Internet: Technology, Protocols and Services

Homework Assignment 1 - ANTWORTEN

1. **Given the current growth of the Internet and a human population of (then approximately) 10 billion, when will every human being on the planet have an IP address?**

   Nach dem Bild „Internet Growth“der Folien verzehnfacht sich die Zahl der Rechner mit IP Verbindung alle 3.5 Jahre. Damit wird etwa im Jahre 2006 die 10 Mrd. Grenze überschritten.

2. **Compare the NSAP and IPv4 address formats.**

   NSAP-Adressen gehören nicht nur zu einem speziellen Rechner, sie bestimmen auch den Zugangspunkt zu dem Transportprozeß. IP-Adressen identifizieren nur eine Maschine eindeutig. Um diesen Nachteil zu umgehen, werden zusätzlich feste Portnummern benutzt (vgl. Frage 14).

3. **Sketch (pseudo-code) an algorithm for reassembling IP fragments.**

   Der folgende Algorithmus verwaltet eine Tabelle *T* von Einträgen (*p.id*) für jeden Absender und Protokolltype. Der Algorithmus arbeitet auch fehlerfrei, wenn die fragmentierten Pakete nicht in der richtigen Reihenfolge ankommen. Die Kontrolle, ob fragmentierte Pakete komplett empfangen sind, ist nicht optimiert.

```
p.id = packet identification
p.mf = packet 'more fragments' flag

for each arriving IP packet p:

if ( p.id in T ) {
  insert p.id into buffer based on offset
  check if last fragment has zero p.mf
  for each fragment i in buffer{
    check if offset(i-1) + length(i-1) = offset(i) (packet complete?)
    if both checks are true
      pass packet p.id from chain buffer to upper layer
  }
}
else if ( p.mf ) {
  add p.id to T
  start reassembly timer
  start buffer chain for p.id
}
else {
  /* no fragmentation */
  pass p to upper layer
}

timer expires:
  remove p.id from T
  remove p.id from buffer chain
```

4. **How could you discover the minimum MTU on a path?**

   Ein Host sendet ein Paket mit dem do-not-fragment Flag zum Zielrechner. Wenn ein Link auf dem Weg eine solche Paketgröße nicht transportieren kann, dann verwirft er das Paket und sendet eine ICMP-Fehlermeldung zurück. Dadurch erfährt der Sender, daß der MTU-Wert (Maximum Transport Unit) kleiner sein muß. Nun sendet er ein kleineres Paket und tastet sich somit an den wahren MTU-Wert heran. In neueren IPv4-Versionen

sowie in IPv6 wird in der ICMP-Meldung die maximale Paketgröße des Linkes mit angegeben. (Siehe auch in W. Richard Stevens: ICP/IP Illustrated Volume 1, Seite 151ff. oder RFC 1191.)

5. What are the advantages and disadvantages of doing hop-by-hop vs. end-to-end fragmentation and reassembly?

Wenn die fragmentierten Pakete nach jedem Link mit zu kleinem MTU-Wert wieder zusammengesetzt werden, dann werden sie unter Umständen auf dem Weg mehrfach fragmentiert. Dies bedeutet zusätzlicher Rechenaufwand im Netz und Verzögerung. Allerdings können dann außerhalb dieser Links große Pakete komplett übertragen werden, das spart Bandbreite im Netz.

6. What are reasons for limiting the MTU in the data link layer?

Alle Netzwerke haben eine maximale Paketgröße bedingt durch das interne Übertragungsprotokoll. (z.B.: Ethernet 1500 Byte, FDDI 4352 Byte). Diese Beschränkung ist notwendig, da das Übertragungsmedium von einer Maschine nur eine begrenzte Zeit belegt werden darf. (Andernfalls käme es bei sehr langen Paketen in dem Netz zu Blockierungen.)

