SS2011
BAI4-RN
Klausurspicker

R. C. Ladiges

07. Juli 2011

RN

Abkürzungen

A

Ack = Acknowledgement | ADSL = Asymmetric DSL | AES = Advanced Encryption Standards | AH = Authentification Header

ANSI = American National Standards Institute | AP = Access Point | API = Application Programming Interface | AS: Autonomes System

APNIC = Asia Pacific Network Information Centre (IANA) | ARIN = American Registry for Internet Numbers (IANA) | BBM = break-before-make

ARP = Address Resolution Protocol (IP → MAC) | ARPANET = Advanced Research Projects Agency NETwork (US: DoD, NSF)

B

ARQ = Automatic Repeat reQuest | ATM = Asynchronous Transfer Mode | BGP = Border Gateway Protocol (DVR, EGP) | BSS = Basic Service Set

C

BootTP = Bootstrap Protocol (Vorgänger von DHCP) | BSI = Bundesamt für Sicherheit in der Informationstechnologie | CBC = CipherBlock Chaining (DES)

CCMP = CTR-Mode/CBC-MAC Protocol (Integrität) | CDMA = Code Division Multiple Access | CF = CipherFeedback (DES) | CTR = Counter

CH = Corresponding Host | CoA = Care-of-Address | CR = Connection Request | CRC = Cyclic Redundancy Code (aka ZRP) | CT = Care-of test

CSMA = CarrierSense Multiple Access | CSMA/CA = CSMA/Collision Avoidance | CSMA/CD = CSMA/Collision Detection | CTi = CT init

D

DES = Data Encryption Standard | DH = Diffie-Hellman | DHCP = Dynamic Host Configuration Protocol | DMZ = Demilitarized Zone

DoS = Denial of Service | DQDB = Distributed Queue Dual Bus (2 Kabel) | DR = Disconnect Request = Data/Date Request | DWH = TWH

DS = Distribution System | DSL = Digital Subscriber Line | DSSS = Direct Sequence Spread Spectrum | DVR = Distanzvektor Routing

E(F)

ECB = Electronic Code Block (DES) | EGP = Exterior Gateway Protocol (zwischen AS) | ESP = Encapsulating Security Payload | FA = Fremd Agent

F(H)

ESS = Extended Service Set | FDDI = Fiber Distributed Data Interface | FDM = Frequenz Divisions Multiplexing | HA = Home/Heimat Agent

G

FHSS = Frequency Hopping Spread Spectrum | FMIPv6 = Fast Handovers for MIPv6 | GSM = Global System for Mobile Communication (digitales Radiosystem)

H

HDLC = High Level Data Link Control (OSI 2) | HEC = Header Error Control | HMAC = Hash MAC | HMIPv6 = Hierarchical MIPv6 | ICP = Internet control Protocols

I

HoA = Home Address | HOB = High Order Bit (aka MSB) | HT = HomeTest | HTi = HT init | ICMP = Internet Control Message Protocol (OSI 3)

IANA = Internet Assigned Numbers Authority (Technisch, delegiert) | IEEE = Institute of Electrical & Electronics Engineering

ICANN = Internet Corporation for Assigned Names and Numbers (Politisch, Adressbereiche, TLDs, Root-DNS-Server) | INET = ARPANET + NSF

ICV = Integrity Check Value | IETF = Internet Engineering Task Force | IGP = Interior Gateway Protocol (im AS) | IHL = IP Header Length (1 Kopfl)

IKE = Internet key Exchange | IMP = Interface Message Processor (ARPANET) | IP = Internet Protocol (OSI 3) | IV = Initialization Vector

IPX = Internetwork Packet eXchange (Net Ware) | IRTF = Internet Research Task Force (< IETF, Zukunftsforschung) | ISP = Internet Service Provider

L

ISO = International Standard Organization | ITU = International Telecommunication Union | LAN = Local Area Network | LSB = least significant bit

LCP = Link Control Protocol (PPP) | LLC = Logical Link Control 802.2 (OSI 2 zu 3) | LSR = Link-State-Routing (LZR) | LZR = Linkzustands Routing

M

MAC = Medium Access Control (OSI 2 zu 1) = Message Authentification Code | MACA = CSMA/CA | MAS = Medium Access Sublayer

MAN = Metropolitan Area Network (Gebäude, Stadt, no Switches, DQDB) | MBB = make-before-break | MIC = Message Integrity Code

MH = Mobile Host | MIPv6 = Mobile IPv6 | M-i-t-M = Man-in-the-Middle = Meet-in-the-Middle | m. Nr. = mobiler Nutzer (MH)

N

MSB = most significant bit | NAK = Negativ Ack | NCP = Network Control Protocol (PPP) = Netware Core Protocol | NIC = Network Interface Card

NIST = National Institute of Standards & Technology | NSA = National Security Agency | NSAP = Network Service AP | NSF = National Science Foundation

O

Nw = Netzwerk | NWM = NW Maske | OAM = Operation and Maintanance zellen (ATM) | OSPF = Open Shortest Path First (LSR, IGP)

P

OF = Output Feedback (DES) | OSI = Open System Interconnection (von ISO), 7 Schichten | P2P = Peer to Peer | PAR = Positive Ack with Retransmission

PMK = Pairwise Master key | PPP = Point-to-Point Protocol (OSI 2) | PRF = Pairwise Random Factory (2) | PRNG = Pseudo Random Number Generator

QR

PTK = Pairwise Transient key (flüchtig) | QoS = Quality of Service | RARP = Reverse ARP (MAC → IP) | RIP = Routing Information Protocol (DVR, IGP)

