Internet: Technology, Protocols and Services

Homework Assignment 2 - ANTWORTEN

1. **One of the TCP options permits a receiver to specify the maximum segment size (MSS) it is willing to accept. Why does TCP support an option to specify maximum segment size when it also has a window advertisement mechanism? (Comer 13.2)**

   Mit der Fenstergröße (WIN) gibt der Empfänger an, wieviel Speicher er für Pakete momentan frei hat. Die MSS wird benutzt, um der anderen Maschine die maximale Paketgröße zu übermitteln, damit die Pakete durch das IP nicht fragmentiert werden müssen.

   MSS = MTU - TCP-*header* - IP-*header*

2. **Lost TCP acknowledgements do not necessarily force retransmission. Explain why. (Comer 13.4)**

   Ein TCP-ACK bestätigt nicht ein empfangenes Paket, sondern gibt das Byte an, welches als nächstes erwartet wird. Damit werden implizit alle Pakete bis zu diesem Byte bestätigt. Ist die Fenstergröße N, dann können N-1 verlorene ACK's durch das folgende rechtzeitig ankommende ACK kompensiert werden. Rechtzeitig heißt in diesem Fall, bevor der Sender ein *timeout* für das verlorene ACK feststellt.

3. **What are the arguments for and against automatically closing idle TCP connections? (Comer 13.8)**

   Nach welcher Zeit ohne Paketverkehr soll man eine TCP-Verbindung als *idle* betrachten? Verbindungen die *idle* sind, benötigen auch *resourcen* (*identifier, squence number, sockets, ...*). Werden unbenutzte Verbindungen sofort abgebaut, dann kann unter Umständen ein baldiger Wiederaufbau nötig sein.

4. **Describe (abnormal) circumstances that can leave one end of a connection in state FIN WAIT 2 indefinitely (hint: think of datagram loss and system crashes). (Comer 13.18)**

   Aus einer vollständig aufgebauten TCP-Verbindung sendet die Maschine_1 ein FIN. Der Partner (Maschine_2) antwortet mit einem ACK. Nun befindet sich die Maschine_1 im Zustand FIN WAIT 2. Wenn jetzt die Maschine_2 gebootet wird, dann verbleibt die erste Maschine in diesem Zustand.

5. **Sketch a UDP echo client that sends a datagram to a specified echo server, awaits a reply and compares it to the original message. (Comer 19.1)**

```
char buf= "The quick brown fox jumps over the lazy dog."
sd=socket(AF_INET, SOCK_DGRAM, 0)
struct sockaddr_in cli_addr, serv_addr

bzero( &cli_addr, sizeof( cli_addr) )
cli_addr.sin_family = AF_INET
cli_addr.sin_port = 0                                    select a free port
cli_addr.sin_addr.s_addr = INADDR_ANY
bind( sd, (sockaddr *) &cli_addr, sizeof( cli_addr) )

bzero( &serv_addr, sizeof( serv_addr) )
serv_addr.sin_family = AF_INET
serv_addr.sin_port = 7                                   TCP/UDP echo port
serv_addr.sin_addr.s_addr = inet_addr( "192.35.149.213")
connect( sd, (sockaddr *) serv_addr, sizeof( serv_addr) )
```

```
for(;;){
       write( sd, buf, sizeof( buf) )
       read( sd, buf1, sizeof( buf) )
       /* compare buf and buf1 */
}
```

6. **When is the *listen* call important? (Comer 20.3)**

   Das *listen*-Kommando verwaltet eine Liste von *connect*-Anfragen an einen Socket (passiv open). Wenn dieser Prozeß (z.B.: durch die Annahme einer anderen Anfrage) nicht im *accept*-Kommando wartet, dann werden die *connect*-Anfragen in die *listen*-Warteschlange aufgenommen und nicht zurückgewiesen.

7. **Sketch the socket-related parts of a finger server. Indicate some of the error conditions that might occur. Assume that the function call *getuserinfo* will return the necessary information from the local system.**

```
make socket, bind, listen
new_sd=accept(...)
fork()
parent: returns to accept
child: if getuserinfo() returns infos
              write infos to new_sd
       else
              write "no such user"
```

8. **GMD Fokus chose to place the following type A resource record entry in its domain name server: `localhost.fokus.gmd.de 127.0.0.1`. Explain what will happen if a remote site tries to *ping* a machine with domain name `localhost.fokus.gmd.de`. (Comer 22.15)**

   Eine DNS-Anfrage nach `localhost.fokus.gmd.de` wird mit `127.0.0.1` beantwortet. Diese Adresse ist aber in jeder Maschine die interne *loop*-Adresse. Ein *ping* auf `127.0.0.1` kontrolliert somit nur die eigene Netzsoftware.

9. **Assume that an MUA is located on an NFS server. It can send encrypted email messages. A corporate spy manages to get an account on a machine on the local Ethernet and wants to intercept messages from the MUA, without cracking the encoding. Any suggestions? (Hint: remember that on an Ethernet, every packet is broadcast. Assume that the spy has access to the source code of the MUA.)**

   Da der Spion Zugang zum *source*-Code des MUA hat, kann er nach seinen eigenen Wünschen einen neuen MUA compilieren. Wenn ein Benutzer den (originalen) MUA starten will, dann wird die Anfrage zum NFS-Server über das *broadcast*-Netz gesendet. Der Spion hat noch ein weiteres Programm geschrieben, welches genau auf diese Anfrage wartet und sie mit dem Senden des modifizierten MUA beantwortet.

