Die TCP/IP–Protokollfamilie

Übersicht

- TCP/IP und das Internet
- Schichtenmodell
- IP, UDP, ARP
- Routing
- TCP und Congestion Control
- Anwendungen
- Angriffspunkte

TCP/IP und das Internet

- hardware–unabhängig
- Entwicklung seit 1983 (DARPA)
- keine zentrale Verwaltung
- kein zentraler Knoten (im Gegensatz zu SNA)
- Ausfallsicherheit als Designziel (DOD)
- Routing Paket–per–Paket

TCP/IP und das Internet

- Applikationsunabhängig
- Standards: “RFC” (Request for Comment), numeriert – auf vielen Servern verfügbar, z.B. http://www.ietf.org
- Auch auf der Chaos–CD enthalten
- Im Gegensatz zu den ISO/OSI–Protokollen kein “offizieller” Standard, aber weithin akzeptiert

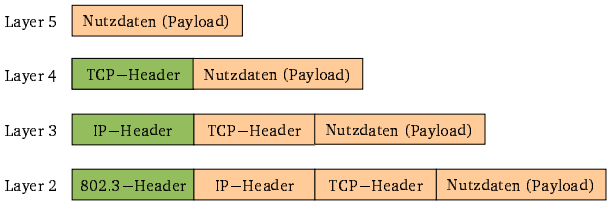
Protokolle

- Paketorientiert
- Schichtenmodell
- Adressierung

Internet- Schichtenmodell

Konzeptebene	*Bezeichnung*	Beispiel
Layer 5	*Application Layer*	HTTP
Layer 4	*Transport Layer*	UDP/TCP
Layer 3	*Internet Layer*	IP/ICMP
Layer 2	*Data Link Layer*	IEEE 802.3
Layer 1	*Physical Layer*	UTP/LWL

Encapsulation



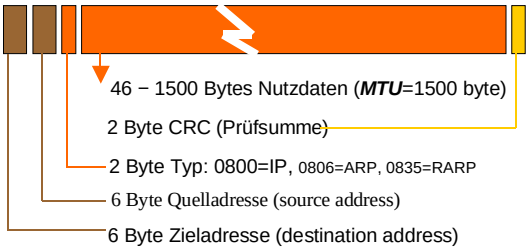
Link Layer

- Ethernet
- Token Ring
- HDLC (für WAN)
- ISDN
- usw.

Ethernet

- 10 oder 100 Mbit/s: Gleiches Protokoll
- Frame–Typ: Ethernet–II
- 6 Byte Adressen, weltweit eindeutig (in der Hardware festgelegt)
- CSMA/CD, “Shared media”, nicht kollisionsfrei, stochastisches Verhalten
- Neuerdings auch “switched Ethernet”, kollisionsfrei (gleiches Protokoll)

Ethernet (RFC894)



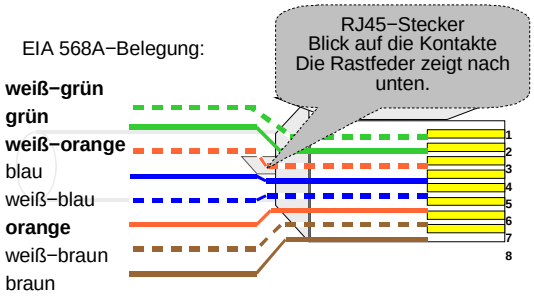
Der Datenteil muß notfalls auf 46 Bytes aufgefüllt werden.