RFC = Request for Comments (Einsendung IETF → draft 3-Stimmen → RFC) | RIPE = Réseaux IP Européens (IANA) | RSA = Rivest Shamir Adleman

SSL = Secure Socket Layer

S

RSVP = Resource reSerVation Protocol | RTT = Round-Trip-Time (sec) | RTS = Request-to-Send | SA = Security Association | SqNr = Sequence Number

SAD = SA Database | SAP = Service Advertising Protocol | SHA = Secure Hash Algorithm (von NSA) | SONET = Synchronous Optical Network

Internet

SLIP = Serial Line IP (Kompression aufeinander folgender TCP/IP-Köpfe, OSI 2) | SPD = Security Policy DB | SPI = Security Parameter Index

T

SPX = Sequence Packet eXchange | SSID = Service set Identifier | TC = Transmission Convergence Schicht (OSI 2) | TWH = 3-way-handshake

TLS = Transport Layer Security

TCP = Transmission Control Protocol (OSI 4) | TDM = Time Division Multiplex | TKIP = Temporary key Integrity Protocol | TADU = Transport Protocol Data Unit

UV

TSAP = Transport service AP | UDP = User Datagram Protocol (OSI 4) | UMTS = Universal Mobile Telecommunications System | VK = Virtueller Kanal

W

UTRA = UMTS Terrestrical Radio Access | W3C = WorldWideWeb Consortium | WAN = Wireless Area Network (Land, Kontinent) | (WLAN-stats)

WDMA = WavelengthDivision Multiple Access | WEP = wired Equivalent Privacy | WPA = WiFi Protected Access | WWWD = Worldwide WarDrive

Z

ZRP = Zyklische Redundanz Prüfung (aka CRC) | Protokoll: Vorstellen, Nachrichten, Reaktion, standardisiert (Format, Reihenfolge, Aktionen)

Peer, Trailer, SIM/DUPLEX

Peers: versch. Maschinen, same Schicht/Proto., VK | Trailer = Anhang, Endemarkierung | Simplex: eine Richtung | half-duplex: beide Richtungen, nicht gleichzeitig

Verb. orient.: Etabliere, Nutze, Freigabe v. Sitzung (Telefon), richt. Reihenfolge, VK, Flusskontrolle inkl. | full-duplex: beide Richtungen, gleichzeitig

ISO/OSI

Prinzipien: andere Abstraktionsgranularität → neue Schicht, klar def. Funktion, schon Standards berücks., Datenfluss zu Schichten minimieren

Schichten: Balance groß → Funktionsmix vermeiden klein → handhabbar sein | Protokollstack: Router = 1-3, PC = 1-7

HDLC, SLIP, PPP, TC (INET, ATM)

TCP/IP-Ref-Mod

IPsec / ICMP / IP

TCP/UDP

1	Physikalisch	Bit	1	Host zu Host	wieviel Volt 1/0, ms/bit, init, sim-/duplex?	Framezerlegung/Grenzen
2	DataLink (Sicherung)	Frame/Rahmen			Zerstörung, Verlust/Duplikate auflösen, Übertragungsfehler/Rauschen → resend, Ack, Flussregul.	
3	Netzwerk (Vermittlung)	Paket	2	Internet	wegeauswahl/Routing, Staukontrolle (überlast), Accounting, Bridge in andere Netze	
4	Transport	TPDU	3	Transport	(De)fragmentieren, Flusskontr., Diensttypen (Point2Point kanal, isolierte Nachrichten)	
5	Sitzung	SPDU	TCP/IP Ref. Modell (ARPANET)		S. Aufbau, Dialog kontr., Tokenverw. (crit. sect.), synchronize (bei S. unterbr.)	
6	Darstellung	PPDU			Syntax/Semantik (HTML, XML, ...)	Aufgabe: posi. übertr. zw. benach. Systemen,
7	Anwendung	APDU	4	Anwendung	N/A	Strukturv. Bits → Rahmen, Multiplexbildung, Flusskontr., Fehlererken./behob.

OSI2

Framing

Dienste: a) unbestä. verb. los (no resend), bestä. verb. los (ACK), bestä. verb. orient. (SQNR) | Rahmenbildung/Framing: einteilen, dann Prüfsumme

Framing-Verfahren: a) length vorne b) Anfang/Endezeichen (ASCII) (ADLE B → DLE STX A DLE DLE B DLE ETX) c) steig/sink. Flanke (cross Layer)

d) Anfo./End-Flag: 0111 1110 Bit Stuffing: 1111 1 → 11111 0 | Fehlererkennung (resend) vs. Fehlerbehandlung (richtiges errechnen)

Hamming Code

1 bit-Behebung, Prüfbits an Stellen 2^x, Parität mit Nutzdaten an Posi. K = 11 = 1+2+8, zu k-bit Bündelbehebung (n·k Matrix) erweiterbar

n = |Codewort| = |msg| + |redundanz| | Hamming-Distanz: Anzahl Bitdreher, HD(A,B) := 1 1 (A xor B), HD(0010, 1100) = 3 | Hamming Distanz | OSI2

n+Fehler aufdecken: Codewort mit Distanz n+1 benötigt, n-Fehler beheben: CW mit Dist. 2n+1 benötigt. | bsp: 2bit-Behebung: CD = 2·2+1 = 5, 0→00000, 1→11111

CRC (Prüfsumme, Fehlererkenn.) S/E auf Generator-Polynom G(x) einigen (LSB=1=MSB), XOR und Shift, | Standard-Polynome: CRC-16, CRC-32, ... | CRC

Simpl. Prot. Stop&Wait: Sender wß 4 ACK | Simpl. Prot. f. Rauschbehafteten Kanal: SQNR, PAR, ARQ, Timer → resend | Sliding Window: full duplex auf 1 Kanal

Piggybacking: ACK mit Next Data Package, ACK = SQNR, keine Daten → Timer → nur ACK | ← Fenstergröße = w, Sender = unbest. Pakete, Empfänger: Buffer!

