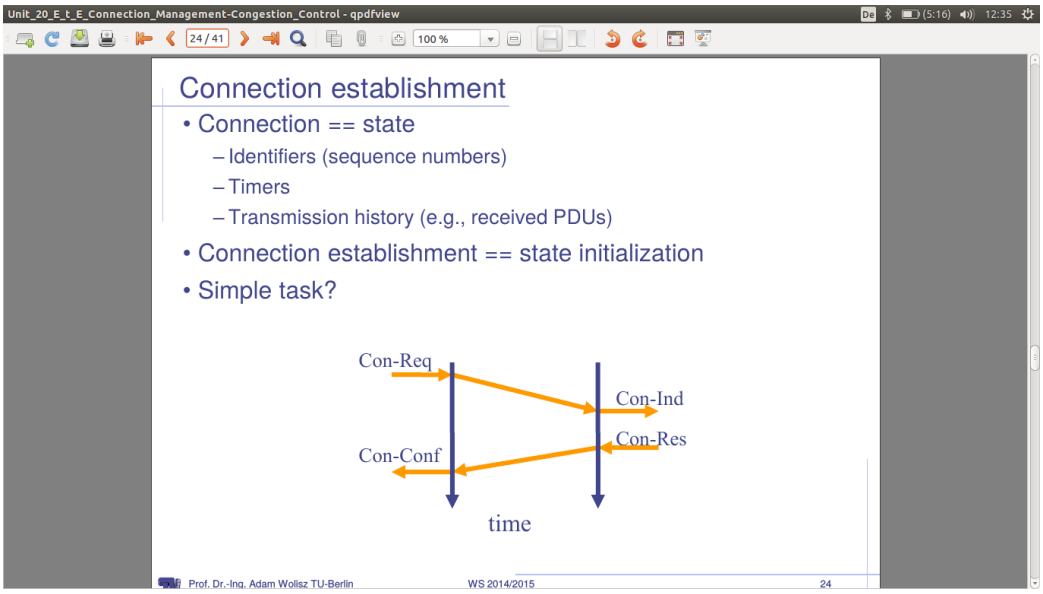
2. how to rapidly obtain permit (permit starvation)
3. how to recover destroyed permits (how many permits are in the network?)
4. overload congestion is possible

– reasonably inefficient in networks without flow control

- other approaches to congestion control
  - Packet discarding:
    1. simply discard excess packet and transfer the task to higher protocols
    2. standard datagram transmission
  - flow based routing approach
- **End-to-End Connection Management**
- Transport Layer: Connections - reliable data handling provided irrespective of the reliability of underlaying subnetworks
- connections: association between peer entities: point to point connection: two peers
- connection management: distributed protocol for
  - state management: (error control, flow control)

  - the three phases:
    1. Initialization (at both ends)
    2. state evolution (during transfer)
    3. Terminnation (reset) or state information (when done)

- connection establishment: connection == state : identifier(=sequenznumbers), Timers, transmission history (received PDUs)
- connection estabishment== state initialisation



- two way handshake only works in special cases (e.g. reliable network layer without corruptions)
- Complications with unreliable service (loss, duplication, delays, corruptions)
- solution to the complications
  - unique PDU/packet identifiers
    1. limited lifetime
    2. large enough sequencenumber space depends on transmission rate and PDU/packet lifetimes
    3. what about crash and re-start of partner: → continously growing seqnum
    4. three-way handshake

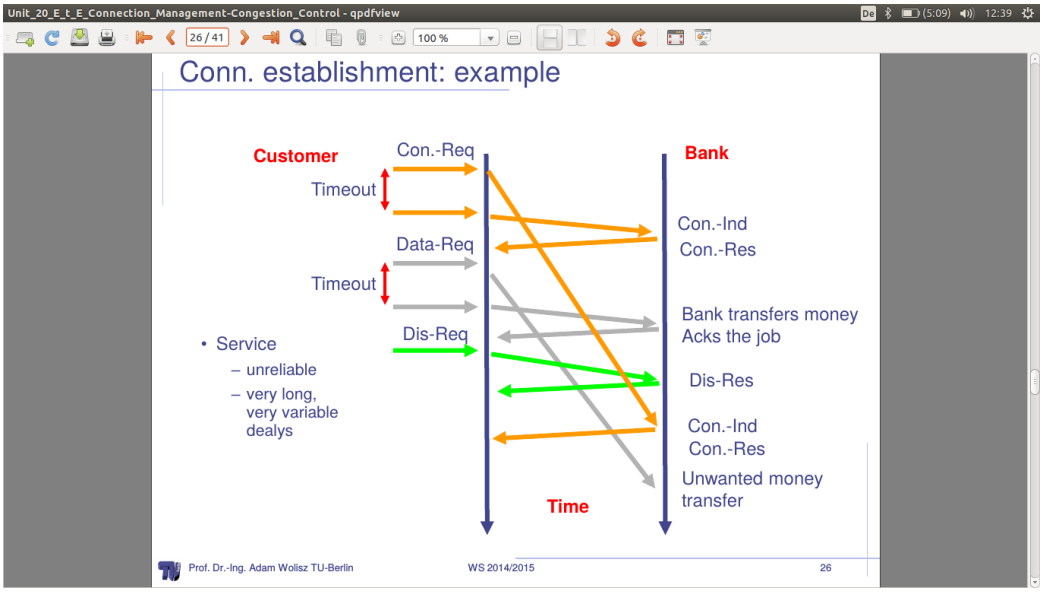
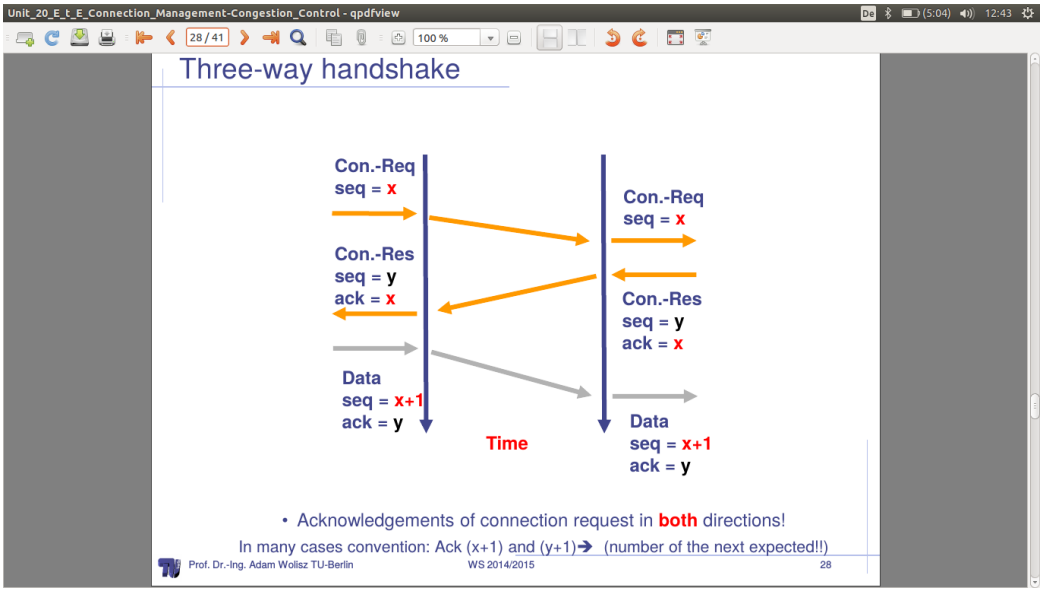


