Examination
IK2218 Protocols and Principles of the Internet
EP2120 Internetworking

Date: 25 October 2014 at 14:00–19:00

a) ***No help material is allowed - You are not allowed to use dictionaries, books, or calculators!***
b) *You may answer questions in English or in Swedish.*
c) *Please answer each question on a separate page (not sheet).*
d) *Please write concise answers!*
e) *Put a mark in the table on the cover page for each question you have addressed.*
f) *The grading of the exam will be completed no later than 17 November 2014.*
g) *After grading, exams will be available for inspection online.*
h) *Deadline for written requests for grading review is 30 November 2014.*
i) *Course responsible IK2218 is Peter Sjödin, phone 08-790 4255.*
j) *Course responsible EP2120 is György Dán, phone 08-790 4253.*

Important note!

Your grade is F in any of these two cases:
- if you do not reach at least 10 (ten) points out of 20 for problems 1-4 or
- if you reach less than 30 points in total.

We advise you to start with problems 1-4.

Part I (Problems 1-4)

1. *IP and addressing (5p)*

You are setting up a local network that will be connected to the Internet through a router. The network should be able to accommodate 125 hosts.

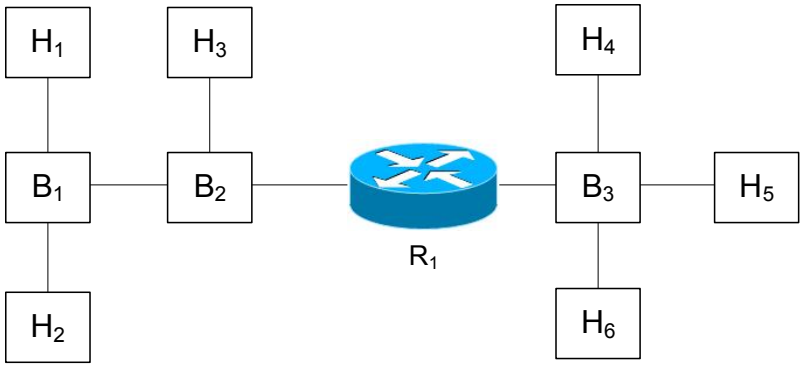
a) What is the smallest network block that would accommodate all hosts? What is the corresponding network mask? (1p)
b) Choose a minimal block of addresses from the address block 192.168.0.0/16 for your network. Specify the network address and the directed broadcast address of your chosen block. (2p)
c) The addresses in the block 192.168.0.0/16 are private IPv4 addresses. Compare private IPv4 addresses to link local IPv6 addresses and to unique local IPv6 addresses in terms of their use. Can a network interface have one of each of these addresses? (1p)
d) Why is the header checksum field not present in the IPv6 header? (1p)

Answers:

a) The network requires 125 addresses for the hosts, 1 network address, 1 broadcast address, and 1 address for the router. Therefore, the smallest network block should contain 128 addresses, which requires 7 bits and gives 32-7=25 bits for the network prefix. The netmask is 255.255.255.128.
b) Any /25 block in the range would be a valid choice, for example 192.168.0.0/25. For this block the network address is 192.168.0.0, the directed broadcast address is 192.168.0.127.
c) Link local addresses are not forwarded by routers, while unique local addresses serve the same purpose as IPv4 private addresses. Yes, a network interface can have one of each of these addresses.
d) Since both transport- and data link-layer protocols compute a checksum, the checksum computed in the network layer can be regarded as redundant. Removing the checksum also reduces the overhead as the IPv4 checksum needs to be recalculated in each router.

2. *Delivery and address resolution (5p)*

Consider an IPv4 network consisting of 6 hosts, 3 bridges and 1 router shown in the figure. Hosts H1 to H6 have one interface each. B1 to B3 are learning bridges. R1 is a router with an appropriate routing table. All ARP caches and the bridges' learning tables are empty.