Exkurs: Ethernet–Kabel

- 10Base–2, Cheapernet
- Koaxial
- RG58 Kabel, 50 Ohm
- Bus–Verkabelung, T–Stücke
- Kabel muß mit je 50 Ohm terminiert sein
- Max 185 m / Segment
- 10 Mbit/s

- 10Base–T usw.
- Verdrillte Paare
- EIA568 Cat.3/4/5
  4 Adernpaare (2 genutzt)
- Stern–Verkabelung vom Hub
- Max. 90+10 m/Strang
- fehlertoleranter
- 10 und 100 Mbit/s

Exkurs: RJ45–Stecker



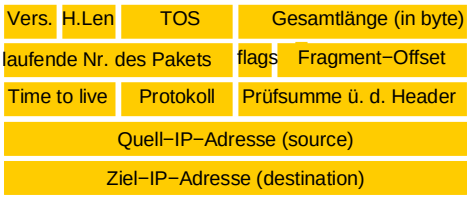
IP – Internet Protocol

- Paketvermittelnd
- Ungesichert
- verbindungslos
- 32 bit Adressen, meist als 1.2.3.4 (dezimal) geschrieben

IP – Adressen

- Eine Adresse gehört jeweils zu einem Interface. D.h. ein Rechner mit mehreren Netzwerkkarten hat auch mehrere Adressen.
- Wenn Daten zu groß für den nächsten Hop, werden sie zerlegt (fragmentiert)
- Mehr Details später

IP Paketaufbau



20 Byte Header, plus Optionen

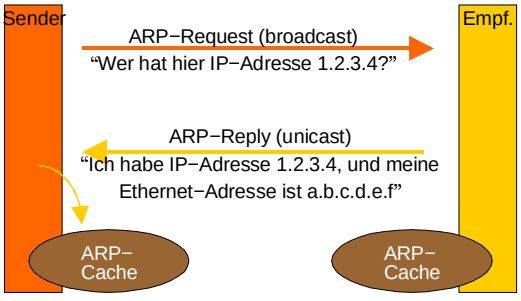
ICMP: Internet Control Message Protocol

- Für Status– und Fehlermeldungen
- "host unreachable"
- "network unreachable"
- PING nutzt ICMP echo request/reply
- Teil von IP (network layer), nutzt aber IP–Pakete zur Datenübertragung

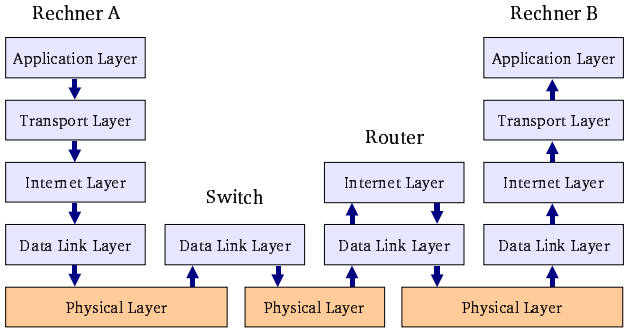
ARP: Address Resolution Protocol

- Zur Zustellung von IP–Paketen im LAN wird die physische (z.B. Ethernet–) Adresse benötigt
- Ethernet–Adressen (48 bit) sind weltweit eindeutig in der Hardware "eingebrannt"
- Logische IP–Adressen (32 bit) sind vom Netzverwalter festgelegt
- ARP erlaubt Umsetzung

ARP



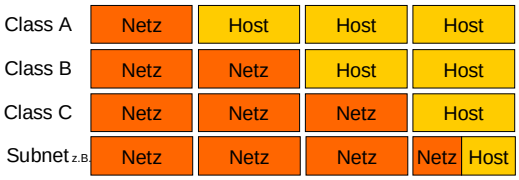
Kommunikation im Internet



Routing

- IP–Adressen sind 32 bit organisiert.
- Es gibt Class–A, Class–B und Class–C Netze. Diese unterscheiden sich durch die Netzmaske, die angibt, welcher Teil der Adresse das Netz und welcher den Rechner innerhalb des Netzes bezeichnet.
- Routing–Entscheidungen werden nur nach dem Netz–Teil der Adresse getroffen!

