Rechnernetze I
Übungsblatt 5

Anne Martens, Felix J. Oppermann

5. Juni 2006

1 IP/ICMP (50)

1. Anwendung

   - Erreichbarkeit eines Hosts: ICMP-Echo-Request-Nachrichten können verwendet werden, um die Erreicharkeit einen Hosts zu prüfen. Das Systemprogramm ping geht so vor.
   - Ermittlung der Route durch ein Netz: ICMP-Time-Exceeded-Nachrichten können verwendet werden, um die Route, die ein Paket durch ein Netz nimmt, nachzuvollziehen. Ändern die Router zwischendurch ihr Routing, ist die Angabe nicht mehr richtig, da der Sender in Wirklichkeit mehrere Pakete mit zunehmendem TTL sendet, und durch die zurückkommenden Time-Exceeded Nachrichten die Route nachvollziehen kann.
   - Ermittlung der maximalen Datagramm-Größe auf einem Pfad: Hier setzt der Sender das Don't-Fragment-Bit der Pakets und testet mit unterschiedlichen Paketgrößen das Maximum aus. Die Router antworten mit der ICMP-Nachricht Fragmentation Required, falls das Paket zu groß ist. Auch hier darf sich die Route nicht währenddessen ändern, ansonsten sind die vorherigen Ergebnisse ungültig.

2. ping und traceroute

```
anne@rossfeld:~/> ping 193.45.3.16
PING 193.45.3.16 (193.45.3.16): 56 data bytes
^C
--- 193.45.3.16 ping statistics ---
8 packets transmitted, 0 packets received, 100% packet loss

--> Der Host scheint unerreichbar!

anne@rossfeld:~/> ping www.bundeskanzlerin.de
PING www.bundeskanzlerin.de (195.43.53.88): 56 data bytes
64 bytes from 195.43.53.88: icmp_seq=0 ttl=248 time=9.104 ms
64 bytes from 195.43.53.88: icmp_seq=1 ttl=248 time=9.625 ms
64 bytes from 195.43.53.88: icmp_seq=2 ttl=248 time=9.044 ms
64 bytes from 195.43.53.88: icmp_seq=3 ttl=248 time=9.217 ms
64 bytes from 195.43.53.88: icmp_seq=4 ttl=248 time=9.051 ms
64 bytes from 195.43.53.88: icmp_seq=5 ttl=248 time=9.334 ms
^C
--- www.bundeskanzlerin.de ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max/stddev = 9.044/9.229/9.625/0.204 ms

anne@rossfeld:~/> ping 134.106.55.1
PING 134.106.55.1 (134.106.55.1): 56 data bytes
64 bytes from 134.106.55.1: icmp_seq=0 ttl=63 time=0.586 ms
64 bytes from 134.106.55.1: icmp_seq=1 ttl=63 time=0.530 ms
64 bytes from 134.106.55.1: icmp_seq=2 ttl=63 time=0.530 ms
64 bytes from 134.106.55.1: icmp_seq=3 ttl=63 time=0.524 ms
64 bytes from 134.106.55.1: icmp_seq=4 ttl=63 time=0.529 ms
64 bytes from 134.106.55.1: icmp_seq=5 ttl=63 time=0.514 ms
64 bytes from 134.106.55.1: icmp_seq=6 ttl=63 time=0.557 ms
64 bytes from 134.106.55.1: icmp_seq=7 ttl=63 time=0.526 ms
^C
--- 134.106.55.1 ping statistics ---
8 packets transmitted, 8 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.514/0.537/0.586/0.022 ms

--> Beide Hosts sind erreichbar.

anne@rossfeld:~/> /usr/sbin/traceroute 193.45.3.16
traceroute to 193.45.3.16 (193.45.3.16), 64 hops max, 40 byte packets
 1  cata-ol2.HRZ.Uni-Oldenburg.DE (134.106.11.1)  1.766 ms  0.323 ms 0.292 ms
 2  cata-ol1.hrz.uni-oldenburg.de (134.106.204.1)  1.592 ms  0.566 ms 0.511 ms
```

```
 3  ar-oldenburg1-ge4-3.x-win.dfn.de (188.1.47.205)  1.669 ms  0.304 ms 0.290 ms
 4  cr-hamburg1-po6-0.x-win.dfn.de (188.1.18.161)  3.461 ms  3.395 ms 3.419 ms
 5  hbg-b2-pos1-0-0.telia.net (213.248.103.97)  3.676 ms  3.789 ms  3.670 ms
 6  hbg-bb2-link.telia.net (80.91.249.202)  46.380 ms  3.736 ms  3.730 ms
 7  s-bb2-pos7-3-0.telia.net (213.248.64.37)  48.727 ms  49.606 ms  49.238 ms
 8  s-b4-pos5-0.telia.net (213.248.66.14)  16.112 ms  16.106 ms  16.193 ms
 9  s-hdn-i4-link.telia.net (80.91.249.209)  16.525 ms  16.434 ms  16.441 ms
10  s-hdn-i2-link.telia.net (80.91.249.229)  49.435 ms  48.509 ms  48.900 ms
11  * * *
12  * * *
13  * * *

--> Die Route kann nicht weiter aufgelöst werden.

In beiden Fällen kann der Grund sein, dass der Router die TTL des Antwortpakets
zu gering setzt. Ein häufig verbreiteter Bug ist es, die TTL auf den verbleibenden
Wert zu setzten... das wäre 0!

anne@rossfeld:~/> /usr/sbin/traceroute www.bundeskanzlerin.de
traceroute to www.bundeskanzlerin.de (195.43.53.88), 64 hops max, 40 byte packets
 1  cata-ol2.HRZ.Uni-Oldenburg.DE (134.106.11.1)  1.898 ms  0.415 ms 0.741 ms
 2  cata-ol1.hrz.uni-oldenburg.de (134.106.204.1)  1.584 ms  0.574 ms 0.536 ms
 3  ar-oldenburg1-ge4-3.x-win.dfn.de (188.1.47.205)  1.577 ms  0.297 ms 0.279 ms
 4  cr-hamburg1-po6-0.x-win.dfn.de (188.1.18.161)  3.427 ms  3.355 ms 3.382 ms
 5  cr-berlin1-po0-0.x-win.dfn.de (188.1.18.109)  48.611 ms  39.086 ms 47.970 ms
 6  init-ag.bcix.de (193.178.185.99)  8.927 ms  8.900 ms  8.830 ms
 7  * * *
 8  * * *
 9  *^C

 --> auch hier kann die Route kann nicht weiter aufgelöst werden.

anne@rossfeld:~/> /usr/sbin/traceroute 134.106.55.1
traceroute to 134.106.55.1 (134.106.55.1), 64 hops max, 40 byte packets
 1  cata-ol2.HRZ.Uni-Oldenburg.DE (134.106.11.1)  1.731 ms  0.323 ms 0.280 ms
 2  einstein.kowalk.Informatik.Uni-Oldenburg.de (134.106.55.1)  4.008 ms 0.458 ms  1.496 ms

 --> eine kurze Route.
```

