IK1203
Networks and Communication

Recitation 1 – Introduction and Application Layer

Solutions

1. Short questions:

a) A communication protocol is an agreement between communicating parties defining the rules for communication.

b) Application: HTTP, FTP.

Transport: TCP, UDP.

Network: IP, Routing.

Link: Ethernet, IEEE 802.11 WLAN.

Physical: Bit coding, TP Category 6.

c) The link layer.

d) ”Routing” is about selecting a route (or path) – when the switches (routers) decide which path a packet should take through the network. ”Forwarding” is when a switch (router) redirects a packet from an incoming port to an outgoing port.

e) The transport layer provides the services of transferring data from a process on one computer (host) to a process on another computer (end-to-end delivery).

2. In the first step, the sending e-mail client sends the message via SMTP to the local server for outgoing e-mail (Message Transfer Agent). Second, the e-post message is sent from the server for outgoing e-mail to the recipient’s server for incoming e-mail. This step also uses SMTP. Finally, in the third step, the recipient uses a protocol for e-mail access, such as POP or IMAP, to collect the e-mail message from the server for incoming e-mail.

3.

a) SMTP

b) The communication between the client and the server includes a number of handshakes. In each handshake, the client sends a command to the server and waits for a response. Each handshake takes 2×2.5 ms = 5 ms. A complete transfer to the outgoing server involves the following handshakes::

- ”HELO”, ”MAIL FROM”, ”RCPT TO”, ”DATA”, the actual e-mail message, ”QUIT”

All in all, the transfer takes 30 ms.

c) RCPT TO adds a recipient for the message. Hence, there is one handshake for each recipient. So 45 ms total.

4. Yes, it is possible. According to the SMTP standard, the is solved by replacing any period at the beginning of a line with two periods. The SMTP receiver will change back to a single period.

5. HTTP-request/response:

a) http://www.kth.se/

b) Yes, it was successful. The answer is ”200 OK”.

c) Content-Length gives the size of ”entity-body”, in other words, the data portion od the response: 60044 byte.

d) The client asks for a ”persistent connection”, meaning that the TCP connection should remain open so that the client can send more HTTP requests over the same connection. The server does not accept, and wants to close the TCP connection directly after the HTTP response.

e) The five first characters of the returned object are: ”<html”.

6. *Peer-to-peer protocols are not included in this course round, but here are the solutions anyway.*

Answer the following questions about peer-to-peer protocols.

a) When a sender answers requests from multiple peers, the peers are ranked according to transfer rate. The sender chooses the peers from which it receives file parts (“chunks”) at the highest rates, and sends file parts to them (”tit-for-tat”).

b) When a node fetches file parts from its peers, the node asks the peers what parts they have available at the moment. The nodes gives priority to fetching the parts that are most unsual at the moment (that is, the parts that are at lowest number of peers). This leads to more copies of the rarest parts to be created. It also increases the possibility that the node gets file parts that other peers are missing. So this is good also for “tit-for-tat”, since it makes it more attractive for other peers to exchange file parts with the node.

7. a)

1. The computer on Newton Technologies, “atom.nt.com,” sends a DNS query to the local DNS server.
2. Since the local DNS server has an empty cache, the local DNS server sends the query to a root DNS server.
3. The root DNS server knows the IP address(es) of the the Top-Level Domain (TLD) servers for the top-level domain ”com” and responds with the IP addresses of those servers to the local DNS server.
4. The local DNS server sends the query to one of the TLD servers for the “com” top-level domain.
5. The “.com” TLD DNS server has the IP address to the authoritative DNS server for the domain name ”fws.com”. The TLD DNS server responds with this IP address to the local DNS server.
6. The local DNS server sends the query to the authoritative DNS server for the domain name ”fws.com”. The authoritative DNS server is responsible for all names in this domain, so (most likely) it can answer the question.
7. The authoritative DNS server responds to the local DNS server with the IP address(es) of the host in question.
8. The local DNS server now has the complete answer to the query, and sends this answer back to “atom.nt.com”.

b) DNS query (1) is processed in a recursive way by the local DNS server, while queries (2), (4) are (6) handled iteratively.

c) Yes, there is nothing in DNS that would prevent this.