Prob: A&B init. gleichz. d. Verbi. → ACK fürs falsche Paket, Prob: Fensterverschiebung (SQNR wieder 0) Lösung: max(w) = (max-sqnr + 1)/2 | 5 bis max-sqnr

Pipelining: Folge Rahmen send ohne wß 4 ACK, fehlender R.: Strategie a) gehen zurück, verwerfe (Buffer-optimierung), Strategie b) Selektive Wiederholung | E. wß 4 S. ACK-Timeout, resend (Bandbreitenoptim.) | NAKs

PPP: Fehlererk., Authentikation, LCP (Verb. auf-/abbau), NCP (OSI3 aushandeln), Framing (Rahmen zeugen) | PPP

TC: 1B CRC (Kopf), Zelle = 5B kopf + 48B Data, HEC: mit Schieberegister Kopf finden (Suche, Presynch, Synch), OAM (Coderate anpassen) | ↳ meist SONET | TC

MAC (wer redet next?) | FDM (static MAC, jeder eigenes Frequenzband) | Idle-cell: keine Daten → sende empty zelle | MAC

dynamic MAC Annahmen: n Stationen, 1 kanal, blockt bis übertragen, kollisionserkennbar: unbrauchbar → resend, begin: Zeitschlitze v kontinuierlich | OSI2 MAC

mit/ohne Carrier Sense (Kanal abtasten: belegt?) | Poisson-Verteilung: $P_\lambda(k) := \lambda^k / k! \cdot e^{-\lambda}$ (wahrsch. d. k-Ineuerb. i während vers. Zeitabschn.)

ALOHA: 1 Shared Kanal, ankoord. Nutzer | pure ALOHA: jederzeit begin, doppelte übertr. geschw. benötigt (verw. Zeitabsch), p = (100% - koll wahrsch.) | ALOHA

N = avg(1 Rahmen / Rahmenzeit), G = avg(Isend-Rahmen/Rahm-Zeit), G > N hohe Last, G ≈ N geringe Last, Durchsatz / Rahm-Zeit = S = G·P = G·e^{-2G}

slotted ALOHA: feste Intervalle (entspr. Rahmengröße), globale Uhr nötig, S = G·e^{-G} | CSMA (Kanal belauschen → Effizienzsteigerung) | CSMA

1-persistent CSMA: frei? send: next slot | p-pers. CSMA: frei? send m. wahrsch. p: next slot | non-pers. CSMA: frei? send: wß rnd(1 Slotsl)

CSMA/CD (Kollision? → stop, rnd später): Contention Intervall = slotted ALOHA mit Slotgröße = 2τ, mit τ = sendtime (max-range)

WDMA: Frequenzspektr. in Kanäle aufteilen (Data + kontrol für jede Station), Station = R1, R2, T1, T2, R1 = Kontr. hören, R2 (variabel) = T2 hören, | WDMA

WLAN: Hidden-Terminal (C hört nicht A→B), | MACA: RTS/CTR-Rahmen | Transmitter, Receiver T1 (var.) = R1 schreiben, T2 = Daten schreiben | WLAN / MACA

Exposed Terminal (A→B, C könnte C→D denkt belegt) | mit gewünschter/verlaufte Länge, Kontr.-Rahmen kollis. → Lösung: binary exponential backoff

Cocktail Party Analogie: TDM (1 kanal, abwechselnd), FDM (Gruppen/kanäle, lebhaft reden), CDMA (1 kanal, lebhaft reden, sprechen) | CDMA

CDMA: jede Station eigene Chipfolge T (m-Bit Code) für 1 bit (T = 1, $\overline{T}$ = 0), S = Nutzdaten, chipfolgen: paarweise orthogonal: $S \times T := \sum_{i=1}^{m} \frac{s_i \cdot T_i}{m}$ unbegr. Kapa., aber Hauptprob: needzeitsync. | Wideband CDMA ∈ UTRA | bipolare Notation | S x T = 0

CDMA-Bsp: B = -1 -1 +1 -1 +1 +1 +1 -1, C = -1 +1 -1 +1 +1 +1 -1 -1, S = B+C = -2 0 0 0 +2 +2 0 -2 → Empfänger | 0→-1, 1→+1

S x C = Σ(+2 0 0 0 +2 +2 0 +2)/8 = 1 (heißt: C schreibt 1, = -1→0, = 0→nix) | Bsp.: 1010 → +1 -1 +1 -1

Ethernet 802.3 (OSI 1&2), 1-pers. CSMA/CD, Manchester Encoding: jedes Bit 2 zeitintervalle A,B (send 0→ ↓↑, 1→ ↑↓) | Ethernet

MAC: Präambel (7 mal 10101010), Rahmenstart: 10101011, 6B phys. Adressen, Data-Length: max = 1500B, Padding (auffüllen) bei < 46B, CRC-32

binary exponential backoff: nach k = kollisionen, warte rnd([0, 2^k]) Slots | Token Bus: physisch Bus, Logisch Ring (Token) 802.4 (MAC) | Token Bus

jeder kennt linken & rechten Nachbarn, Tokenkreis, Kontrollrahmen, Prior. Klassen, 1B Präambel, Rahmen Start/Ende: 1B Markierung, 1B ab Kontr. Rahmen v Daten, Data < 8182 B, CRC-32 | Token Ring: 802.5 (MAC), P2P, Token, Interface: Strom v. Ring, Bits wieder beim Sender → verwerfe | Token Ring