Bild 1.2: connection establishment: example

  - timers

- the Threeway handshake

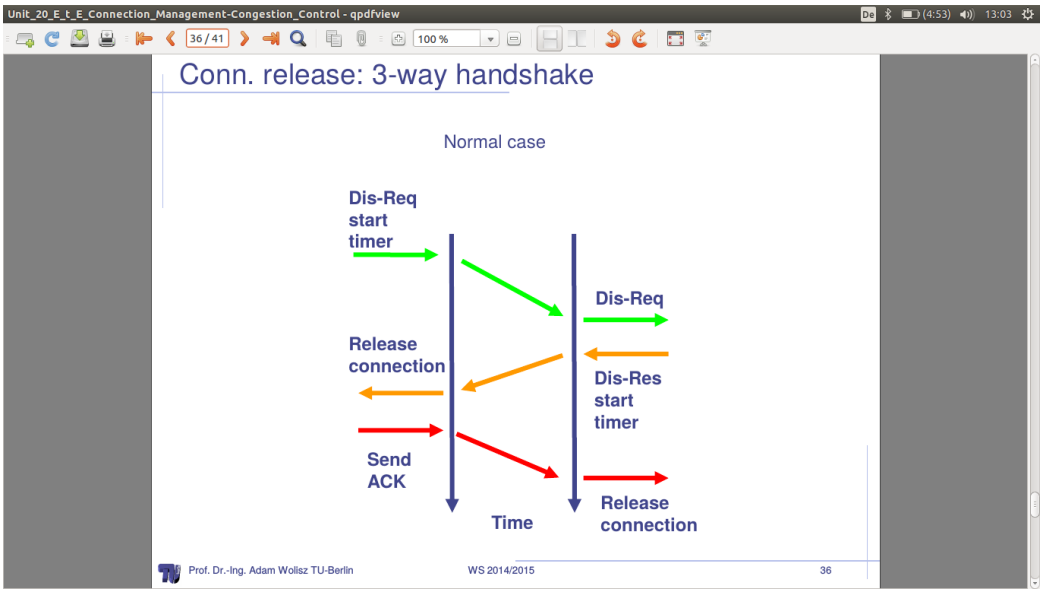


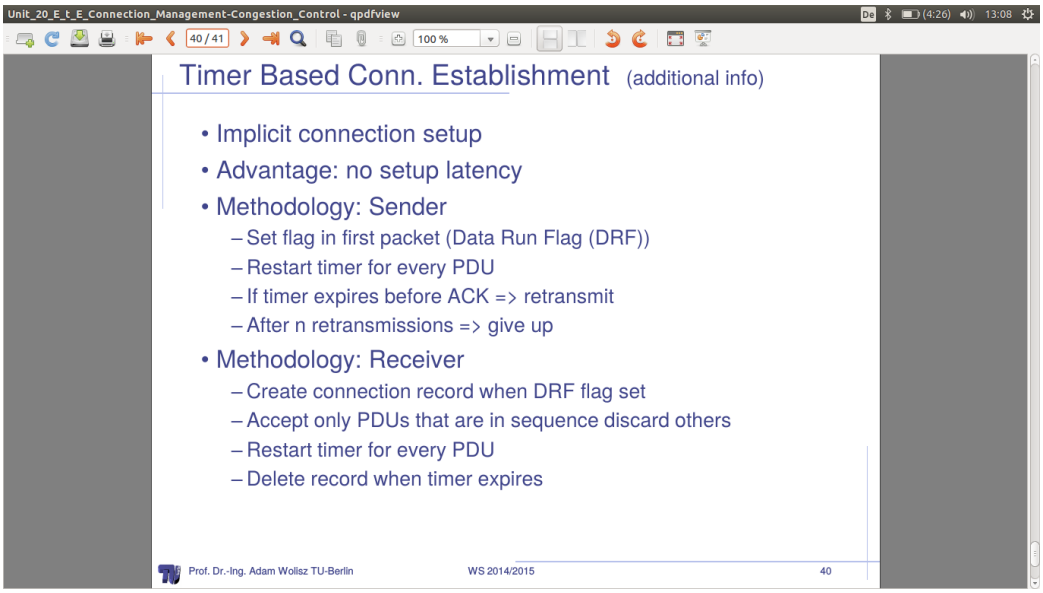
- How to chose initial seqnum?

  - works when initiator is stable (no reboots)

  - what about system crashes?
    1. defer new connections until PDU lifetime has expired => can cause long delays
    2. use explicit connection identfiers: two components ([initiator][responder])
    3. determine initial seqnum via timer!
  - Problems
    1. long-lived slow sessions (longer than wraparound time): 1. Forbidden region begins to early 2. same sequence number begins within T 3. enter curves from above
    2. high data rate: 1. runs out of sequence numbers 2. enter curve from below
  - solutions
    1. PAWS: protoect against wrapped sequence numbers
    2. TCP extensions for high speed data paths

- connection establishment summary
  - problem delayed duplicates
  - solution conditions
    1. no connection + late Con-Req => no connection initialization
    2. connection exists + PDU from closed connection => PDU should be rejected
  - solution see above:
- connection release: three way handshake
- the Threeway handshake for connection release
- if final ACK lost: dis-res Start timer: timer expiration => release connection
- for the other error cases: check unit 20 p. 36 ff
- timer-based connection establishment (additional info: see slide)