	MAC	IP
H1	a	A
H2	b	B
H3	c	C
H4	d	D
H5	e	E
H6	f	F
B1	g (north) h (south) x (east)	G (north) H (south) X (east)
B2	i (west) j (north) k (east)	I (east) J (north) K (west)
B3	l (west) m (north) n (east) o (south)	L (west) M (north) N (east) O (south)
R1	p (west) q (east)	P (west) Q (east)

a) Identify the subnets of the network. Which of the physical (MAC) and logical (IP) addresses shown in the table on the left are not needed? (1p)

b) A process on Host H1 sends 100 bytes via UDP to a process on host H4. Using the notation in the table, show the contents of the learning tables and of the ARP caches after the datagram has been delivered. Assume that the process on Host H1 knows the IP address of Host H4, and that ARP snooping is used. (1p)

c) A process on Host H5 sends a message with 200 bytes via UDP to Host H2. Using the notation in the table, show the new contents of the ARP caches and of the learning tables after the datagram has been delivered. Assume that Host H5 knows the IP address of Host H2 and that ARP snooping is used. (1p)

d) How different would the ARP caches and learning tables be in b) and in c) if ARP snooping was not used? (1 p)

e) If a router fails to forward a datagram to the next hop, it usually sends an ICMP destination host unreachable or an ICMP destination network unreachable message to the sender. Name two conditions under which the router does not send an ICMP error message. Briefly explain the reason why an ICMP message is not sent under these two conditions. (1p)

a) Subnet 1: A, B, C, P. Subnet 2: D, E, F, Q. Bridges B1 to B3 do not need MAC addresses and IP addresses.

b) Contents of the ARP caches are as follows.

H1: P-p
H2: A-a
H3: A-a
B1: a-north, p-east
B2: a-west, p-east
B3: q-west, d-north
R1: A-a, D-d
H4: Q-q
H5: Q-q
H6: Q-q

c) New entries in the ARP tables are as follows.

B1: b-south
B2: b-west
B3: e-east
H2: P-p
H3: P-p
R1: E-e, B-b

d) Question b): the ARP caches at H2, H3, H5, and H6 would be empty (0.5p).
Question c): the ARP cache of H3 would not contain P-p as a new entry.
e) The router will not send an ICMP error message to the sender if:
(i) the datagram contains an ICMP error message,
(ii) the destination address is a multicast or a broadcast address,
(iii) datagram sent as link layer broadcast, and
(iv) the datagram's source address is 0.0.0.0.
The reason is:
(i) to avoid loops,
(ii-iii) to avoid broadcast storms, and
(iv) the lack of sender information.

3. *IP forwarding (5p)*

a) A router receives a datagram with an incorrect IPv4 checksum. What will it do? Will the sender be notified? Motivate your answer. (1p)

A router has the IPv4 forwarding table shown below. Determine the next-hop address and the outgoing interface for the packets arriving to the router with destination addresses as given in points (b)-(e).

Destination	Next hop	Flags	Interface
10.180.64.0/18	–	U	m0
172.30.16.0/20	–	U	m1
192.168.200.0/24	–	U	m2
172.30.32.161/32	192.168.200.1	UGH	m2
192.168.17.0/24	172.30.18.77	UG	m1
10.180.128.0/18	10.180.64.253	UG	m0
0.0.0.0/0	192.168.200.2	UG	m2
192.168.7.31/32	192.168.200.2	UGH	m2

b) 172.20.32.161 (1p)
c) 192.168.17.9 (1p)
d) 194.132.196.7 (1p)
e) 10.180.65.100 (1p)

a) The router discards the datagram. Since the router cannot know if the source IP address is correct, it will not report an error.
b) Next hop: 192.168.200.1 on m2
c) 172.30.18.77 on m1
d) 192.168.200.2 on m2
e) direct delivery on m0

4. *TCP (5p)*

a) Consider the following 4-way handshake protocol used for connection establishment between a client, *A*, and a server, *B*. *A* chooses an initial sequence number, ISNA, and sends a segment containing (SYN, seq=ISNA) to *B*. *B* responds with (ACK ISNA+1). *B* then chooses initial sequence number ISNB, and sends a segment (SYN, seq=ISNB) to *A*, to which *A* replies with (ACK ISNB+1). When a node receives the ACK in response to the SYN it sent, it considers the connection to be established. If a node has an established connection on a port