Modi: höre (bits weiterreichen, 1 bit Verzögerung), übertrage (unterbreche I/O, eigene bits auf Ring) | Kabelbruch: Ring tot → Lösung: Kabel-Center mit Bypass Relay

Differentiales Manchester Encoding: 10 1100: | Data Length nur durch Token Verweilzeit begrenzt, Stationen lokalisieren Brüche: Beacon-Rahmen

Monitor: special Station: überwacht Brüche, doppelte Adressen, Timeout Timer (Token), CRC | Rahmenstatus (A,B)-Bits nicht ECRC, werden Redundant übertragen (2x)

(A=0, C=0) → Ziel nicht erreichbar, (1,0) → erreichbar, Rahmen nicht angenommen, (1,1) → erreichbar, angenommen | User Agent: Prozess zw. NW & user

802.y→x	3	4	5
3		1,4	1,2,4,8
4	1,5,8,9,10	3	1,2,3,8,9,10
5	1,2,5,6,7,10	1,2,3,6,7	6,7

Brücke: OSI2 Header ersetzen, Probleme: ① neue Prüfsumme ② bitreihenfolge umkehren ⑥ Ring leeren ③ Prioritätskopieren ④ fake prioritäten setzen ⑤ Prior. verwerfen ⑦ A&C Bits (willkürlich) setzen ⑧ Stau berücks. (slow/fast LAN) ⑨ berücks. verzög. Token ACK ⑩ verwerfe zu große Rahmen fürs Ziel Netz | Bridge

Spann-Baum Brücke: Brücke mit min (seriennr.) ist Wurzel | Source-Routing Brücke: 5. weiß Weg, setzt HOB = 1 und schreibt Pfad in Rahmenkopf

FDDI: Fiberoptisch Token Ring, LED statt Laser, mehrere Token zeitgleich, 2 Leitungen (Backup bei Bruch) | Schnelles Ethernet: 802.3u, abwärtskompatibel

Rahmen: Präambel, start/Endebegrenzer, Kontr./status (ACK), Adr.-Addr., Prüfsumme, Daten (unbegrenzt) | Bit-Zeit reduzierung 100ns → 10ns

WAN: Satelliten, Prob: 270ms Verzög. → kein CSMA/CD, slotted ALOHA, Kanalallokation v. Bodenstation?, Referenz Bodenstat. periodisch Signal (sync) | an Satelliten, der weiter broadcastet an BS

WLAN 802.11b: 11 Mbit/s 2.4 GHz, 802.11a: 54 Mbit/s 5 GHz, 802.11n: 600 Mbit/s 2.4/5 GHz

Funktestation (AP, OSI2-Bridge) | BSS (AP, Funkzelle) | DS (verbindet BSS's) | ESS (alle BSS an einem DS) | Abhörbar: 150m - 1,5 Km | XOR = ⊕ | OSI2 WLAN

PRNG (seed/Saat als init.) | Chiffrat C = verschluss. plaintext | RC4 (stromverschlüsselungsverfahren): C = PRNG(pw) ⊕ plaintext

WEP: übertragen (PHY-kopf, MAC-kopf, 24 bit key-ID, CRC-32, IV, C = PRNG(IV.PK) ⊕ (plain.ICV)), ICV (4B CRC), RC4 | WEP

IV (3B, 2^{24} Werte → zu klein (IV-wiederholung n. wenigen h), einige Hersteller: IV init. = 0) | M1 ⊕ M2 = (M1 ⊕ K) ⊕ (M2 ⊕ K) Prob. m. Redundanzen | Probleme

Prob: crc(M1 ⊕ M2) = crc(M1) ⊕ crc(M2), Integrität unsicher, Schlüsse plain ↔ crc möglich | Challenge Response (Protokoll: Herausf/Frage → Antwort)

Nachrichtenveränderung: C ⊕ D = (K ⊕ M) ⊕ D = K ⊕ (M ⊕ D), C' = C ⊕ (D|crc32(D)), knackbar in ≈ 60s | WPA (2) 10-15% Durchsatz verlust zu WEP | WPA

WPA (128 > WEP): +TKIP, PMK (256bit, erzeugt 2 PTK), → PRF-X (PMK, Str.konst., MAC-Adressen, ...) = (PTK1 (Data verschl.), PTK2 (MIC-key)) | TKIP

TKIP: konti. Key ändern, RC4, MIC (MICHAEL statt CRC), IV 24bit → 48bit, new Key für jedes Datapacket, Schlüsselerneuerungsprotokoll | MICHAEL

MICHAEL: nur Shift, +, ⊕, 64bit Keys, 32bit Worte, Anfällig: Brute Force, differentielle kryptoanalyse → Angriffserkennung: stop 1 min Verkehr | WPA2

WPA2 (16B > WEP): 128bit Blöcke, AES (temp key) ⊕ Nutzdaten + MIC, CCMP-Pakete: wie TKIP aber ohne ICV | TKIP: MIC = (PTK, MAC-Z/Q, SaNr)

static keys: manuell, PMK = hash(pw), |pw| > 20 | dynamic keys: nach Authentifizierung (802.1x) new PMK von AP

Wegeauswahl/Routing: (Topologie wissen nötig) Eigenschaften: korrekt (ziel), einfach, robust, fair, optimal (kürzester Weg) | Routing | OSI3

a) statisch: off-line im Vorfeld, immergleich, beim Booten aufspielen b) adaptiv (dynamisch): on-line, reagieren auf Topol./Verkehrsänderung, | zweiter b)