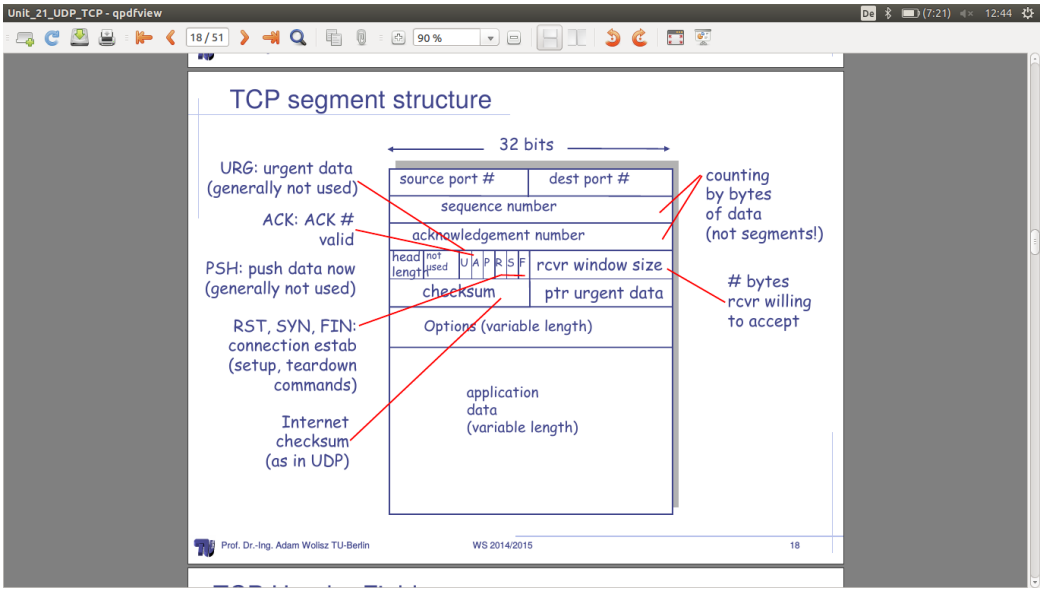

Unit_20_E_t_E_Connection_Management-Congestion_Control - qpdfview
36 / 41
100 %
Conn. release: 3-way handshake
Normal case
Dis-Req
start
timer
Release
connection
Send
ACK
Time
Dis-Req
Dis-Res
start
timer
Release
connection
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
36


Unit_20_E_t_E_Connection_Management-Congestion_Control - qpdfview
40 / 41
100 %
Timer Based Conn. Establishment (additional info)
• Implicit connection setup
• Advantage: no setup latency
• Methodology: Sender
– Set flag in first packet (Data Run Flag (DRF))
– Restart timer for every PDU
– If timer expires before ACK => retransmit
– After n retransmissions => give up
• Methodology: Receiver
– Create connection record when DRF flag set
– Accept only PDUs that are in sequence discard others
– Restart timer for every PDU
– Delete record when timer expires
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
40

1.21 TCP and UDP

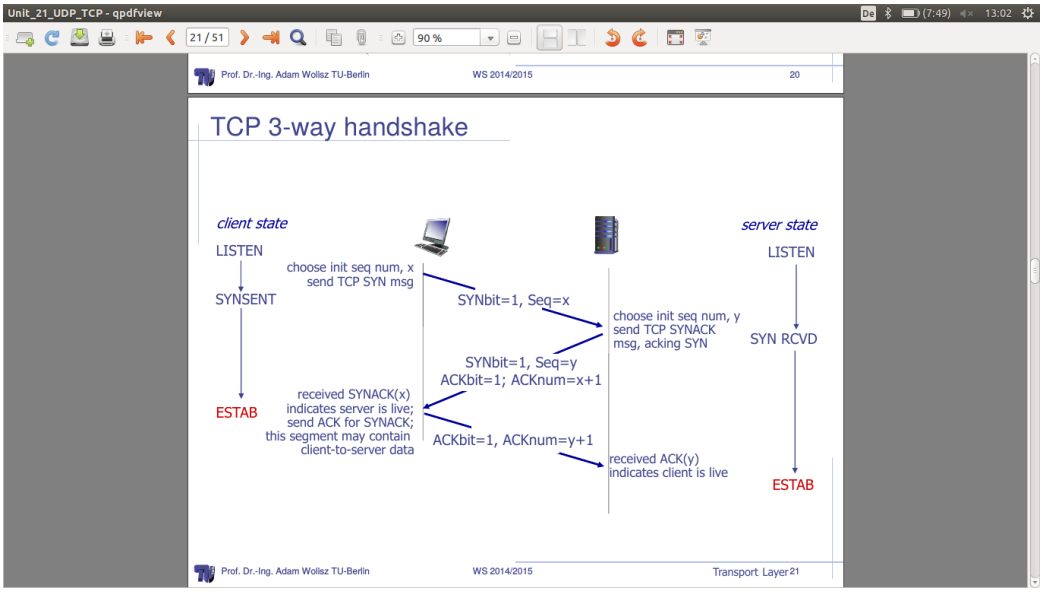
- network layer: IP
  - connection less: packets may be lost, duplicated or be received out of order, Variable Delay through media
- Transport layer
  - UDP: connectionless
  - TCP: connection orientated, reliable
- Addressing of Applications
  - IP is used to address an interface (host)
  - Protocol Identifier of IP header is used to select receiving protocol
  - Ports are used to select the communication end point (application)
- UDP -specs
  - end-to-end checksum (optional)
  - total length field (redundant: since IP has a length field, too)
  - each user request is transfered using a single datagram
  - UDP provides no send buffer but a receive buffer
- UDP checksum
  - ones complement of 16 Bit words (as IP)
  - covers header and data plus a 12 byte pseudo header
  - pad bytes in case of an odd packet length (not transmitted)
  - receiver has to verify checksum
- **TCP**
- connection oriented vs. connectionless
  - connection oriented: user requests a reliable service, in order, no dups etc. Data streams - thins of the conn-oriented socket interface
  - connectionless: requests handled independently, unreliable transmission, order of delivery not sure, duplications