3. Das Programm ,,ping" verwendet ICMP-Echo-Request-Pakete um eine Antwort durch den zu überprüfenden Server zu erreichen. Die Zeit zwischen Dem Absenden des Request und dem empfangen der Antwort wird gemessen und so die Laufzeit durch das Netz bestimmt

   Das Programm ,,traceroute" nutzt ICMP-Time-Exceeded-Nachrichten. Dem Rechner zu dem die Route bestimmt werden soll werden Pakete mit steigender Time to live gesendet. Die Paketer können den Server jedoch aufgrund der zu geringen Lebensdauer nicht erreichen und die Zwischenstationen an denen die Lebensdauert erreicht wurden antworten mit ICMP-Time-Exceeded-Paketen. Durch die Steigende Lebensdauer werden antworten jeder Zwischenstation empfangen, so dass die Route bestimmt werden kann.

4. Header von IPv4 und IPv6
   In beiden Versionen vorkommende Headerfelder (die sich evtl. leicht in der Funktionalität unterscheiden, aber prinzipell dasselbe sind): Version, Diensttyp bzw. Priorität, Lebenszeit bzw. Hop Limit, Transport bzw. Next Header, Sender- und Empfängeradresse.

   In IPv4 wird die Gesamtlänge eines Datagramms sowie die Länge des Headers angegeben, in IPv6 nur die Payload Length, die Länge des Paketinhalts (ohne Header aber inklusive der Erweiterungs-Header), die Länge des Headers ist schließlich fix. Somit kann dasselbe ausgedrückt werden.

   Felder die in IPv6 nicht vorkommen mit daraus folgenden Einbußen und Ausgleichen:

   - Informationen zur Fragmentierung.
     Einbußen: Router können Pakete nicht bei Bedarf fragmentieren und defragmentieren.

Ausgleich: Die Fragmentierung ist durch Extension Headers möglich, hier kann aber nur der Sender bzw. der Empfänger fragmentieren bzw. defragmentieren.