Netzmaske



Es gibt nur wenige Class B und noch weniger Class A Netze.
Class C–Netze, oder Blöcke davon, sind das Übliche.
Die Klassen wurden eingeführt, um die Tabellen in den Routern klein zu halten.
Es ist möglich, die Netzmaske unabhängig von der Klasse frei zu setzen, um ein Netz weiter zu unterteilen (subnetting)

Netzmaske

- Adresse 192. 168. 153. 23
- Netzmaske 255. 255. 255. 240

- Adresse & Netzmaske 192. 168. 153. 16
- (bitweises UND)
- Netz–Teil der Adresse 192. 168. 153. 16
- Host–Teil der Adresse 0. 0. 0. 7

Routing–Ablauf

- Aufspalten der Zieladresse in Netz– und Host–Teil (durch AND mit der Netzmaske)
- Durchsuchen der Routing–Tabelle nach dem errechneten Netzteil
- Wenn gefunden: An das in der Tabelle eingetragene Interface senden (next hop)
- Sonst: Zur Default–Route, falls vorhanden
- Ansonsten: ICMP “Network unreachable”

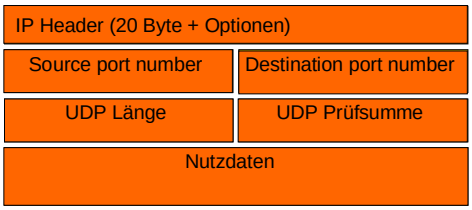
Routing–Protokolle

- Routing–Protokolle dienen dazu, die Routingtabellen automatisch zu pflegen, wenn z.B. Wege unpassierbar werden
- Es gibt verschiedene Protokolle
  - RIP: Fuer LAN geeignet, viel Traffic
  - EGP, BGP: Fuer WAN / ASN zu ASN
- Unter UNIX in routed (nur RIP) bzw. gated implementiert

UDP – User Datagram Protocol

- Simpelstes Internet–Protokoll auf Transportebene (Ebene 4)
- Verbindungslos
- Ungesichert – “fire and forget”
- Anwendungen: z.B. DNS, NFS
- Zusätzlich zum 20 Byte IP–Header noch zwei 16 bit Portnummern, 16 bit Länge und eine 16 bit Prüfsumme

UDP Paketaufbau



Ports

- UDP und TCP verwenden “Ports” als Erweiterung der IP–Adresse.
- Eine Datenübertragung ist durch 4 Adressen gekennzeichnet:
  - Quell–Adresse, Quell–Port
  - Ziel–Adresse, Ziel–Port
- Ports beschreiben einen Prozess auf einem Rechner

Transport Layer

verbindungsloser, ungesicherter Datentransport

UDP
User Datagram Protocol

keine Garantie, ob Daten in der richtigen Reihenfolge oder überhaupt ankommen

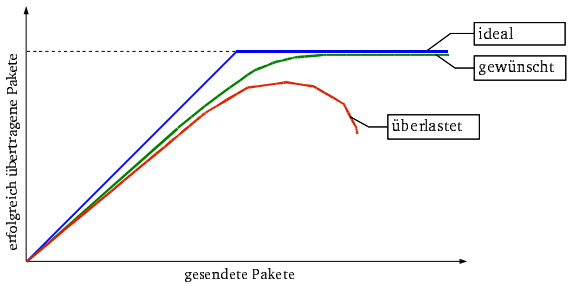
verbindungsorientierter, gesicherter Datentransport

TCP
Transmission Control Protocol

Verbindungsaufbau, Sortierung und automatische Wiederholung bei fehlerhafter Übertragung

➡ TCP dient zum zuverlässigen (gesicherten) Datentransport

Überlastzusammenbruch



Überlastabwehr

unelastischer Verkehr

braucht feste Bandbreiten, keine Anpassung der Rate möglich

Überlastabwehr durch Rufannahme (Call Admission Control)

Beispiel: Telefonie, Video Im Internet ein Problem!

elastischer Verkehr

kann seine Rate an den Zustand des Netzes anpassen

Überlastabwehr durch Ratenregelung (Congestion Avoidance)

Beispiel: ftp, E–Mail Protokoll: TCP

➡ TCP dient zur Ratenregelung der Quellen

TCP Grundprinzip (1)

- Byte–Stream–Transport
  - Anwendung übergibt Byte–Strom an TCP–Protokollmaschine
- Segmentierung
  - TCP–Protokollmaschine zerlegt Byte–Strom in nummerierte Segmente
- Sendefenster („Congestion Window“)
  - Segmente innerhalb des „Congestion Windows“ werden an IP–Layer übergeben