- conn. oriented Service: in packet switched networks connection oriented service can be provided above the virtual ciruit switching
- Forwarding in VCs (Virtual Circuits) is much easier through definitive routing tables
- connection oriented service can also be offered on top of datagram switching
  - connection establishment via datagram handshake
  - loss free-operation, in-sequence packet delivery will be provided by end-to-end mechanism between entities establishing the connection
  - TCP over IP
- Basic TCP Operation
  - at sender
    1. application data is broken into TCP segments
    2. TCP uses a timer while waiting for an ACK of every packet
    3. un-ack'd packets are retransmitted
  - at receiver
    1. errors are detected using a checksum
    2. correctly received data is acknowledged
    3. segments are reassembled into their proper order
    4. duplicate segments are discarded
- window based retransmission and flow control + congestion control
- TCP segment structure
- TCP header fields
  - Port numbers: like for UDP
  - 32 bit SN is in bytes, defines the first byte of data
  - 32 bit RN is used for piggybacking ack (implicit ack for all the bytes up to that point)
  - data offset is a header length in 32 bit words (minimum 20 bytes)
  - window size used for error recovery (ARQ) and as a flow control mechanism Sender cannot have more than one window of packets in the network simultanously



  - specified in bytes: window scaling used to increase the window size in highspeed networks
  - checksum covers the header and data

- Sequence numbers in TCP
  - tcp regards data as a byte stream (each byte in byte stream is numbered)
  - 32 bit value wraps around (initial values selected at start up time)
  - TCP breaks up byte streams in packets: packet size is limited to Maximum Segment Size (MSS)
  - each packet has a sequence number: seq no of 1st byte indicates where it fits into the byte stream
  - TCP connection is duplex: data in each direction has its own sequence numbers
- TCP threeway handshake: are you alive, yes: are you?, yes: i have request (like a polite phone call)
- TCP connection management
  - Step 1: client end system sends TCP syn control segment to server
    1. specifies initial seq#
    2. specifies initial window#



  - Step 2: server end system receives SYN, replies with SYNACK control segment
    1. ACKs received SYN
    2. allocates buffer
    3. specifies server: receiver initial seq no.
    4. specifies initial window
  - Step 3: clients system receives SYNACK

- TCP closing a connection
  - client, server EACH side closes their connection seperately : send TCP fin bit = 1
  - respond to received FIN with ACK
  - simultanous fin exchange can be handled

- Error Control: a Variation of Go-Back-N
  - sliding window with commulative ACKs
  - last one is highest and acks all lower ones: in error case starting point of retransmission
  - duplicate acks are sent when out of order receive
  - sender only retransmit packet at a time: the only (common case, hopefully) packet is retransmitted to avoid congestion

  - error control is based on byte sequences: retransmitted packet can, due to fragmentation, be different from the original lost packet

- TCP Sender events
  - Data received from application:
    1. create segment with seq. no.
    2. sequence number is a byte stream number of first data byte in segment
    3. start timer if not already running: Think of timer as for oldest UN-acked segment
    4. timer expiration interval: timeout
  - Timeout
    1. retransmit segment that caused timeout
    2. restart timer
  - ACK received
    1. if acknowledges previously unACKed segments
    2. update what its known to be ACKed
    3. start timer if there are outstanding segments
- delayed ACK(wait for next pending ack to reduce ack traffic), comulative ACK (ackes all previous ones), duplicate ACK(out-of-order)
- Fast Retransmit
  - when TCP receives an ACK with an SN which is greater than the expected SN, it sends an request packet with a request with a request number of the expected packet SN : this could be due to out of order delivery or packet loss:
  - if a packet is lost, then duplicates RNs will be sent by TCP until the packet is correctly received, but the packet will not be retransmitted until a timeout occurs: this leads to inefficiency
  - fast retransmit assumes that if 3 duplicate RNs are received by the sending side module that the packet was lost, so retransmit and then continue to send new data
  - TCP fast retransmission allows the protocol to behave more like ARQ-SR than ARQ-GbN
- SACK - option for selective ACKs also widely deployed:

  - selective ACK essentially adds a bit mask of packets received
  - when to retransmit; Packets may experience different delays, still need to deal with reordering, wait for out-of-order by three packets (3,4 ranges)

- TCP retransmission timeout
  - one timer per packet only
  - retransmission Timeout (RTO) calculated dynamically
  - based on roundtrip timedelay (RTT)
  - importance of accurate RTT estimators: too low rtt -> unneeded retransmission, too high rtt -> poor Throughput
  - rtt estimator must adapt to change in RTT

- TCP flow control - sliding window protocol
  - when data is acked window slides
  - receiver informs sender about dynamically changing buffer space: rcvr window size field in TCP segment
  - sender amount of transmitted unACKed data, less than most recently received

- Silly Window Problem:
  - sender opens window small amount
  - inefficient because most packet contain packet overhead
  - small segment size remains indefinitly: problem silly window syndrome
  - mechanism needed to wait for opportunity to send larger amount of data

- when to transmit: Nagle Algorithm:
  - waiting too long hurts interactive application
  - without waiting, risk of sending a bunch of tiny packets (silly window)
  - -> Nagles Algorithm
    1. continue to buffer data if some un-acked packets are still outstanding
    2. if no outstanding data, send packets without delay
    3. if more than MSS worth of data, send segment without delay

  - notice: TCP self clocking mechanism

- congestion control details: TCP sending rate send cwnd bytes, wait RTT for ACKs, then send more bytes
- TCP slow start: summary initial rate is slow but ramps up exponentially fast
- TCP detecting, reacting to losses
  - loss indicate by timeout
    1. cwnd set to 1 MSS
    2. window then grows exponentially as in slow start to threshold, then grows linearly
  - loss indicate by 3 duplicates: cut cwnd in half then grows linearly
  - TCP switching form slow start to congestion avoidance
- TCP congestion avoidance: additive increase (linear), multiplicative (cut in half) decrease
- TCP throughput: raised heaviside function: half window (half Througput) to full (full)
- multimedia apps often use TCP, because they do not want their rate sheduled by congestion control mechanism for UDP

1.22 Quality of Service

- Internet protocol stack: a review
  - application: supporting network application FTP,SMTP,STTP
  - transport: host data transfer: TCP, UDP
  - network: routing datagrams form source to destination: IP, routing protocols
  - link: data transfer between neighboring networking elements: PPP, Ethernet
  - physical: bits on wire
- Internet can do more than data: telehone network, internet, TV distribution network: Vision -> everything over the internet