- Kopfprüfsumme
  Einbußen: Die Korrektheit des Pakets kann nicht auf der Schicht 3 geprüft werden.
  Ausgleich: Man überlässt die Fehlererkennung den Schichten 2 und 4. Da das Paket in jedem Router durch Verändern des TTL verändert wurde, musste auch die Prüfsumme jedes Mal neu berechnet werden. Dies kann nun eingespart werden.
- Optionen
  Einbußen: Es können im ersten Header keine Optionen zum Network Management oder zum Testen bei neuen Implementierungen eingebaut werden.
  Ausgleich: Durch Extension Headers können diese Optionen ausgeglichen werden: Es gibt auch einen Header im IPv6, der von allen weiterleitenden Routern gelesen wird: der Hop-by-Hop Options-Header.
- Füllbits
  Werden ohnehin nicht gebraucht, also ein Vorteil.

5. Abarbeitungsgeschwindigkeit

   - Keine Prüfsumme Im IPv4 beinhaltet der Header eine Prüfsumme, mit der der Header auf Richtigkeit geprüft werden kann. Da sich bei jedem Weitersenden eines IPv4 Datagramms die TTL (time-to-live) verändert, muss auch diese Prüfsumme jedes Mal neu berechnet werden. Diese Berechnung möchte man in IPv6 ersparen und fügt nun keine Prüfsumme mehr ein.
   - Keine Fragmentierung und Defragmentierung
     Im IPv6 kann nur der Sender fragmentieren und nur der Empfänger defragmentieren. Ein Router verwirft ein Datagramm, dass nicht ausreichend klein ist. Bei IPv4 wurden Pakete auch von Routern fragmentiert, wenn sie zu groß für die Leitungen waren. Die Verantwortung, ausreichend kleine Datagramme zu erzeugen, wird also von den Routern auf den Sender verlagert und senkt somit die Last der Router.
   - Knapper Header
     Der Hauptheader enthält nur die absolut notwendigen Informationen, für weitere Optionen werden optionale Zusatzheader verwendet. So muss ein Router weniger auswerten.

6. Fragmentierte Datagramme
   Ein Datagramm kann in IPv6 nur vom Sender fragmentiert werden. Dazu sollte der Sender zunächst herausfinden, was die maximale Größe eines Pakets auf dem Weg zum Empfänger ist. Ist der Wert kleiner als das zu sendende Datenpaket, wird dieses fragmentiert. In alle Fragmente wird zusätzlich zum Hauptheader ein Fragmentheader eingeführt, der durch das Kennzeichen 44 im Next-Header Feld des Vorgängerheaders gekennzeichnet wird. Jeder Fragmentheader enthält eine Identifikation des jeweiligen Fragments sowie die Position des Fragments im Gesamtdatenpaket (Fragment Offset) Das letzte Fragments wird durch das nicht-setzen des More-Bit gekennzeichnet. Der hauptsächliche Unterschied zu IPv4 besteht darin, dass die Fragmentierung nicht mehr durch Router durchgeführt wird. Erhält ein Router ein IPv6 Paket, das zu groß für die Weiterleitung ist, so verwirft er es und sendet eine ICMPv6-Fehlermeldung Too Big (Typ 2) zurück. Da Router nie fragmentieren, entfällt das Do-not-Fragment-Bit. Für fragmentierte Datagramme ist der Aufwand im IPv6 jedoch höher, da ein zusätzlicher Header eingefügt werden muss.

2 Adress-Auflösung (20)

1. - Address Resolution Protocol (ARP)
   - Neighbor Discovery Protocol (NDP)
   - Domain Name System (DNS): Auflösung von Domainnamen zu IP-Adressen

2. Adress-Auflösung bezeichnet die Zuordnung einer logischen Adresse, beim IP-Protokoll beispielsweise der IP-Adresse, an eine Hardwareadresse, wie zum Beispiel die MAC-Adresse.

3. Zunächst prüft die Station A ob sich die Hardwareadresse der Station B im eigenen Speicher für Adressbindungen befindet. Wir gehen hier davon aus dies sei nicht der Fall, so dass Station A einen Broadcast sendet um die Zieladresse zu ermitteln. Station A erstellt hierzu eine Anfrage für die IP-Adresse 172.16.1.18. Die Bridge leitet die Anfrage in das zweite Teilnetz weiter, da es sich um einen Broadcast handelt, der immer weitergeleitet wird. Station B kann die Anfrage folglich empfangen und antwortet mit einem an Station A adressierten ARP-Antwortpaket mit der eigenen Hardwareadressse. Die Adresse von Station A wird

zugleich in den eigenen Speicher für Adressbindungen aufgenommen. Die Bridge erkennt, dass sich das Ziel im ersten Teilnetz befindet und leitet das Paket weiter. Station A empfängt die Antwort und nimmt die Hardwareadresse im Speicher für Adressbindungen auf. Im folgenden können Pakete unter zuhilfenahme des Speichers direkt korrekt adressiert werden, bis der Speicher gelöscht wird und das Verfahren wiederholt werden muss.