Sequenznummern: 1 2 3 4 5 6 7 8 9 10

Congestion Window

TCP Grundprinzip (2)

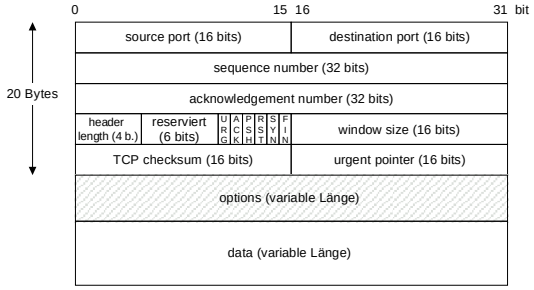
- Acknowledgements
  - Empfänger schickt Empfangsbestätigungen für jedes korrekt empfangene Segment zurück
- Sliding Window
  - Für jedes bestätigte Segment wird das „Congestion Window“ soweit nach rechts verschoben, daß sich links vom Sendefenster nur bestätigte Segmente befinden

Sequenznummern: 1 2 3 4 5 6 7 8 9 10

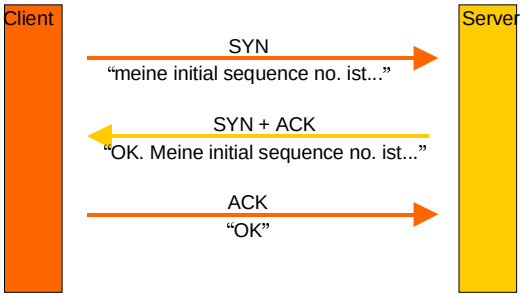
Congestion Window

- Retransmission
  - unbestätigte Segmente werden erneut übertragen

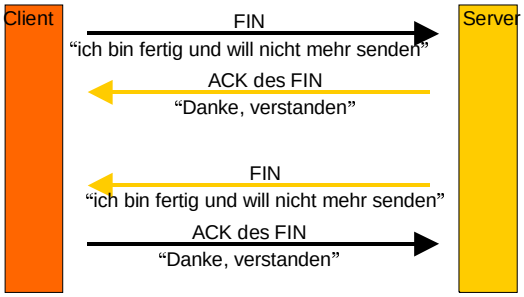
Aufbau eines TCP–Segments



TCP – Verbindungsaufbau



TCP – Verbindungsabbau



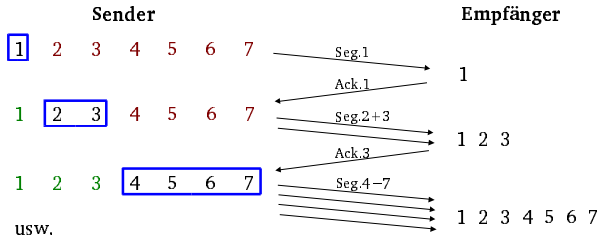
Netzknoten (Gateways)

- Warteschlangen im Router
  - heutzutage werden fast ausschließlich FIFO–Warteschlangen verwendet
- TCP erhöht die Rate so lange, bis Warteschlange überläuft
- es kommt zu Paketverlusten
  - TCP geht davon aus, daß Paketverluste **ausschließlich** Folge von Warteschlangen–überläufen sind
  - die Rate wird reduziert
  - Problem: „natürliche Verluste“, z.B. beim Mobilfunk

 ➡ Ratenregelung basiert auf Paketverlusten

Ratenregelung (1)