Metrik (Distanz, Hops, gesch. RTT), Informationen (lokal, von Nachbarn, von Allen), update alle AT sec, bei Last-/Topologieänderungen

Optimalitätsprinzip (optimaler Pfad): Senke-Baum (kürzeste Pfade von bestimmten Router aus) | kürzester Weg Routing (gewichteter Graph) | Senke Baum

über Dijkstra-Algo.: vom 1.: ∀ Nachbarn: (kürzeste Distanz, Vorgänger) merken, nächsten Nachbarn markieren und von ihm erneut | Dijkstra

P.N | OS13:

Fluten: Paket → alle Ausgänge (außer Eingang), besser: Hop-Zähler (init: Ø(Subnetz), proHop: -1), selektiv (Ausgänge mit angestrebter Richtung)

Distanz Vekt

Flussbasi. Routen: statisch, Topol.- und Last-infos (Verkehrs- & Kapazitäts-matrix) | DVR (aka. Bellman-Ford): jeder Router R.Tabelle beste Entfern. zu jedem Ziel über welchen Ausgang, QueTsec austausch m. Nachbarn: RTT über ECHO zu dir. Nachbarn zu allen anderen Zielen schätzen, Router Weg → zähle bis ∞ prob

Link State

LSR: jeder Router muss: a) beim booten HELLO-Paket auf alle Schnittstellen → Antwort: Adressen von Nachbarn
b) Verzögerung (RTT/2) ermitteln mit ECHO und Timer (Verkehrs-Last berücks. Zeiter Queue: leave Queue)
c) gelerntes an >alle< senden (Link-State-Pakete (ID, SQNR, Age (-1 pro sec, 0 verwerfen), Verzög. zu dir. Nachbarn) Reihenfolge erhalten
d) kürzesten Pfad zu allen Routern berechnen (Senkebaum, Dijkstra) | verschiedene Sichten → Inkonsistenzen, Schleifen, unerw. Effekte

Hierarchisch Mobil

Hierarchisches Routing: Router kennt alle Details d. eigenen Region, Referenzrouter an deren Regionen | Roring mobile Rechner optimal(#Ebenen) = ln|Router|, |Einträge/Router| = e · ln|Router| | m.N (kann auch selbst nach FA fragen), Fremdes LAN, Heimat LAN
Heimat-Lokation (statische Heimatadresse, HOA), FA (broadcast eigene Existenz, kontaktet HA, ACK → m.N. im eigenes Netz aufnehmen)

Broadcast

HA (verifiziert Anfragen von FA nach m.N mit ACK | Broadcast Routing: Alternativen: a) n-fach unicast (einzelne Pakete an alle)
b) Fluten c) Mehrfachziele (1 Paket, n Ziele) d) senke-/spannbaum (gemeinsamer, jeder weiß für wen er zuständig ist, ungeeignet: DVR
e) Entgegengesetzter Pfad-Weiterleiten (würde R unicast an Absender über Eingang schicken ? weiterleiten : verwerfen)

Multicast

Multicast Routing (Sender: spannbaum bauen und ausdünnen zu Gruppen | Staukontr.: global (Router & Hosts) | Flusskontr.: lokal (Sender & Empf.)

Stau

Stau = zu viele Pakete in NW (häufiger Grund: Burst) → Packet loss → resend → System kollabiert | Stauvermeidung: strategien

Stau vermeidung

"undichter-Eimer"-Algo. (endliche Queue, konstante Entnehmetaktung), SV für Multicast-Verkehr (RSVP),
"Token Eimer"-Algo. (Tokens im Eimer, alle ΔT sec Token hinzufügen, solange Tokens: Paketsenden und Token entfernen),
SV in Subnetzen mit VK's (stau → keine neue Verbindung zulassen √ Verbindung nicht durchs Stangebiet leiten)

RSVP

RSVP (mit Spannbäumen, jeder Empfänger reserviert Bandbreite mittels Reservierungs-Nachricht an sender (umgek. Pfad Weiterleiten)

OSI3 Internet

Internetworking Wegeauswahl: IGP (OSPF, RIP) oder EGP (BGP) | Tunneln: Paket in anderes Packet packen und über anderes Netz
Verbindungsknoten: Ebene 1 (Repeater), 2 (Bridge, zw. LANs), 3 (Router, untersch. Netze & Protokolle), >4 (Anwendungs-Gateway)
4 (Transport Gateway, Byteströme auf Transportebene) | Fragmentierung a) transparent (nicht nachvollziehbar) wird wieder zusammengesetzt

Fragmentieren

b) nicht-transparent (nachvollziehbar) kommt fragmentiert zum Empfänger | ID? a) Baumnummerierung: 0.0, 0.1, 1.2.4 ...
b) elementare Fragmentgröße: Paketnr. & Fragmentnr. (Byte Nr.) im Kopf

IPv4

IPv4-Kopf: Version, IHL, Dienst-Typ (verfügbar, schnell, etc.), Ikopf+Daten (max. 65535B), Identifikation (Fragment zuordnung), Q/Z-IP, DF/MF (Don't Fragment, More Fragment), TTL, Optionales, Fragment Offset (Fragmentnr.), OSI4-Protokoll (TCP, UDP), Prüfsumme (auf Kopf, Neuberechnung wegen TTL)

Kopf

IPv4-Kopf-Optionen: Sicherheit (über welche Länder nicht Routen), Quell-Routing (strikt: vollständiger Pfad, lose: unvollständig),
Route festhalten bit (damit sich jeder Router verewigt), Zeitstempel bit (damit jeder Router Zeitpunkt verewigt)