8. "Dig" is a very useful program for doing DNS queries from the command line. The dig commands for the (valid) queries are included in the table below.

Query description	Query type	Dig command
IP version 4 address for "www.kth.se"	A	dig a www.kth.se
IP version 6 address for "www.kth.se"	AAAA	dig aaaa www.kth.se
TCP port number for HTTP server at "www.kth.se"	–	
Incoming mail server for "kth.se"	MX	dig mx kth.se
Outgoing mail server for "kth.se"	–	
Authoritative name server for "kth.se"	NS	dig ns kth.se
Web server for "kth.se"	–	
Main (canonical) name for alias "www.kth.se"	CNAME	dig cname www.kth.se
Host name with address "130.237.28.40**"**	PTR	dig –x 130.237.28.40 *or* dig ptr 40.28.237.130.in-addr.arpa

Problems from course book (Kurose and Ross, 7th ed)

P2.
At time $N*(L/R)$ the first packet has reached the destination, the second packet is stored in the last router, the third packet is stored in the next-to-last router, etc. At time $N*(L/R) + L/R$, the second packet has reached the destination, the third packet is stored in the last router, etc. Continuing with this logic, we see that at time $N*(L/R) + (P-1)*(L/R) = (N+P-1)*(L/R)$ all packets have reached the destination.

P6.
a. $d_{prop} = m/s$ seconds.
b. $d_{trans} = L/R$ seconds.
c. $d_{end\text{-}to\text{-}end} = d_{prop} + d_{trans} = (m/s + L/R)$ seconds
d. The bit is just leaving Host A.
e. The first bit is in the link and has not reached Host B.
f. The first bit has reached Host B.
g. Want
$m = (L/R)*s = (120/56*10^3)*2.5*10^8 = 536$ km

P7.
Consider the first bit in a packet. Before this bit can be transmitted, all of the bits in the packet must be generated. This requires $56*8/(64*10^3) = 7$ ms.
The time required to transmit the packet is $56*8/(2*10^6) = 224$ µs
Propagation delay = 10 ms.
The delay until decoding is
7 ms + 224 µs + 10 ms = 17.224 ms
A similar analysis shows that all bits experience a delay of 17.224 ms.

P10.
The first bit sent by the host with the lowest propagation delay reaches Router A after 2 ms while the last bit of the packet arrives after $2*10^{-3} + 1500*8/(4*10^6)$ s = 5 ms.
The first bit sent by the host with the highest propagation delay reaches Router A after 6 ms. At that time the packet sent by the other host is already fully received by the router, so no queuing delay occurs.

P11.
Assume $d_1<d_2$. No buffering occurs when $d_2>d_1 + L/R_1$.

P13.

a. The queuing delay is 0 for the first transmitted packet, L/R, for the second transmitted packet, and generally, $(n\text{-}1)*L/R$ for the n^{th} transmitted packet. Thus, the average delay for the N packets is:
   $(L/R + 2L/R + \ldots\ldots. + (N\text{-}1)L/R)/N$
   $= L/(RN)*(1 + 2 + \ldots.. + (N\text{-}1))$
   $= L/(RN) * N(N\text{-}1)2$
   $= LN(N\text{-}1)/(2RN)$
   $= (N\text{-}1)L/(2R)$
   Note that here we used the well-known fact:
   $1 + 2 + \ldots.. + N = N(N+1)/2$
b. It takes NL/R seconds to transmit the N packets. Thus, the buffer is empty when each batch of N packets arrive. Thus, the average delay of a packet across all batches is the average delay within one batch, i.e., $(N\text{-}1)L/2R$.