- Die Senderate wird durch die Größe des „Congestion Windows“ eingestellt

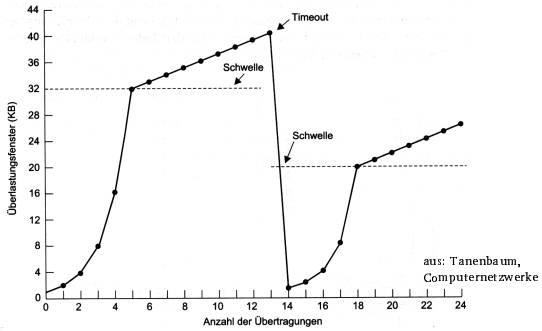


➡ Dieser Algorithmus heißt **„Slow Start“**

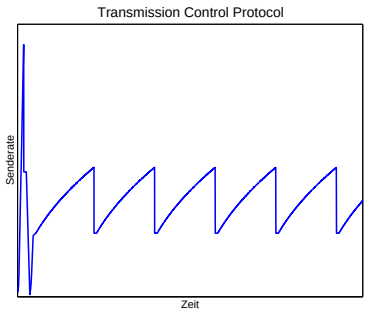
Ratenregelung (2)

- Rate übersteigt „Slow Start Threshold“
  - Wechsel vom Slow Start zum Congestion Avoidance Algorithmus
  - Im Congestion Avoidance Algorithmus wird das Sendefenster nur noch um **ein** Segment vergrößert, wenn **alle** Segmente des alten Fensters bestätigt wurden (linearer Anstieg)
- Paketverlust führt zur Ratenreduktion
  - Reduktion des Sendefensters auf ein Segment im Slow Start – Fall
  - Halbierung des Sendefensters im Congestion Avoidance – Fall

Slow Start und Congestion Avoidance

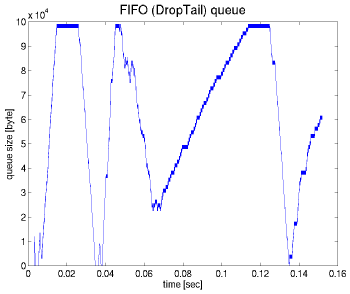


TCP im Single Bottleneck



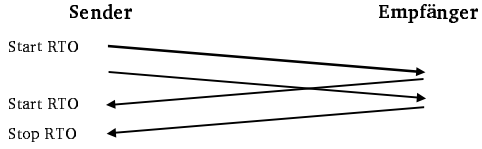
➡ TCP (Congestion Avoidance) ist ein AIMD–Algorithmus

FIFO Queues



Loss Detection and Retransmission

- Erkennung von Paketverlusten durch Ablauf des „Retransmission TimeOut“ (RTO) Timers

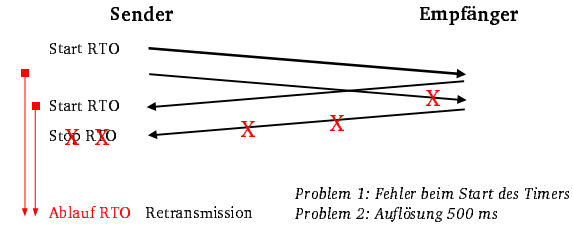


RTO = durchschnittliche Round Trip Time (RTT) +
+ 4 x gemittelte Standardabweichung der RTT–Messungen

Genauigkeit der RTT–Messungen: 500 ms

Loss Detection and Retransmission

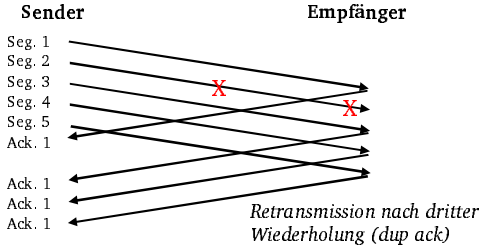
- Erkennung von Paketverlusten durch Ablauf des „Retransmission TimeOut“ (RTO) Timers



➡ RTO–Mechanismus führt zu sehr schlechten Datenraten

Loss Detection and Retransmission

- Erkennung von Paketverlusten durch doppelte Acknowledgements



➡ Dieser Algorithmus heißt „**Fast Retransmit**“

Fast Recovery

- neu eingeführt in TCP Reno (1997)
- doppelte Bestätigungen zeigen an, daß ein Paket den Empfänger erreicht hat
- das Sendefenster wird entsprechend vergrößert, so daß neue Pakete in das Netz eingespeist werden können
- wurden alle verlorenen Pakete bestätigt, wird das Sendefenster wieder verkleinert