- Time uncritical applications
  - the classic up/down load links (file, email)
  - requirement: error free delivery
  - the user is interest in time that takes to complete
  - classical TCP usage: What throughput is offered (payload[bit]/s)
- delay critical Application
  - real time control (hard real time)
    - requires error free communication with exact time limits
    - usually targeted by specialised protocol architecture/specially dimensioned networks
  - soft- Real time applications
    - intuitively: the required time delay can occasionally be violated **without** serious consequences
    - real time audio- Video services
    - interactive web access
- reminder: delay on the way: four instance (nodal processing, queuing, transmission delay, propagation delay)
- variable network delay: -> Jitter (problematic with voice over IP)
- neccessary receiver client buffering: play-out-delay compensates for network delay jitter
- Transformability of QoS - Parameters
  - error rate can be reduced at the expense of decreasing throughput, e.g. principle of forward error correction
  - error rate can be reduced at the expense of increasing delay (mean, variance), e.g. principle of ARQ
  - error rate can be reduced at the expense of increasing the data volume and (marginal) increase of delay (FEC)
  - delay jiiter can be reduced at the expense of increasing delay, e.g. play-out buffer
- Quality of Experience
  - real interest: user experience!!

  - how to measure? (computate mean afterwards)
  - network admin can only influence QoS

- Problem of the lot: FIFO switching: (reminder: small packet stuck behind big packet)
  - in order to maximes its chances of success a source has to maximize the rate at which it transmits

fairness - when many flows pass through it, a FIFO queue is unfair if it favourises to the most greedy flow

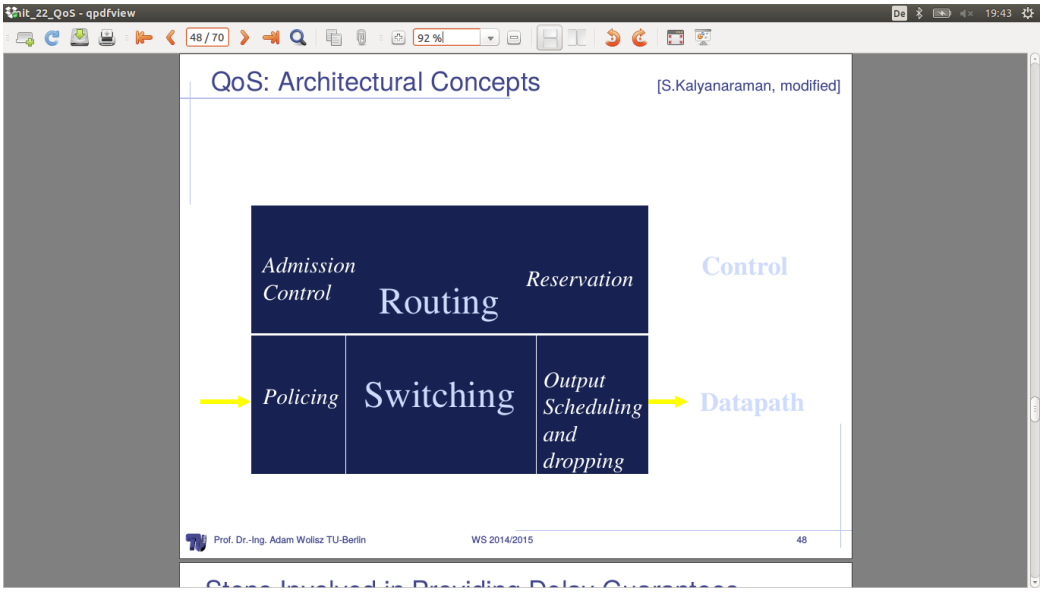
delay gurantees - it is hard to control the delay of packets through a network of FIFO queues

- wo what to do?
  - over provisioning: just have enoug ressource: low load factor, low delays
  - reservation of ressources: (what criteria? - consider admission contorl (busy phone signal))
  - prioritization, sheduling, policing
- Max-Min-Fairness: prioritize the one with the least demanded flow rate
- FIFO vs. Per Flow treatment in routers - classification and sheduling required
- first attemp on fairness: the round robin: (bit like token bucket, token access)
- fair queueing: classification first (sepereate flows) -> bit by bit processing (in round robin manner, but allows different packet lengths)
- weighted bit by bit fair queueing (servicing a different number of bits per flow at each round) also calles GPS: general processor sharing
- Packetized weighted fair queueing (WFQ)
  - we need to serve a whole packet at a time
  - computate time a packet if served will complete first (bit by bit)
  - call this finishing time: sort all packets in the order of increasing finishing time

- the use of WFQ for weighted fairness
  - different definitions are possible: prefer x because of y
  - long term excessive use cannot be served for any infinite time (it will collapse: SDA)
  - big temporary traffic leads to losses: cannot increase buffers endlessly because of delay increase
- some applications need bounds on packet delay
  - multimedia application (streaming)
  - real time control
  - other delay sensitive application (premium internet access)
- traffic shaping/ policing mechanism
  - goal: limit traffic in order not to exceed declared parameters
  - 1. long term average rate: how many packets can be sent per unit time
  - 2. Peak Rate: e.g. 4x times the avearage ppm peak reate
  - 3. burst size max. number of packets sent consecutively (with no intervening idle)
- Controlling Burstiness: Token Bucket
- Assuring a delay gurantee with WFQ
  - token bucket and WFQ combination to gurantee upper bound on delay, i.e. QoS gurantee!
  - the combination of Maximum Arrivals and Minimum service Rate makes it out
  - 'If flows are token bucket constrained and routers use WFQ, then end-to-end delay gurantees are possible'
- Providing delay gurantees
  - before starting transmission: source asks network for end-to-end delay gurantee
  - source negotiates values with each router along the way to achieve end-to-end delay gurantee: routers perform admission control to check whether they have sufficient resources

  - each router along path reserves resources (admission control)
  - flow starts, and source transmits packets at negotiated values (r,b)
  - router perform classification (ordering flows to admitted resources)
  - routers serves queues using WFQ, so as to bound packet delay through router

- architectural concepts of QoS



- Steps incvolved in providing delay gurantees
  - per session
    - call setup, call admission, resource reservation
  - per packet
    - packet classification (identify flows)
    - shaping (keeping my side of the contract)
    - policing (did user keep side of their contract)
    - packet shedudling (sending at the right time)
- QoS in the internet
  - integrated service architecture
    - following much the above steps

    - signalling using the RSVP protocol
    - having per flow state in each router
  - the differentiate service architecture
    - no per flow state
    - relative service distinction (Platinum, Gold, silver)