3 TCP (70)

1. **Verbindungsorientiert** Das TCP-Protokoll stellt virtuelle Verbindungen zwischen zwei Hosts bereit.

   Duplex-Verbindungen Die durch das TCP-Protokoll bereitgestellten Verbindungen ermöglichen eine gleichzeitige Kommunikation in beide Richtungen.

   Punkt-zu-Punkt-Übertragung Die Datenübertragung erfolgt immer zwischen genau zwei Teilnehmern.

   Zuferlässigkeit Das Protokoll stellt eine unverfälschte und vollständige Übertragung der Daten sicher. Übertragungsfehler werden duch das Protokoll selbstständig behandelt.

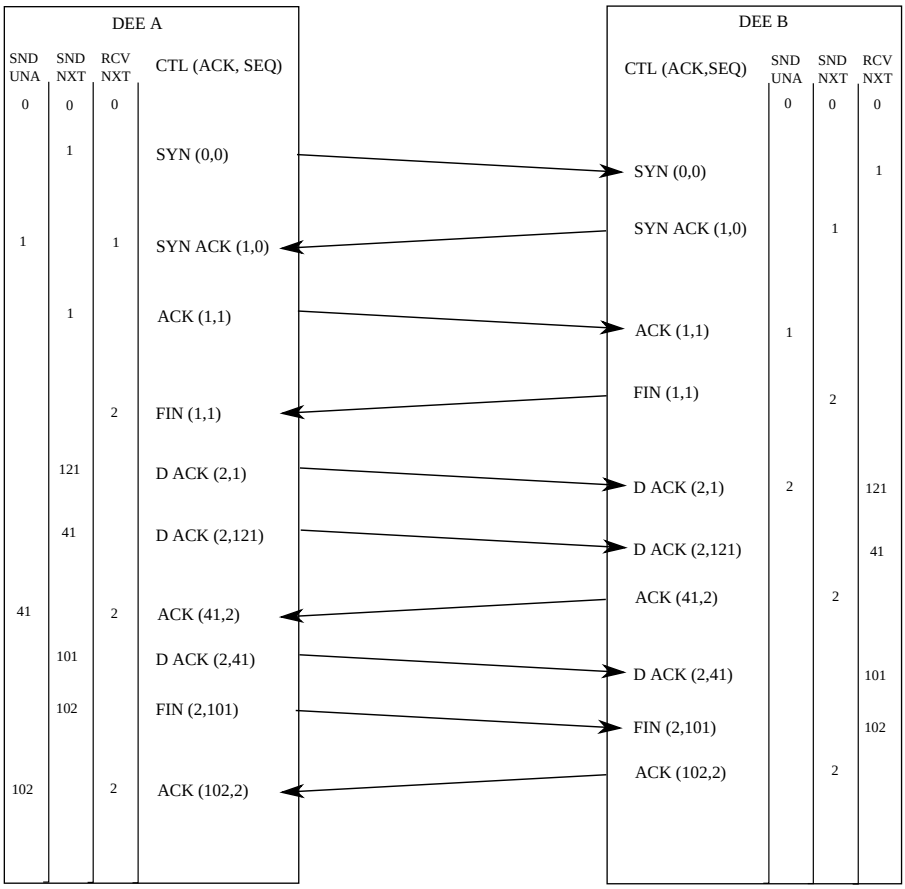
   Stromorientiert Die Daten werden in der gesendeten Reihenfolge an höhere Schichten übergeben.

   Gepufferte Übertragung Die Daten können durch einen Puffer zwischengespeichert werden bis ein Versand möglich ist. Höhere Schichten müssen auf Verzögerungen keine Rücksicht nehmen.

   Unstrukturierter Strom Die Struktur der übertragen Daten wird nicht vorgegeben sondern kann durch die Anwendung bestimmt werden. Es erfollgt eine unbearbeitet Übertragung der Rohdaten.

   Zuverlässiger Verbindungsauf und -abbau Kurz nacheinander aufgabautet Verbindungen beeinflussen sich nicht. Es kommt zu keiner Verfälschung von Daten. Beim Verbindungsabbau werden noch ausstehende Daten zuverlässig übertragen auch wenn eine Station die Verbindung vorzeitig abbaut.