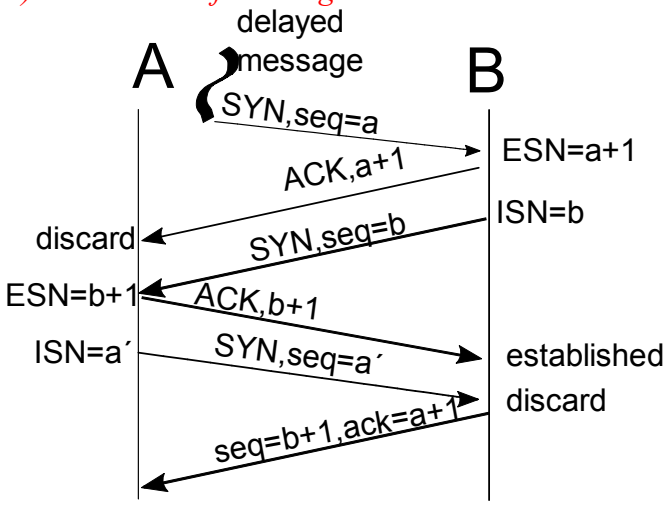
with another host from a particular port, it will not accept a new SYN segment with a different ISN than what it has already acknowledged for the same port-pair.
Show an example of message exchanges that would lead to a deadlock, i.e., using the same ports B could not accept a connection from A, nor A from B. How does TCP connection establishment mitigate such deadlocks? (1p)

Consider two hosts, A and B, connected by a network running IPv6. The capacity of all links is 100Mbps and the round trip time is 300ms. The path MTU is known to be 2060 bytes. A process P_A on host A would like to transmit 30000 bytes to a process P_B on host B using TCP. TCP on the receiving host has a receiver window size limit of 10000 bytes, which it advertises during connection establishment. The sender uses a value of 65535 for *sshthresh* for congestion control. Delayed acknowledgements (two full sized segments) are used with a maximum delay of 200ms. The receiver can process the data as fast as they arrive.

b. What is bandwidth delay product of the channel? How big should the receiver window size be in order to be able to fully utilize the channel (not considering congestion control)? (1 p)
c. Consider that process P_B reads all data from the receive buffer as soon as they arrive. The active open is performed by A. The initial congestion window size is 3xMSS. How much time does it take to transmit the data from A to B including the connection establishment, until the last ACK is received by A? You can ignore the transmission times of the packets, but you should consider the impact of congestion and flow control. If a delayed ACK is to be sent at a time instant when a new segment arrives, the delayed ACK is sent first. Support your solution with a drawing of the segments sent, including the CWND, time, ACKed data, etc. (2p)
d. Consider that at a later point in time, the sender process would like to send 30000 bytes again, but the receiving process reads data slowly, at a rate of approximately 20 byte per second. The initial congestion window size is 3xMSS.What will happen in this case with and without Clarks's solution? Approximately how long will it take TCP to transmit the data (i.e., until the last ACK is received)? You do not need to provide a drawing of the segments sent to support your solution. (1p)

SOLUTION:

a) Consider the following scenario:

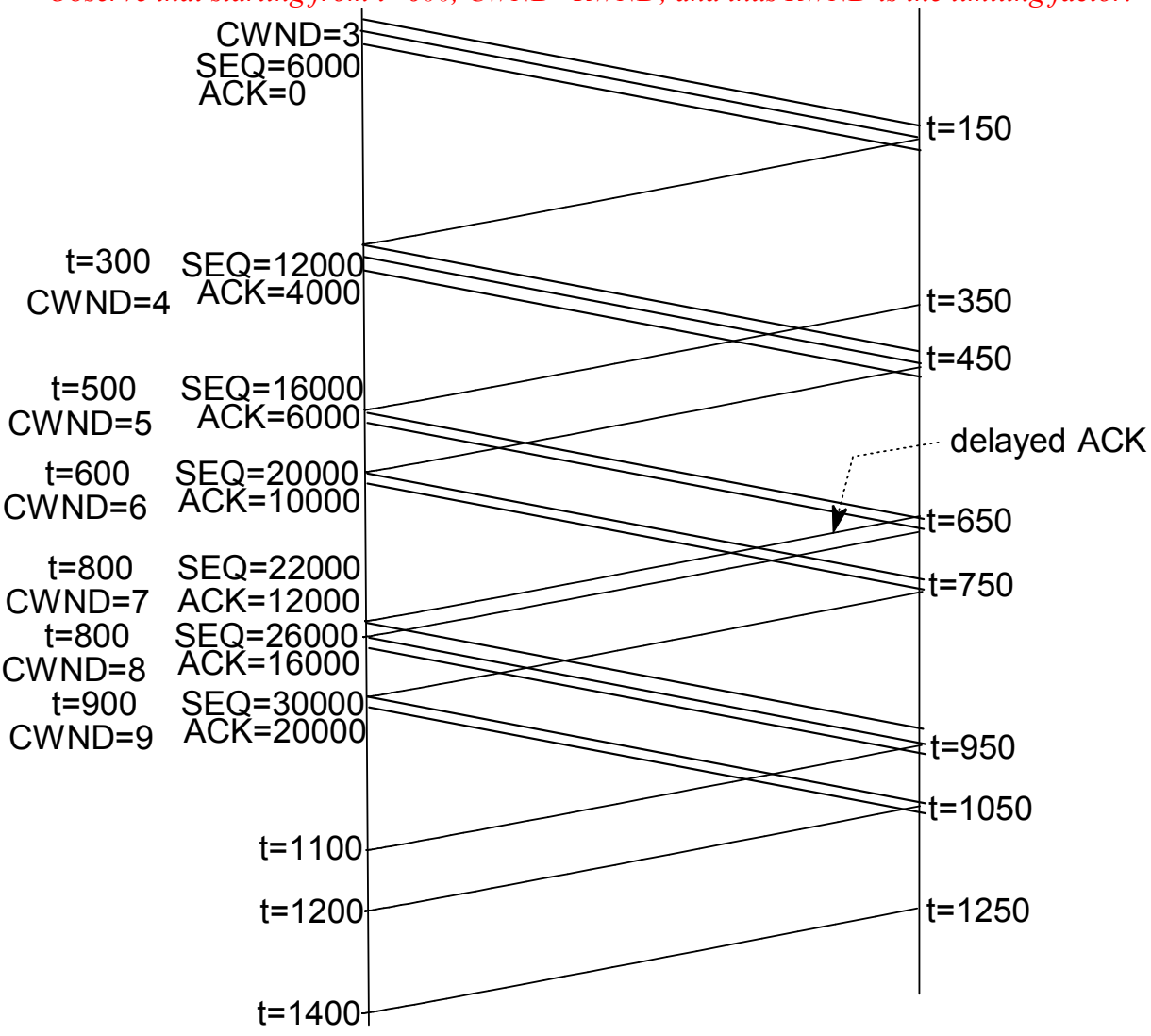


In TCP connection establishment the ACK and the corresponding SYN are piggybacked. Thus, the segment sent by B would be (SYN,seq=b,ACK a+1), which tells to A that the SYN segment was sent in response to a SYN sent with sequence number 'a'. Since A has

already forgotten that sequence number, it would discard the SYN+ACK segment sent by B. Hence no deadlock.

b. The bandwidth delay product is $100*10^6*0.3s=30Mbit$. This is the receiver window size that one would need.

c. The MSS=2000bytes. The connection establishment takes 1RTT, data can be sent after that. In the figure time 0 corresponds to the first data segment sent, connection establishment is not shown. CWND is measured in terms of MSS, SEQ corresponds to the last byte sent, ACK is the last byte acknowledged. In total the transmission takes 1700ms. Observe that starting from t=600, CWND>RWND, and thus RWND is the limiting factor.



*d)Without Clark's solution the receiver would advance the receive window by 20 bytes every second, thus the sender would send segments worth 20 bytes of data. This is the receiver initiated silly window syndrome. TCP will use Clark's solution to avoid the receiver initiated silly window syndrome. The receiver will advance the receive window by 1 MSS every 100s. Note that at the time the receiver has read 20000 bytes of data, the sender is allowed to send the last 10000 bytes of data, which the receiver will acknowledge. It will thus take about 100*10=1000seconds to transmit the data.*

Part II (Problems 5-12)

5. *Fragmentation and UDP (5p)*

Consider an IPv4 network shown in the figure. An application on Host A transmits 2557 bytes of data via UDP to Host B. The UDP header is 8 bytes long, there are no IP options used.

a) Consider that Host A assumes that the path MTU equals the MTU of its directly connected link. How many IP fragments arrive at the router and how many at Host B? Give the segment sizes, the fragmentation offset and the more fragments (MF) bit of all fragments. (2p)

b) Consider now that Host A knows the path MTU. How many IP fragments arrive at the router and how many at Host B? Give the segment sizes, the fragmentation offset and the more fragments (MF) bit of all fragments. (1p)

c) Assume that the last fragment gets delayed in the intermediate router. It arrives 1s before the reassembly timer expires. How does this affect the application? (1p)

d) According to rfc1812 "a (IPv4) router must not reassemble any datagram before forwarding it", and according to rfc2460 an IPv6 router does not perform fragmentation or reassembly. Consider that an IPv6 datagram is sent from the IPv6 capable Host A to the IPv6 capable Host B, and is tunneled over the IPv4 network between the two IPv6/IPv4 capable routers R1 and R2, as shown in the figure below.