- everything is always soft state in the network
- IntSev Mechanism
  1. the flow
     - is QoS abstraction
     - each has stable or fixed path
     - routers along the way maintain state of the flow
     - state is used to deliver apropiate service
  2. reservation protocol transmits service request to network
  3. admission control: determines whether to accept or deny request
  4. packet sheduling: ensures router meets service regulations
  5. routing: pin routes, look for ressource-rich routes
- IntSev Service
  1. kind of service assurance:
     - guranteed: (never fails unless major failure
     - predictive (will almost never fail)
  2. corresponding admission contol
     - guranteed worst case - no guessing about traffic
     - predictive measurement-based - gamble on aggreed behaviour changing slowly
- Reservation Protocol: RSVP
  - sender sends PATH message via data delvery path: set up path state each router including the address of each previous hop
  - receiver sends RESV message on the reverse path
    - specifies reservation style: QoS desired (RSpec)
    - set up the reservation state at each router

  - things to notice: receiver initiated reservation; decoupling routing form reservation

- End-to-End Reservation
  - source s sends a message containing traffic characteristics
  - receiver r send back this information + worst case delay
  - each router along the way gurantees per hop delay and forwards

- Reservation styles: Filters
  - a **session** is a simplex data flow sent to a unicast or a multicast address, characterized by <IP dest, protocol number, port number>
  - Wildcard filter: all session share the same resources (good for small number of simultaneously active senders)
  - Fixed Filter: no sharing among senders, sender explicitly identiefied in reservation
  - Dynamic Filter: resource shared by senders which are explicitly speciefied

- Service classes: Scalability simple functions in network core, relatively complex functions at edge routers (or hosts) - like signalling, maintain per-flow router state, difficult for large number of flows

- do not define service classes: rather build functional components to build service classes

- a flow is not necessarily homogenous: might consist of more or less important information (video streaming)

- video encoding: mpeg -> prediction packet (image changes differentially, so reduce data to difference)

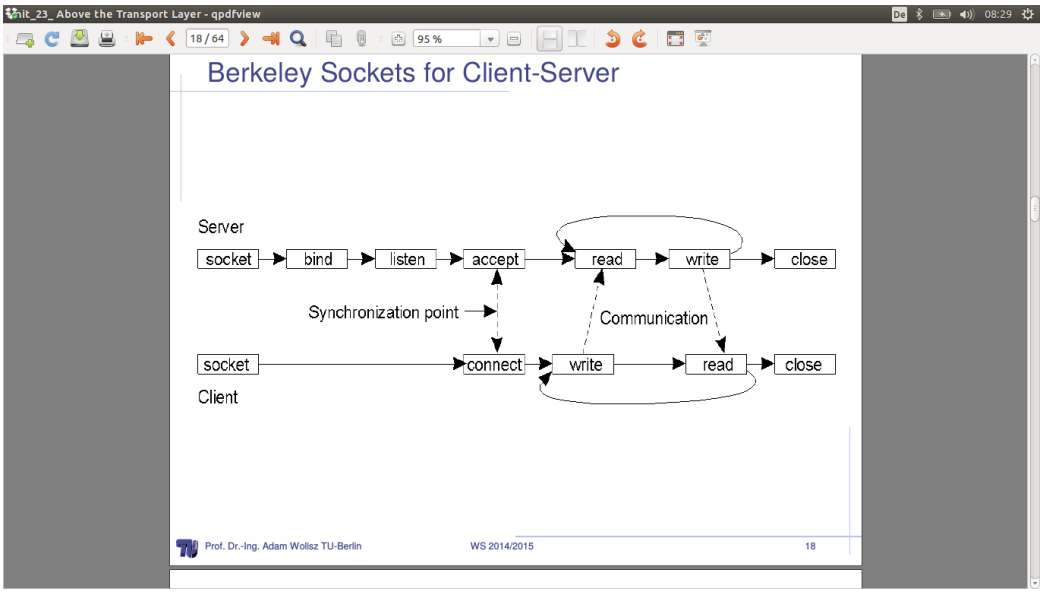
1.23 Above the Transport layer - Application/Session etc.

- remember internet hourglass model - IP is the waist: main target interoperability

- IP address: four part number used by the transport layer to route a packet

- Process Address
  - to receive a message, a process needs to a have an identifier
  - identifier includes: 1. IP address 2. port numbers associated with process at the host
  - e.g. http Port 80, mail server port 25
- creating a network app
  - write programms that run on different end-systems and communicate over network
  - no need to write software for network core devices:
    - network core devices do not run user application
    - application on end-systems allows for rapid app development propagation
- be aware of IEEE 754, Big endian, Little endian
- Taxanomy: data types: base, flat, complex ... conversion stragtegy (google translater: O(n), O(n^2))
- Abstract Syntax notation: ASN.1
  - each transmitted data value belongs to an associated data type
  - for the lower layers only a fixed set of data types is needed (frame formats)
  - ASN.1 distinguishes between data type (as set of the possible values of this type) and values of this type
  - Basic ideas of ASN.1
    - every data type has a globally unique identifier: data type
    - every data type is stored in a library with its name and a description of its structure
    - a value is transmitted with its type identifier and some additional information (e.g. length of a string)
- Definition of Datatypes using ASN.1
  - a data type definition is called abstract syntax:
  - lexical rules: e.g. case sensitive, type identifier starts with big letter,...
  - e.g. enumerated, real, bit string

- The client - server Approach
  - pc(client) - printer(server) example
  - client transmits a request message to the server (including the file to be printed)
  - the server receives message and (probably) will perform apropiate action
  - result a send back to the client via reply message
- Def. **Service:** any act or performance that one party can offer to another that is essantially intangible and does not result in the ownership of anything. Its production may or may not be tied to physical product
- focus is on output: the result of the service: »« not the means to achieve it
- server is always on host and needs constant IP (+ access to data centers for scalability)
- reminder Berkely sockets:



- P2P architecture - an alternative
  - not always on server

  - self scalability: new peers bring new service capacity, as well as new service demands
  - peers change IPs -> complex management