7. Compare the German telephone number system with the IPv4 and IPv6 address assignments.

Jedem Ort in Deutschland ist eine lokale Vermittlungsstelle (im Festnetz der Telekom) zugeordnet. Diese Vermittlungsstelle hat eine feste Nummer, in vielen Orten fällt diese Nummer mit der Vorwahl zusammen. In großen Städten besteht diese Nummer aus der Vorwahl und z.B. in Berlin den ersten drei Ziffern der Rufnummer. Da diese großen Städte nun kurze Vorwahlen haben, ist die Rufnummernlänge in ganz Deutschland fast gleich. Durch diese Zuordnung zu einer lokalen Vermittlungstelle sind somit alle Ziffern bis auf die letzten vier oder fünf durch die räumliche Lage festgelegt. Die letzten Ziffern werden „willkürlich“ vergeben. Damit kann der Nummernraum sehr gut ausgenutzt werden. IPv4-Nummern werden üblicherweise in Gruppen an Institutionen vergeben, die einen geschlossenen Nummernraum für sich beanspruchen. Da man schlecht abschätzen kann wieviele IP-Adressen nach 2 oder 3 Jahren nötig sind, werden oft deutlich größere Nummernbereiche belegt als genutzt. Eine räumliche Zuordnung gibt es in dem IPv4 nicht. Der Nummernraum in dem IPv6 ist bekannterweise mit 128 bit Adressen absichtlich überdimensioniert. Dadurch wird aber eine mehrfache Zuordnung möglich. Das heißt aus einer IPv6-Adresse kann man in Zukunft Ort, Provider und Benutzer ablesen.

8. What is the single greatest advantage of having the IP checksum cover only the datagram header and not the data? What is the disadvantage? (C7.1)

Jeder Router prüft die Prüfsumme, wenn sie falsch ist wird das Paket einfach weggeworfen. Ist der Datenanteil eines IP-Paketes verfälscht worden, so kann dieser Fahler nicht vom IP erkannt werden. Sollte die IP-Prüfsumme über das gesamte Paket berechnet werden, dann entstände in jedem Router ein deutlich höherer Rechenaufwand. Ein Vorteil wäre natürlich, daß verfälschte Pakte nicht erst bis zum Adressaten übertragen werden - um dort weggeworfen zu werden. Allerdings ist es bei der Übertragung von synchronen Daten (z.B.: Audio-, Videoanwendungen) meist sinnvoll, lieber die verfälschten Daten zu bekommen als gar keine.

9. Is it ever necessary to use an IP checksum when sending packets over an Ethernet? Why or why not? (C7.2)

Im Ethernet wird eine Prüfsumme über das ganze Paket berechnet. Trotzdem ist die Verwendung der IP-Prüfsumme sinnvoll, da auch Verfälschungen im Rechner auftreten können.

10. What are the advantages of doing reassembly at the ultimate destination instead of doing it after the datagram travels across one network?

vgl. Frage 5

11. When a minimum size IP datagram travels across an Ethernet, how large is the frame?

Ethernetheader (14 Byte) + IP-Header (20 Byte) + Pad (26 Byte) + Prüfsumme (4 Byte)= 64 Byte

Da die minimale Ethernetpaketgröße 64 Byte ist, werden dem Paket Pad-Byte zugefügt. Eine minimale Paketgröße wurde festgelegt, um leichter eine Unterscheidung zu ungültigen Rahmen treffen zu können.

12. If all Internet service providers used supernetting and assign subscribers numbers from their block of addresses, what problem occurs when a subscriber changes from one provider to another? (C10.20)

Bei der Verwendung von Supernetting kann ein Provider seinen Adressraum beliebig in Unternetzwerke einteilen. Damit kann der gesamte Adressraum genutzt werden. Der Nachteil des Supernetting (abgesehen von dem höheren Aufwand beim Routing) zeigt sich bei dem Wechsel des Providers. Da der gesamte Adressraum des Supernetting einem Provider zugeordnet sein muß, kann dieser nicht für die Benutzung bei einem anderen Provider übernommen werden. Bei diesem Wechsel würden sich somit alle IP-Adressen ändern.

13. Consider a fixed subnet partition of a class B network number that will accommodate at least 76 networks. How many hosts can be on each network? (C10.5)

Ein Class B Netzwerk hat 16 bit Adressraum. Um darin 76 Netzwerke durch Subnetting zu unterscheiden, sind 7 bit notwendig (2^7 = 128). Jedes Teilnetz kann 2^9-2=510 Hosts fassen. (512 abzüglich der Netzwerkadresse: xxxx xxxx x000 0000 und der Broadcastadresse: xxxx xxxx x111 1111). Damit wird ca. 60% des verfügbaren Nummernraums benutzt.