Adressen

MSB	Class	NWM
0xxx	A	/8
10xx	B	/16
110x	C	/24
1110	D Multicast	
1111	E Reserved	

IPv4-Adressen: 32 bit, special: a) alles 0 → selbst (beim booten), b) Netzteil alles 0 → eigenes Netz,
c) alles 1 → broadcast eigenes Netz d) host alles 1 → broadcast in angegebenem Netz, e) 127.x.y.z → loopback

IPv6

IPv6-Adressen: 8-4 zeichen Hex^16B, führende Nullen kürzen (:0000:→::), einmalig Nullen weg (8:0:0:0:0:1:0:A:3 →
IPv6: Routingtabellen verkl., IPsec, v4→v6 migration/co-existenz (ohne Adressänder.) | IPv4 (1080.2B(01F).4B → ::133.168.4.1) 8::0:A:3)

IPv6 Kopf

IPv6-Kopf: Version, Verkehrsklasse (Flusskontr.), Idata!, Hop-Limit, next-header-type (andere Schicht, IPv6, IPsec, etc.), Q/Z-Addressen (16B)
weniger als IPv4: IHL, Fragmentierung (nun fest 576B), Prüfsumme (da in OSI2&4) | IPv6 erweiterungs-köpfe (optional): Fragmentierung, Authentifikation, Verschlüsselung, hop-by-hop (zu prüfende Informationen), Routing (Routen angeben), weitere Zielinformationen

Weg von Datagramm: NW-Addr. ermitteln, suche in Routingtabelle, eigenes Netz ? OSI2 direkt an ziel : send an Gateway (Router)

Internet Controls

ICP = ICMP, ARP, RARP, BootP, DHCP | ICMP (bei unerwartetem Ereignissen: ECHO, TIMESTAMP, falsch geroutet, sender aufs drosseln, TTL=>0
ARP (IP→MAC): broadcast (wer hat IP a.b.c.d? Ich), optimierungen: a) broadcast eigene MAC bei boot | IP-Param. falsch, unerreichbar, kann nicht Fragmentieren
b) cache c) Proxy ARP (Adressen anderer LANs haben) | RARP (MAC→IP): kennt jemand meine IP?, 1 Server pro 1 LAN
BOOTP: Nachfolger von RARP (nun UDP über's Netz (Router) hinweg) 1 Server n-LANs | DHCP: Nachfolger von BOOTP (konfigurierbare workstations
DHCP discover (Host: UDP broadcast 0.0.0.0 → 255.255.255.255), DHCP offer (Server: broadcast Response), DHCP request (Host → Server: bitte um IP)
DHCP ack (Server → Host: IP), DHCP NAK, DHCP decline (Host → server: IP schon vergeben), DHCP release, DHCP inform (Host → Serv.: ich habe die Adresse) statisch

OSPF

OSPF (Nachfolger von RIP): mehrere Metriken (Entfernung, Hops, Verzögerung), Topologieänder., Dienst-basiertes Routing (realtime: 1 Pfad, andere: alle Pfade)
Last über mehrere Pfade, gericht. Graph, AS in Gebiete (Subnetze) aufteilen, Gebiet = 0 → backbone (mit allen anderen Gebieten verbunden)

BGP

Kürzeste Weg nur für eigenes Gebiet | BGP: Router Inter Communication mit TCP, manuelle Policies mögl., vollständige Routen im

IPsec

IPsec (auch für IPv4 als Modul): Modi: a) Transport (Ende-zu-Ende) | Paket (selber Pfad → Schleife, zähle bis ∞ - Prob. gelöst.
b) Tunnel (ISP-zu-ISP/Ende-zu-ISP) | SPD (Schnittstelle zu user, Regelsätze/Filter, Q/Z-Adr., UDP/TCP-ports, etc.) | Modi-Kombination
SA (simplex-Verbindung, SPI (table-lookup, wie zu behandeln, Zieladdr., welches Protok. (AH/ESP), SA-Bundel (Schachtelungsfolge von SA's)

AH

SAD (DB aktiver SA's, Eintrag (Z-Addr., Proto (AH/ESP), SPI, Selektoren aus SPD um SA auszuwählen, SA-key, SA-Algo) | AH (kollid m. NAT)
suche SA in SAD mit (Z-Adr., Proto, SPI)

ESP

Integrität & Authentifikation (& Replay-Attacke), H-Position: nach IP-H, vor Header höherer Schicht
Header: SPI, SQNR, MAC = hash(key, IP-H., AH, rest-daten, key), Length, next-header-type | ESP (kollidiert m. Firewall/Paketfilter)
Vertraulichkeit & Integrität (& Authentifikation & Replay-Attacke), H-Position: wie AH, aber AH besser vor ESP, da früher verworfen werden kann
Header: IV, Payload-Data (verschlüs. Data, DES-CBC √ Null-Algo), Padding (auf 64 bit Blöcke), Padding-Length | Mobile IPv6
optional in Header: MAC = hash(key, SPI, SQNA, IV, rest.data, key) | Transparent f. Endgeräte (nur von stationären betrieben), Roaming

MIP

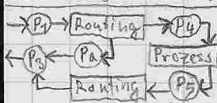
IPs für Wegeauswahl (Netzteil) → Konflikt → um 2. Adresse erweitern, MH, CH (mit dem MH redet), HA (in jedem IPv6-Router), HoA (ID),
CoA (Wegeauswahl), Binding (HoA → CoA) | Bi-direktionales Tunneln (Kommunikations Modi): CH kann kein MIP (transparent)
MH—BU→HA (IPsec ESP Transport), HA repräsentiert MH unter HoA (proxy neighbor discovery), IP-in-IP-Tunnel über HA