2. (a)



(c) Im TCP Header wird die Größe des Pakets nicht angegeben, es ist nur in der Prüfsumme enthalten. Die Länge kann dem Pseudo-Header entnommen werden. Der Pseudo-Header enthält die Länge des TCP-Pakets. Diese Information wird durch das IP-Protokoll bereit gestellt.

(d) Die Verbindung ist beendet, wenn beide Stationen jeweils die Anfrage nach einen Verbindungsabbau mit einem Acknowledge bestätig haben. Beim aktiven Schließen ist die Verbindung erst nach einem Warten um die zweifache maximale Segment Lebensdauer vollständig beendet.

(e) Zur Berechnung der Prüfsumme werden zusätzlich die Absender- und Empfänger-IP-Adressen, der Protokolltyp und die Länge des TCP-Packets benötigt. Diese Daten werden in Form eines Pseudo-IP-Header in die Berechnung einbezogen und entstammen der zugrundeliegenden Schicht.

3. Eine aktive TCP-Verbindung wurde selbst durch Angabe eines Ports auf einem anderen Rechner initiiert. Dies ist in der Regel beim Clientzugriff auf einen Server der Fall. Bei einer passiven Verbindung wartete der Rechner auf einem bestimmten Port auf Verbindungsversuche. Mit dem Anfragenden Rechner wird eine Verbindung aufgebaut. Dies wird in der Regel für Serverprozesse Verwendet. In diesem Fall übernimmt ein neuer Prozess die Kommunikation mit dem Client und der ursprüngliche Prozess steht sofort wieder für weiter Verbindungsversuche zur Verfügung

4. Nach dem Verbindungsaufbau könnten noch ,,verlorene" Packete einer unsauber beendeten Verbindung eintreffen. Um zu verhindern, dass eine neue Verbindung durch Packete aus einer vorhergehenden Verbindung gestört werden kann wird für die doppelte maximale Lebenszeit der Packte ein erneuter Verbindungsaufbau auf dem Port verhindert.

4 UDP (60)

1. Eigenschaften
   UDO stellt dem Anwender einen Mechanismus zur verbindungslosen Kommunikation zur Verfügung. Das Konzept ist als analog dem von IP, setzt jedoch auf der Anwendungsschicht auf. UDP erlaubt es, einem Anwendungsprozess auf einem anderen Rechner ein Datagramm zu senden, wobei der Anwendungsprozeß durch eine Portnummer identifiziert wird.

2. RFC
   RFC 768 spezifiziert UDP an sich. RFC 1700 spezifiziert teilweise festgelegte Portnummern, unter denen Anwendungsprozesse zu erreichen sind.

3. Anwendungsprotokolle

   - TFTP Trivial File Transfer Protocol
   - DNS Domain Name System, verwendet standardmäßig UDP, kann aber auch auf TCP aufsetzen.
   - DHCP Dynamic Host Configuration Protocol
   - NTP Network Time Protocol, verwendet standardmäßig UDP, kann aber auch auf TCP aufsetzen.
   - RTP Real-Time Transport Protocol für Voice-over-IP
   - Freenet, ein Peer-to-Peer-Netz
   - BOOTP Bootstrap Protocol
   - L2TP Layer 2 Tunneling Protocol, ermöglicht das Herstellen einer VPN-Verbindung, auch TCP.

4. Adressen
   Das UDP Paket wird immer fest in ein IP Paket eingebettet, deswegen kann davon ausgegangen werden, dass die IP-Adresse des Senders sowie des Empfängers schon im IP Paket steht. Ansonsten würde man die Adressen immer jeweils zweimal senden.

5. Fehlererkennung
   Der UDP Header enthält eine Prüfsumme, die mit dem UDP Header sowie vorangestellt den IP Adressen des Senders und des Empfängers, der UDP Protokollnummer (17) und der Länge des UDP Pakets gebildet wird. So können Fehler im UDP Paket selbst erkannt werden. Weiterhin kann auch überprüft werden, ob ein Paket auch tatsächlich den richtigen Rechner erreicht hat, da die IP Adressen mit einbezogen werden. Zugleich wird aber die Trennung zwischen der Netwerkschicht (IP) und der Transportschicht (UDP) aufgeweicht.

6. Sockets
   Im UDP werden ebenfalls Portnummern verwendet. Da Sockets definniert sind als die Konkatenation einer IP-Adresse eines Hosts mit einer Port-Adresse gibt es auch in UDP Sockets. Die Angabe des Quell Ports ist allerdings in UDP optional, somit gibt es nicht beim Sender nicht immer einen Socket. Die Sockets werden im Gegensatz zu TCP nur zum Senden einer einzelnen Nachrihct benötigt, da UDP verbindungslos ist.