TCP Varianten (1)

- TCP Tahoe (1989)
  - RFC 1122, 2001 (Standards)
  - Implementiert Slow Start, Congestion Avoidance, Fast Retransmit
- TCP Reno (1997)
  - RFC 2581 (Standard)
  - modifizierter Fast Retransmit mit Fast Recovery
- TCP NewReno (1999)
  - RFC 2582 (Experimental)
  - modifizierter Fast Retransmit mit „Partial Acks“

TCP Varianten (2)

- TCP SACK–Option (1996)
  - RFC 2018 (Standard)
  - selektive Acknowledgements bestätigen jedes einzelne empfangene Segment
  - Problem: Empfänger muß SACK ebenfalls unterstützen
- TCP Vegas (1995)
  - basiert **nicht** auf dem AIMD–Algorithmus
  - Ratenregelung anhand der gemessenen RTT
  - „bester“ Algorithmus, da er Paketverluste reduziert
  - Problem: verliert gegen Tahoe/Reno bei gemischten Netzen

TCP Varianten (3)

Verlust:	TCP Tahoe	TCP Reno	TCP NewReno	TCP SACK
Ein Segment	45 %	55 %	55 %	55 %
Zwei Segmente	45 %	37 %	52 %	52 %
Drei Segmente	43 %	12 %	47 %	52 %
Vier Segmente	42 %	11 %	42 %	51 %

Effektiver Durchsatz verschiedener TCP Versionen bei unterschiedlicher Anzahl verlorener Pakete

TCP Varianten (4)

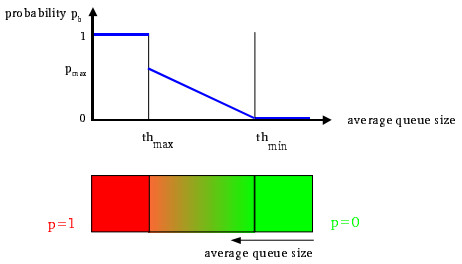
Version	Anteil	Beispiel
Tahoe	15 % davon 71% SACK	ohne SACK: www.ebay.com mit SACK: www.microsoft.com
Reno	23 % davon 3% SACK	ohne SACK: www.tu-harburg.de mit SACK: www.infospace.com
NewReno	62 % davon 46% SACK	ohne SACK: home.netscape.com mit SACK: www.aol.com
SACK Option insgesamt	41 %	

Quelle: Sally Floyd, www.aciri.org/tbit

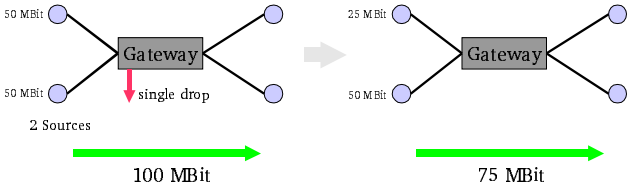
Neue Entwicklungen (1)

- keine Paketverluste mehr
  - Congestion wird durch Paketmarkierungen angezeigt
  - in Verbindung mit Random Early Detection (RED) Gateways
  - Explicit Congestion Notification (ECN)

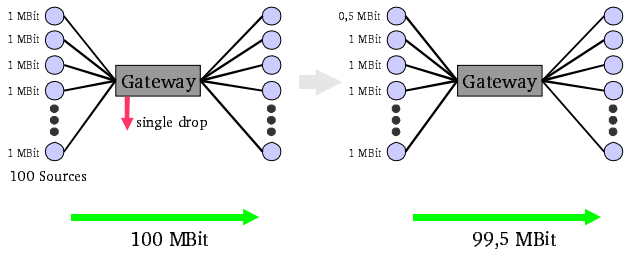
Random Early Detection (RED)



Influence of Congestion Signals (1)



Influence of Congestion Signals (2)



Neue Entwicklungen (2)