2 Chinesen in NYC Problem | Routenoptimierung (Komm. Modi): CH kann MIP, MH—BU→CH, CoA für höhere Schicht m. HoA ersetzen

...,CHspricht direkt mit MH, vorher CoA mit HoA als Optionalem Ziel-Parameter | sicheres Binden (Bu) mit Return Routability 1) MH→CH send 2 Cookies | MIP | OSI3 Internet

1a) HTx: MH–HA→CH (IPsec-Tunnel), 1b) CTx: MH→CH, 2) CH generiert 2 Tokens (HMAC_SHA1(key, HoA+CoA)) sendet CH→MH,
2a) HT: CH–HA→MH, 2b) CT: CH→MH, 3) MH generiert aus Tokens einen SHA1-key zum Bu Authentifizieren (HMAC_SHA1(Bu, Tokenkey))
gering unsicher, da M-i-t-M im Netz von CH mögl. | Handover (HO) – Verzögerung: neue CoA dauert bis CH bekannt (Prob: RTT)
Ansätze zu HO: BBM (nach dem CoA-Wechsel das BU), MBB (besser: 2 Interfaces für alte und neue CoA, BU vorm altCoA trennen)
Roaming Verbesserungen: HMIPv6 (verteilte HA-Proxies), FMIPv6 (HO Vorhersage/Prognose aufgrund bekannter Topologie

Firewall: Paketfilter (OSI3&4, meist zustandslos), Anwendungsfilter | DMZ: Bereich zw. 1. (Internet) und letzter (Netzwerk) FW | Firewall



P1=Preroutings P2=Forward P3=Postrouting P4=Input P5=Output
Filter: P2, Pu, P5 | nat: P1, P3, P5 | mangle: P1, P5
iptables: Reihenfolge wichtig (top-down), Tabellen (filter, nat, mangle)
Regeln: ACCEPT, DROP, QUEUE (→eigene Chain), RETURN (diese Tabelle skip)

OSI4 Aufgaben: Verbindung zw. Anwendungen/Benutzern (nicht nur zw. Hosts), | OSI4
Multiplexen von Verbindungen, verlässlicher Dienst (QoS), Daten Puffern, Fluss-/Stausteuerung | QoS: Parameter werden ausgehandelt | QoS
QoS-Paras: Verzögerzeit, Fehlerwahrsch., Durchsatz, Rest-Fehlerrate, Schutz, Priorität, Elastizität (interne Probleme v. OSI4)
einf. Transp.Dienst: listen(port), connect(IP, port), send(data), receive(), disconnect() | states: idle → active/passive establishment pending → active/passive disconnect pending → established | Socket
Berkley Sockets (API): -disconnect(), +close(), +socket(), +bind() | (API)
Adressierung: hierarchisch (<...><nw><host><port>), flach (2. Abbildungsebene erforderlich), Internet: IP&Port, ATM: AAL-SAPs
Aufbau verzög. Dupli
Etablieren einer Verbindung (mit TWH) Prob m. verzög. Duplikaten: NW kann Pakete verlieren, speichern, duplizieren
NW überlastet → TPDUs kommen zu spät an, versch. Pfade → brauchen versch. Lange, Lösung: a) wegwerf Transport Adressen
b) Verbindungs nr. & SQNR, c) lösche veraltete Pakete (hops/zeitstempel) Zeitstempel prob: Uhrsync, begrenzt ZR. variable (Sliding window)
TWH: OK: →CR(saNR=x), ←ACK(saNR=Y, ACK=x), →DATA(saNR=z, ACK=y) CR-Duplik.: →CR(saNR=x), ←ACK(saNR=Y, ACK=x), →REJECT(ACK=Y) | 3-way-Handshake
CR&Data Duplik.: →CR(SQNR=x), ←ACK(SQNR=Y, ACK=x), →DATA(SQNR=z, ACK≠y) fällt, →REJECT(ACK=Y)
Aufheben einer Verbindung: asymetrisch (abrupt/einseitig → data loss) v symmetrisch (jede Richtung explizit DR) | Aufheben
symmetrisch probs: halboffene-Verbindungen (Timeout) | Puffer: a) fix size b) vario-size c) zirkulärer Puffer pro Verbindung | Puffer
Sender hält TPDUs bis ACK (Terminal), Empfänger hält TPDU (File transfer) | UDP-Header: Q/Z-Port, Length, Prüfsumme (Pseudokopf, UDPkopf, Daten) | UDP
TCP-Kopf: Q/Z-Ports, SQNR, ACK-Nr (nextermarkt SQNR), HeaderLength, datalength, prüfsumme (TCP, Pseudokopf), | well-known-ports: 0–1023 | TCP
Flags: URG(ent), ACK, PSH(direkt→OSI7), RST(verb. zurücksetz.), SYN(verb. Etablieren), FIN(verb. aufheben) | registred-ports: 1024–49151
Optionen: Nutzdatenlast (default: 536 B), Fensterskalierungsfaktor (F.size), selektive wiederholung (NAK) | private-ports: 49152–65535 | ←AR
TCP: DWH (rename CR/ACK/DATA→SYN), Aufhebung (FIN/ACK, m. Timer (2·P.lifetime)), "Silly Window" Syndrom (E: nur 1B lesen), P2P, Full-duplex → ↳x-simplex
TCP: Stankontr.: garantiertes Empfänger-Fenster, stau-Fenster ermitteln (übertrage min(E-Fenster, S-Fenster) Bytes)
S-fenster s: init(s=max_segmentsize), slow start (kein Timeout→S*=2), schwellwert (kein Timeout→S++)
Timeout→schwellwert=S/2; s=init; | TCP-Timer: 1) Übertragungswiederholung (wie lang w 8 4 ACK? RTT+4·D)
2) Persistenz (Deadlocks) 3) Keepalive (verb. ungenutzt) 4) Auflegen d. Verb. (2·max(p.lifetime)) | D=αD+(1−α)·RTT-Zeit
Verbindungszustände: IDLE, WAITING, QUEUED, ESTABLISHED, SENDING, RECEIVING, DISCONNECTING | Ereignis×Zustands-Matrix | OSI4
Pakete: call_Request (etablieren), call_Accepted (CR ACK), clear_request (Aufheben), clear_confirmation, Data, Credit (Kontroll Paket)
TCP-Puffer: voll = noch 1 Byte f. Bekanntmachen d. Fsize erwartet (oder „dringliche" Daten), Naglets Algo: S schickt 1B, erst nach ACK den Rest | TCP
Drahtlos TCP/UDP: Prob: Annahme d. Stau aber Paketverlust, Verlust: macht slow, statt fast resend Lösung: individuelles TCP