- Remote Procedure Calls (RPC)
  - are prefered tool to implement the client server approach
  - in classical procedure calls the code of the procedure is allocated at the same computer (same address space) as the calling program, in an RPC the code is located at another computer
  - goal of RPC: transparancy - the caller should not know if the callee is located locally or remotely
    1. things to be considered
    2. parameter handling and marshalling
    3. semantics
    4. addressing
  - an RPC system is attractive because automatic conversion from local to remote procedural call can be supported (see below)
- Marshalling - taking parameters/results of a procedure call and prepare them for transmission over network
  - to ensure transparancy between different hardware, operating systems, programming languages
  - handled by client stub and server stub/skeleton
- RPC - Parameter passing
  - procedures in common programming languages have different types of parameters and calling conventions, which have to be treated in RPC
  - simple call-by-value parameters are passed 'as is' (e.g. integer values)
  - call-by-reference parameters are pointers, since different address spaces are used by sender and receiver, the denoted value (e.g. a buffer) has to be completely transmitted, if the server changes values of that buffer it has to retransmitted in the answer
  - complex data types using pointers (e.g. graphs, trees or list) cannot or only difficultly be transmitted

- the stub procedures must use a common encoding convention for different parameter types
- Finding an RPC server (addressing)
  - can used hard coded, fixed address - this approach is simple but not flexible
  - a dynamic binding approach:
    - a server stub transmits init massage containing its name, its version number, its address and its unique identification to a special station; to the **bindery** station, which maintains a database of all available services
    - a client stub, if operating for the first time, queries the bindery station for an appropiate server providing the requested service (i.e. service name, version number). if no server exists, client stub fails. Otherwise the bindery returns the address and the unique identification to the client stub
- RPC- Semantics
  - normally RPC behaves like LPC and returns correctly
  - problems arise through: Addressing, the client or server fails, message loss
  - if client cannot find a server, exception handling is needed
  - the server can fail, before executing the request, while, or directly before transmitting results: undistinguishable for client stub
  - RPC are Idempotent (can be repeated without harm)
- **Idempotent operations**
  - doing it twice has the same effect as doing it once
  - doing it partially (several times possible) and then doing it whole has the same effect as doing it once
- do i have to memorize IP address
  - host names depict machines in organizations: more human readable than a silly number
- **DNS** - Domain Name System
  - distributed database: implemented in hierarchy of many name servers

  - application layer protocol host routers, name servers to communicate to resolve names (address/name translation)

- DNS - Features
  - hierachical Namespace: root - edu/mil/...
  - distributed architecture for storing names
  - Administration divided along the same hierachy
  - client server interaction on port 53
- root name servers: lower layers request mapping from above
- TLD and authoritative Servers
  - TLD responsible for classes like: (.edu .org .mil) and top level country domains
  - authoritative DNS Servers: organizations DNS, providing authorative host names for IP mapping (maintenance by organization or service provider)
  - Local Name Server: each ISP has one: local query
- DNS caching - (deep web)
- seperating Naming and Addressing
  - names are to be remembered
  - address can change underneath
  - name could map to multiple IP address
  - aliases: more names for same IP address
- DNS records
  - DNS: distributed database storing ressource records (RR)
- the www
  - content: a distributed database of URLs
  - client-server-principle:
    1. server which store files and execute remote commands
    2. client retrieves and displays pages of content linked by hypertext
  - the basic aspects

1. need a language to define the objects and the layout: HTML,XML
2. need the way to identify the resource - URL
3. need a protocol to transfer information between peers - HTTP

- HTML - hyper text markup language
  - web page components: html file, referenced object (pics) - like latex
  - browser interpret HTML
  - several functions: format text, reference images, embedded hyperlinks (HREF)
- URL - uniform record locator
  - extend the idea of hierachical name spaces to include anything in a file system
  - extend to programm execution as well - server side processing can be incorporated in the name
- Example: how does HTTP work
  1. after finding out the IP address of the host (DNS)
  2. http client initiates a TCP connection on :80
  3. client sends the get request via socket: established in 2.
  4. server sends the html file, which is encapsulated in its response
  5. http server tells tcp to terminate connection
  6. http client receives the file and the browser parses it
  7. client repeates steps 2-5
- internal organization of HTTP:
  - persistent: server leaves connection open: expected reuse -> minimise overhead for connection set-up
  - pipelining/ no pipelining
- Server side caching
  - done by ISP
  - done to decrease server loads
  - avoid transit costs

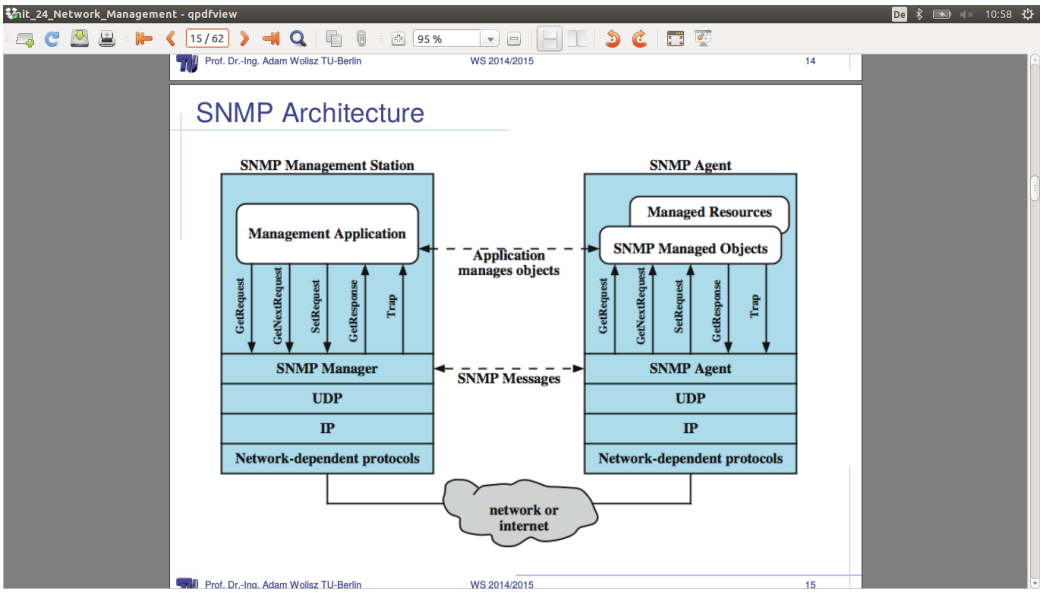
- forward caching: done by ISP and corporate LANs
- HTTP is stateless protocol: each request is handled seperately
  - good for scalability
  - bad for some apps which need persistent state
- state in stateless protocol: –> Cookies
- fate sharing: idea: when storing states in a distributed system, keep it collocated with the entities that ultimately reply on the state: failure only if state loss and then entities which cares about also fails: ... in which case it is irrelevant