- „Congestion Pricing“
  - Senderaten werden mit einem „Nutzen“ verbunden
  - globaler „Nutzen“ soll optimiert werden
  - Unterteilung des globalen Optimierungsproblems in lokale Probleme bei den Quellen
  - Feedback-Signale aus dem Netz (Preise)
  - Preise beschreiben Schweregrad der Überlastung
  - Sender optimiert seinen Nutzen in Abhängigkeit von seiner Zahlungsbereitschaft

Nameserver

- Für Menschen sind numerische Adressen schwer merkbar
- Daher sorgt das DNS (Domain Name System) dafür, daß leichter merkbare Namen verwendet werden können
- Nameserver bilden eine weltweit verteilte Datenbank

Nameserver

- Die Namen sind hierarchisch organisiert, z.B.
  blackbox.fasel.bla.de
- Eine Anfrage fragt nach
  - de? Deutschland! → ns.nic.de
  - bla.de? → ns.bla.de
  - fasel.bla.de.de? → ns.fasel.bla.de
  - Resultat: 195.21.208.23

Anwendungen

- Die meisten Anwendungen basieren auf TCP, nur wenige auf UDP.
- UDP–basierend sind NFS (Network File System) und DNS.

UDP–Anwendungen

- NFS: Network File System, entwickelt von Sun.
  Relativ langsam, aber extrem stabil. Das Protokoll ist “stateless”, d.h. alle Status–information liegt nur auf dem Client. Der Server kann zwischendurch neu starten, ohne daß der Client dies normalerweise merkt.
  NFS Version 2 kann auch TCP verwenden.

UDP–Anwendungen

- DNS: Domain Name system
  Clients fragen über UDP bei den Name Servern an.
  Die Server antworten ihrerseits per UDP.
- Traceroute: Feststellen, wie meine Daten zum Empfänger kommen.
  Nur aktuelles Bild, das nächste Paket kann schon einen anderen Weg nehmen.

TCP–Anwendungen

- Telnet: Terminal–Emulation, login
- SSH: Das “bessere Telnet”, verschlüsselt
- FTP: Dateitransfer
- HTTP: Übertragungsprotokoll des WWW
- SMTP, POP3: Electronic Mail
- usw.

Angriffspunkte

- Dieses Thema bildet einen besonderen Schwerpunkt einer getrennten Veranstaltung.
- Daher hier nur ein knapper Überblick.

Angriffspunkte

- Denial of Service: Berechtigte Nutzer können nicht arbeiten
- Ausspähen von Daten durch passives Mitlesen
- Verfälschen von Daten unterwegs
- Aktiver Eingriff in Netzknoten (Rechner, Router)
- IP Spoofing

Denial of Service

- Flood ping usw.: Überlasten eines Netzes oder Rechners
- Ping mit zu großen Paketen
- SYN–flooding
- Illegale Fragmente
- Beruht meist auf Fehlern in der IP–Implementierung des angegriffenen Systems

Ausspähen von Daten

- Mitlesen (Sniffer) im LAN
  - Bei Ethernet einfach
  - Zur Fehleranalyse oder Spionage
  - etherfind, tcpdump, RMON–Probe
- Mitlesen in WAN–Zwischenstationen, z.B. beim Internet–Provider
- Log–Dateien, z.B. WWW Proxy

Einschleusen von Daten

- Einschleusen anderer Daten (unter falscher Identität)
- Übernahme (hijacking) bestehender Verbindungen, z.B. juggernaut
- Oft bei applikationsspezifischen Client–Server–Systemen, z.B. Datenbanken

IP Spoofing

- Setzen der Quell–Adresse im IP–Header auf eine andere Adresse
- Oft eine Adresse im angegriffenen LAN
- Basis für weitere Attacken
- Keine Antwort möglich
- Einfache Protokolle, wie SMTP, funktionieren auch ohne Antwort (Antwort vorhersehbar) –> E–Mail Spam

Literatur

- Tanenbaum, Computer–Netzwerke
- W.Richard Stevens, TCP/IP Illustrated, Addison–Wesley (besonders Vol.1)
- Olaf Kirch, Linux Network Administrators Guide, LDP bzw. O'Reilly
- Douglas Comer, Internetworking with TCP/IP, Prentice–Hall
- RFCs