SSL/TLS OSI7 Ende zu Ende: IP(TCP(TLS+Record(data))) encrypt(authentif.(compress(fragment(data)))) | Alice=send, Bob=rec, | OSI7
Eve (darf E und D kennen): lese msg., get key, modify msg., „ich bin Alice" | Kryptext M, Chiffrat C, Key K, | Eve=böse | Crypto
Encrypt: $C=E_{KE}(M)$, Decrypt: $M=D_{KD}(C)$, $M=D_{KD}(E_{KE}(M))$ | Kryptoverfahren: symmetrisch ($K_E=K_D=K$), asym. ($K_E \neq K_D$)
Cryptology = Cryptoanalyse (brechen) ∧ Cryptography (gut werfen) | abhören/modifizieren = passiver/aktiver Angriff
Schutzziele: Vertraulichkeit, Datenintegrität, Authentifikation (Signatur), Nicht-Abstreitbarkeit (das Kommunikation, S und E)
Angriffsformen: ciphertext only, known plaintext (und passendes C), chosen plaintext (Zugriff K_E), chosen ciphertext (Zugriff K_D)
DES: 1686 bit K, Blockchiffre M→64 bit, Feistel Chiffre 64 bit (effektiv: 56), 16 Iterationen (Runden) mit eigenen Teil K's | DES
Encrypt (entgegengesetzte Reihenfolge Teil k's), Modi: a) ECB (alle 64 bit seperat, austauschbar), b) CF (Byte Verschlüsselung),
c) OF (auch wie b), d) CBC (Block 0→IV xor, Block i−1 ⊕ Block i, entschlüsseln erst bei vollständigem 64bit Block Empfang
Pro Runde i: linker Blockteil $L_i=R_{i-1}$, rechter Bl.Teil $R_i=L_{i-1} \oplus f(K_i, R_{i-1})$, f=kernfunktion DES (Init./Expans. P-Box und
Doppel DES: DES(DES(M)), Meet-in-the-Middle (ECB): bekannt ($P_i, C_i, C_i=E_{K2}(E_{K1}(P_i))$) | Tripple DES | 8 S-Boxen | 2×DES
Ansatz: $D_{K2}(C_i)=E_{K1}(P_i)$ → ermittel keys, so schwach wie 1×DES | $C=E_{K1}(D_{K2}(E_{K1}(M)))$, $M=D_{K1}(E_{K2}(D_{K1}(C)))$, 2^{112} | 3×DES
RC5: gut, patentiert, IDES=Rijandel: best, serpent: very strong, 3×DES: 2. best | EDE statt EEE, damit kompatibel zu DES ($K_1=K_2$ setzen)
Public-key Cryptography: DH-Schlüsselaustausch, RSA (Faktorisierungsproblem (a·b=c, c bekannt, was ist a und b?)
RSA: Private key d, Public Key e, ① Primzahlen p, q wählen ② RSA-Modul n=p·q ③ Eulersche φ-Funktion von n: φ(n)=(p−1)(q−1) | RSA
④ wähle e mit gcd(e, φ(n))=1 ⑤ berechne d, mit d·e≡1 (mod φ(n)) ⑥ $C=m^e \bmod n$ ⑦ $m=c^d \bmod n$ | e<φ(n)
kongruenz: a≡b (mod n) : ist a=b+n·k | erw. Euklidischer Algo: Init($m_1=\varphi(n), n_1=e, s_0=1, t_0=0, u_0=0, v_0=1$)
Q=Faktor, R=Rest, for(i=1; $R_i \neq 0$; i++){ $Q_i=m_i/n_i$; $R_i=m_i \bmod n_i$; $m_i=n_{i-1}$; $n_i=R_{i-1}$; $S_i=U_{i-1}$; $T_i=V_{i-1}$;
$U_i=S_{i-1}-Q_i \cdot U_{i-1}$; $V_i=T_{i-1}-Q_i \cdot V_{i-1}$; } | s=k, T=d, $e \cdot d+k \cdot \varphi(n)=1$ | $a^3 \bmod n = ((a^2 \bmod n) \cdot a) \bmod n$
→ n=gcd(e, φ(n)) wenn d<0, dann d+=φ(n)

Informationen zur Signatur

	**Unterzeichner**	EMAILADDRESS=robin.ladiges@haw-hamburg.de, CN=Robin Christopher Ladiges
	Datum/Zeit	Thu Jul 07 01:04:01 CEST 2011
	Austeller-Zertifikat	CN=CAcert Class 3 Root, OU=http://www.CAcert.org, O=CAcert Inc.
	Serien-Nr.	44727
	Methode	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signatur)