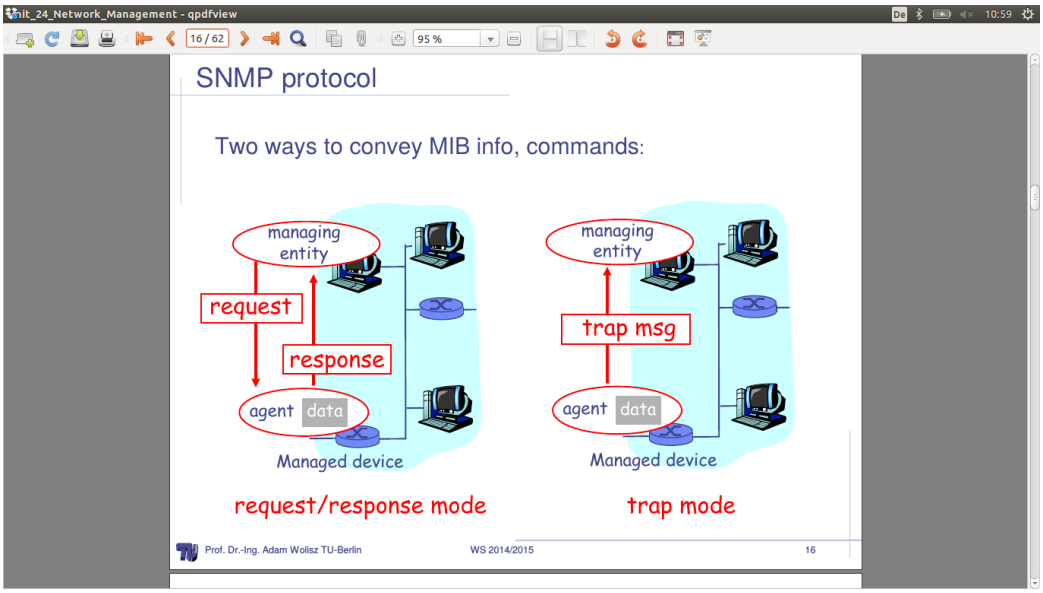
1.24 Network Management

- Def: **network management:** Network management includes the deployment, integration, coordination of the hardware, human elements to monitor, test, poll, configure, analyze, evalutate and control the network and element resources to meet the real-time, operational performance and quality of service requirements at a reasonable cost
- Network Importance & Requirements
  - networks are becoming indispensible, larger and more complex
  - challenges in network management
    - fault management
    - configuration managment
    - performance managment
    - sercurity managment
    - accounting managment
  - requires automatic network managment tools
- Network Management Infrastructure
  - tools supporting network management
  - requirements
    1. operator interface
    2. userfriendly command set

3. minimal amount of seperate equipment
4. view entire network as unified architecture
5. active elements provide regular feedback

- Components of Network Management Infrastructure
  - managing Entitiy (Manager) - application running in control center for managing a network
  - managed device - network equipment, Management information base (MIB)
  - management Protocol - for communication between ME and MD (above two)
- SNMP - Simple Network Management Protocol (Internet Network Management)
  - management is done from the management station (manager)
  - it communicates via SNMP protocol with agents
  - information from a node not being able to run an agent can be retrieved from a proxy agent running on another node
  - biggest part of snmp describes the kind of information that a specified type of agents provides and the **format** of it
  - MIB central for all nodes : communicated information in ASN.1
- SNMP - architecture
- SNMP - Protocol
- Object naming - ISO object identifier tree: hierachical naming of all objects, each branch point has name and number
- SNMPv3 - Services
  - authentication assures that message is
    1. form identified source: not altered, not delayed or replayed
    2. message authentification code
  - privacy - encrypts messages using DES
  - acces control
    1. pre configure agents to provide a number of levels of access to MIB for different managers
    2. restricting access to information
    3. limit operations


nit_24_Network_Management - qpdfview
15 / 62
95 %
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
14
SNMP Architecture
SNMP Management Station
SNMP Agent
Management Application
Managed Resources
SNMP Managed Objects
Application manages objects
GetRequest
GetNextRequest
SetRequest
GetResponse
Trap
SNMP Manager
SNMP Messages
SNMP Agent
UDP
IP
Network-dependent protocols
network or internet
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
15


nit_24_Network_Management - qpdfview
16 / 62
95 %
SNMP protocol
Two ways to convey MIB info, commands:
managing entity
request
response
agent data
Managed device
request/response mode
managing entity
trap msg
agent data
Managed device
trap mode
Prof. Dr.-Ing. Adam Wolisz TU-Berlin
WS 2014/2015
16

1.25 Security

- Security goal technically defined
  - confidentiality
    - data transmitted should only be reveal to attended audience
    - confidentiality of entities is also refered as anonymity
  - data integrity
    - detect modified data
    - identify creator of some data
  - accountability
    - possible to identify entity responsible for any communication event
  - availability
    - service should be available and function correctly
  - conrolled access
    - only autherised entities should be able to access certain services or information
- A threat in Communication systems - Def: is any possible event or sequence of actions that might lead to a violation of one or more security goals - the realisation is called an attack
- Threats technically defined - examples
  - masquerade
  - authorization violation
  - denial of communication acts
- Safeguards against Information security threats
  - physical security: locks, environmental controls
  - personnel security: identification, employee screening, security training/awareness
  - administrativ security: controling import of foreign software, software for investagating security breaches
  - emanation security: radio frequency or other electromagnetic emanation controls

  - media security: safeguarding storage of information, scanning media for viruses,
  - lifecycle controls: trusted system design, programming standards/-controls, documentation controls
  - computer security: protecting information while stored (and device itself)
  - communication security: protection of information during transport from one system to another

- Communication Security
  - security service: abstract service which seeks to ensure a specific securtity property , realised by the following
  - cryptographic alogrithm: mathematical transformation of input data (e.g. data, key)
  - cryptpgraphic protocols: a series of steps or message exchanges to achieve specific security objective

- Security services
  - Authenification, Integrety, Confidentiality, Access Control, Non- Repudiation

- Symmetric (same key) vs. Asymmetric (public and private key) Cryptographic Algorithms

- firewall - works at IP layer: so only IP based operations possible (ports, address, -denial)
  - proxy
  - NAT - network address translation
  - packet filtering