14. What is the chief advantage of using protocol ports instead of process identifiers to specify the destination within a machine? (C12.8)

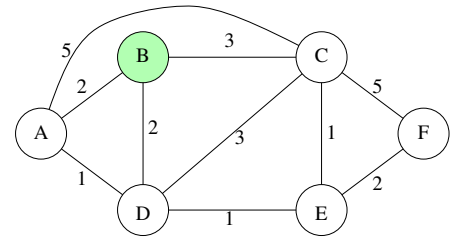
Die Prozeßnummern (in UNIX-Systemen) werden in aufsteigender Richtung vergeben. Die Prozeßnummer (PID) bspw. eines FTP-Dämons kann somit nach jedem Booten oder Neustart des Dämons eine andere sein. Die Portnummern können „nach Wunsch“ belegt werden. Dadurch hat ein externer Prozeß die Möglichkeit Interprozeßkommunikation mit einem Prozeß zu beginnen ohne deren PID oder dessen Namen zu kennen.

15. What are the advantages and disadvantages of using UDP (instead of TCP) for DNS?

Die Aufgabe eines Domain-Name-Servers (DNS) ist es den Namen eines Hostes in eine IP-Adresse umzuwandeln. Die dabei zu übertragenden Daten sind nur wenige Byte (A-Record). Diese Datenmenge kann bequem in einem UDP-Paket übertragen werden. Würde eine TCP-Verbindung aufgebaut werden, dann müßten für die gleiche Anfrage mindestens 6 TCP-Pakete übertragen werden. Abgesehen von der unnötigen Netzbelastung dauert dieses Verfahren auch wesentlich länger.

16. Compute the link-state routing table with source B instead of A.

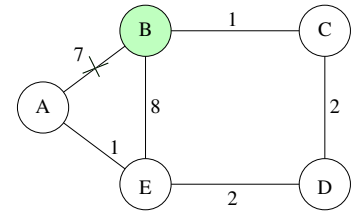
Dijkstra's algorithm



step	N	d(A)	d(C)	d(D)	d(E)	d(F)
0	B	2,B	3,B	2,B	∞,-	∞,-
1	AB		3,B	2,B	∞,-	∞,-
2	ABD		3,B		3,D	∞,-
3	ABCD				3,D	8,C
4	ABCDE					5,E

17. For the five-node network shown in class, show the effect of poison reverse when link AB goes down.

Distance vector routing:



	B	C	D	E
initially	6,C	5,D	3,E	1,A

Über den Link AB geht kein Verkehr - es ändert sich nichts.

18. How might one implement load balancing between several hosts using DNS?

DNS ist eigentlich dafür verantwortlich einem Namen eine IP-Adresse zuzuordnen. Nun tritt aber bei hochfrequentierten Rechnern (z.B.: home.netscape.com) das Problem auf, daß eine Lastverteilung auf andere Rechner unumgänglich ist. Diese Aufgabe kann auch der DNS mit übernehmen, indem er auf eine Anfrage (A-Record) mit einer Reihe von IP-Adressen antwortet. Eine weitere Möglichkeit ist, die IP-Adresse in Abhängigkeit der Anfrageadresse zu variieren.

19. Sketch how a symmetric stream client might work, where either of two processes may contact the other.

```
sd=socket(AF_INET, STEAM_SOCKET,0)          open socket (read data)
sd2=socket( AF_INET, STREAM_SOCKET, 0)      open socket (write date)
bind(sd, ...)
listen(sd, 1)

FD_ZERO(sd_set); FD_SET(sd, &sd_set);       it's necessary for select
for(;;){
      ret=select( sd, sd_set, ..., &timeout) non blocking command !
      if( FD_ISSET(sd,sd_set)  ){           connect arrived - else case do nothing
            new_sd=accept( sd, ...)
            read( new_sd, ...)
            close( new_sd)
      }
      if( there are any data to send )
            new_sd2=connect( sd2, ...)            connect to a socket like sd
            write( new_sd2, ...)
            close( new_sd2)
      }
}
```