The tunneled IPv4 datagram gets fragmented and the fragments are received by Router R2. Should Router R2 reassemble the IPv4 fragments or should it forward the individual fragments to Host B? Motivate your answer. (1p)

Total data to be sent is 2557+8 = 2565 bytes.

a) Unknown path MTU.

The router receives
Fragment 1:1480 bytes IP payload, offset=0, MF=1
Fragment 2: 1085 bytes IP payload, offset=185, MF=0
Host B receives
Fragment 1:976 bytes IP payload, offset=0, MF=1
Fragment 2: 504 bytes IP payload, offset=122, MF=1
Fragment 3: 976 bytes IP payload, offset=185, MF=1
Fragment 4: 109 bytes IP payload, offset=307, MF=0

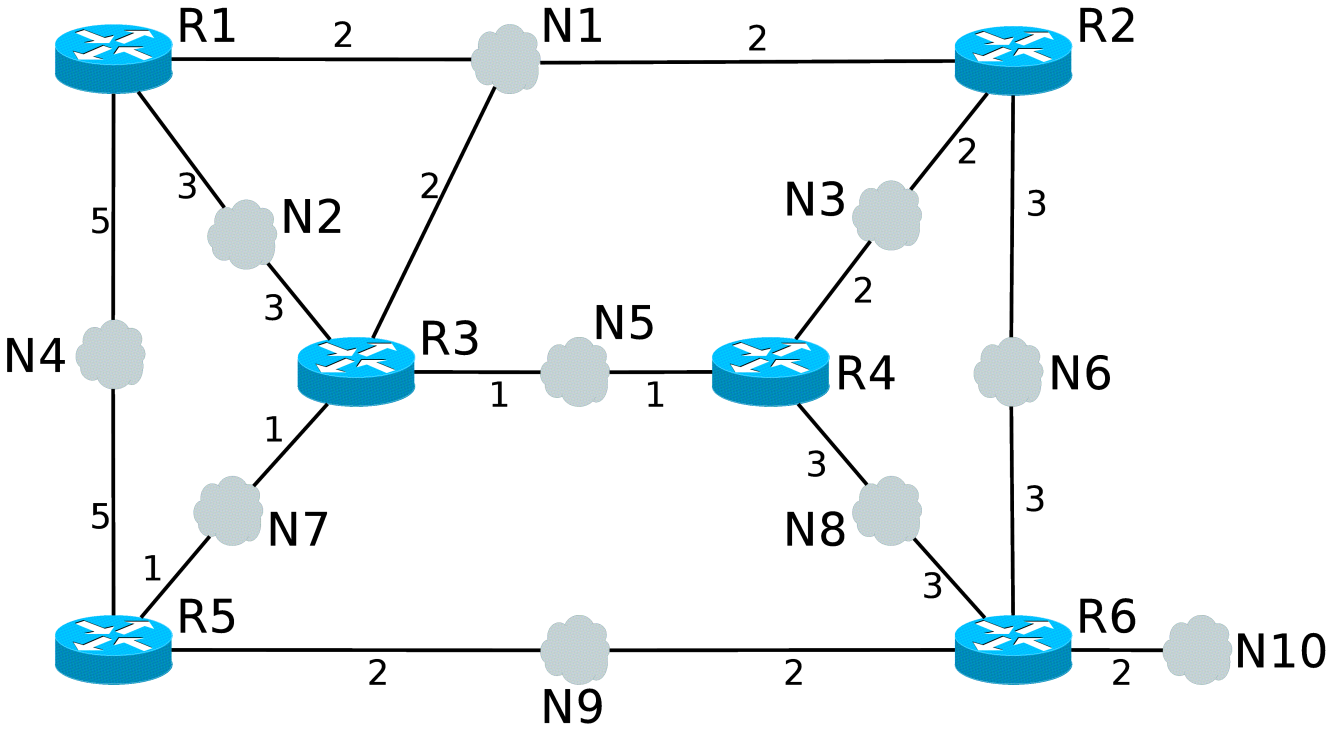
b) Known path MTU (1000 bytes)

Note that 980 is not divisible by 8. Both the router and Host B receive the following fragments.
Fragment 1: 976 bytes IP payload, offset=0, MF=1
Fragment 2: 976 bytes IP payload, offset=122, MF=1
Fragment 3: 613 bytes IP payload, offset=244, MF=0

c) Reassembly occurs at the IP layer, so the application will only notice some delay in the delivery of the packet.
d) Router R2 must reassemble the original IPv4 datagram. Only the first fragment contains the IPv6 header, thus router R2 would not be able to forward the subsequent fragments.