10. **In class, we saw a way of spoofing email. How can the receiver detect (most of) these forgeries?**

    In einer Mail wird von jedem MTA eingefügt, von welcher Maschine er diese Nachricht erhalten hat. Anhand dieser Angabe erkennt man sofort, daß Clinton seine Mail auf der lupus bei der GMD Fokus geschrieben haben muß :)

```
Return-path: <clinton@whitehouse.gov>
Delivery-date: Wed, 24 Jan 1996 19:40:00 +0100
Received: from lupus.fokus.gmd.de by ceres.fokus.gmd.de with SMTP (PP-ICR1v5); Wed, 24
Jan 1996 19:39:23 +0100
...
```

    Es ist nicht möglich den Rechnernamen beim HELO zu fälschen:

```
$ hostname
```

```
glenlivet
$ telnet ceres.fokus.gmd.de 25
...
HELO tao.fokus.gmd.de
250 ceres.fokus.gmd.de: You are bluffing - `glenlivet.fokus.gmd.de` expected
```

11. What headers should an email mailing list program change or add? (A mailing list program distributes a single message sent to an email address to a list of receivers.)

Das Programm-`majordomo` (welches bei FOKUS verwendet wird) ändert folgende Einträge: Im SMTP-*envelope* wird der `RCPT To:`-Eintrag, (der an die Mailliste adressiert war) durch solche Einträge für alle Listenbenutzer ersetzt. Zusätzlich wird der `MAIL From:`-Eintrag (ebenfalls *envelope*) auf die Mailliste gesetzt. Im Mailheader werden die Felder `From:` und `To:` nicht verändert. Nur der `Return-Path:` wird durch die Maillistadresse ersetzt und das Feld Sender: `owner-`*mail-list*`@fokus.gmd.de` wird eingefügt.

12. Why should file transport protocols compute a checksum on the file data they receive, even when using a reliable end-to-end stream transfer protocol like TCP? (Comer 24.4)

Die Prüfsumme die im TCP berechnet wird, besteht nur aus der XOR-Verknüpfung aller 16-bit-Wörter. Für den Fall, daß zwei Bits an der gleichen Position verändert werden, kann der Fehler vom TCP nicht erkannt werden. Auch die Vertauschung ganzer 16-bit-Wörter wird nicht bemerkt. Weiterhin ist es möglich, daß in eine bestehende TCP-Verbindung ein falsches TCP-Paket einsortiert wird, das z.B.: aufgrund von Routingproblemen so lange in Netz kursiert ist, bis der übergelaufene *sequence*-Zähler mit dem eines erwarteten Paketes übereinstimmt. Die Berechnung der Prüfsumme ist somit eine wichtige Bedingung, um die übertragene Datei mit dem Original zu vergleichen.

13. What is the chief advantage of using separate TCP connections for FTP control and data transfer? (Hint: think of abnormal conditions) (Comer 24.4)

Aufgrund der Tatsache, daß es für Daten und Kommandos jeweils eine eigene TCP-Verbindung gibt, können ohne Zeitverzug Kommandos übertragen werden. Dadurch kann z.B. eine Datenübertragung jederzeit beendet werden. Die Kommandos müssen nicht durch spezielle Bitmuster von den Daten unterschieden werden.

14. Would you expect TCP or UDP-based NFS to be faster on a local area network (LAN)? Reasons? How about a wide-area network (WAN)?

In LAN's gibt es kaum Paketverlust und die Round Trip Time (RTT) ist klein. UDP-Pakete brauchen keinen Verbindungsaufbau, somit werden weniger Pakete übertragen - das UDP ist schneller als das TCP. Da UDP eine verbindungslose Übertragung ist, können von dem Protokoll unbemerkt Pakete verloren gehen. In WAN's ist die Wahrscheinlichkeit des Paketverlustes und die RTT höher als in LAN's. Verlorene UDP-Pakete müssen durch eine höhere Schicht wiederholte übertragen werden. Da eine schnelle wiederholte Übertragung bei TCP mit eingebaut (fast retransmission) ist, muß TCP für dieses Beispiel schneller sein.

15. Would RPC be suitable for implementing a remote login service? Why or why not?

RPC ist für die Ausführung einzelner Kommandos auf fremden Maschinen konzipiert. Ein *telnet*-Programm muß zwei Datenströme (von und zu dem *remote*-Rechner) übertragen. Ein RPC-Aufruf kann auf einen Befehl nur die direkte Antwort rückübertragen. Treten Daten unabhängig von dem eingegebenen Befehl auf (z.B.: `wall message` oder `talk connect-request`), dann kann der RPC-Aufruf diese nicht direkt übertragen.

16. Write an HTML form for buying pizza on the Internet.

```
<FORM METHOD="POST" ACTION="/cgi-bin/pizza.sh">
PIZZA Service <p>
Your name:    <INPUT NAME="name" SIZE="32"><br>
Your address: <INPUT NAME="address" SIZE="48"><p>
Pizza Funghi     DM 7.97 <INPUT NAME="funghi" VALUE="0" SIZE=5 ><br>
Pizza Mozzarella DM 8.98 <INPUT NAME="mozza"  VALUE="0" SIZE=5 ><br>
...<p>
<INPUT TYPE=SUBMIT> <INPUT TYPE=RESET>
</FORM>
```

17. Propose an extension to HTTP that allows the transfer of several pages in a single TCP connection.

Momentan beendet ein HTTP-Server jede Anfrage selbständig nach dem Ende einer Übertragung. Dadurch erkennt der Client, daß der Server das Dateiende erreicht hat. Um das automatische Schließen umgehen zu können, muß der Client das Dateiende anders erkennen. Eine Möglichkeit ist, die Dateilänge mitzuübertragen.