6. *Routing (5p)*

Consider the internetwork shown in the figure below. R1 to R6 are routers, N1 to N10 are networks. Numbers along the links represent the cost of sending a datagram from a router to



the adjacent network.

a) Use Dijkstra's algorithm to find the shortest paths from router *R1* to all networks in the internetwork. The result should provide the cost and next hop(s) for each destination. To ease notation, please use R1N1 to refer to the IP address of Router R1 on network N1, etc. (3p)

b) What is equal cost multipath? Give an example from the solution of (a). (1p)

c) Distance vector routing suffers from the count-to-infinity problem. Given an example for how count-to-infinity would happen in a baseline distance vector routing protocol. Name a solution that RIPv2 implements to mitigate count-to-infinity and show how it works on the example. (1p)

Solution:
a) Start with an empty tentative set and a permanent set containing the route to R1:
P: **<R1,0,->**
T:
Add neighbours of R1 to T:
P: <R1,0,->
T: **<N1,2,->**, <N2,3,->, <N4,5,->
Lowest cost is to N1, add N1 to P and N1's neighbours to T:

T: <N2,3,->, <N4,5,->, **<R2,2,->**, **<R3,2,->**
Lowest cost (=2) is to R2 and R3, add them to P and their neighbors to T:
T: <N2,3,->, <N4,5,->, <N3,4,R2N1>, <N6,5,R2N1>, <N2,5,R3N1>, <N5,3,R3N1>, <N7,3,R3N1>
Keep the shorter path to N2 (and sort T for readability):
T: **<N2,3,->**, <N3,4,R2N1>, <N4,5,->, **<N5,3,R3N1>**, <N6,5,R2N1>, **<N7,3,R3N1>**
Lowest cost (=3) is to N2, N5 and N7, add them to P and their neighbors to T:
T: <N3,4,R2N1>, <N4,5,->, <N6,5,R2N1>, **<R4,3,R3N1>**, **<R5,3,R3N1>**
Lowest cost (=3) is to R4 and R5, add them to P and their neighbors to T:
T: <N3,4,R2N1>, <N4,5,->, <N6,5,R2N1>, <N3,5,R3N1-R4N5>, <N8,6,R3N1-R4N5>, <N4,8,R3N1-R5N7>, <N9,5,R3N1-R5N7>
Keep the shorter path to N3 and N4 (and sort T for readability):
T: **<N3,4,R2N1>**, <N4,5,->, <N6,5,R2N1>, <N8,6,R3N1-R4N5>, <N9,5,R3N1-R5N7>
Lowest cost (=4) is to N3, add it to P, N3 has no neighbors that are not in P, hence nothing to add to P:
T: **<N4,5,->**, **<N6,5,R2N1>**, <N8,6,R3N1-R4N5>, **<N9,5,R3N1 >**
Lowest cost (=5) is to N4, N6 and N9, add them to P and their neighbors to T:
T: <N8,6,R3N1-R4N5>, **<R6,5,R2N1>**, **<R6,5,R3N1 >** (two equal cost routes to R6)
Lowest cost (=5) is to R6, add it to P and its neighbors to T:
T: **<N8,6,R3N1 >**, <N10,7,R2N1>, <N10,7,R3N1-R5N7> (two equal cost routes to N10)
Lowest cost is to N8, add it to P, N8 has no neighbors that are not in P.
T: **<N10,7,R2N1>**, **<N10,7,R3N1 >** (two equal cost routes to N10)
Finally add N10 to P:
P: <R1,0,->, <R2,2,->, <R3,2,->, <R4,3,R3N1>, <R5,3,R3N1>, <R6,5,R2N1>, <R6,5,R3N1 >, <N1,2,->, <N2,3,->, <N3,4,R2N1>, <N4,5,->, <N5,3,R3N1>, <N6,5,R2N1>, <N7,3,R3N1>, <N8,6,R3N1 >, <N9,5,R3N1 >, <N10,7,R2N1>, <N10,7,R3N1 >
T:

b) Equal cost multipath maintains two or more paths to a single destination if they have the same cost. The different paths can be used, e.g., for load balancing. There are two paths with cost 7 to N10.

c) A simple example is the one given on the lecture with two routers R1 and R2, R1 connected to a network N1. Split horizon, split horizon with poison reverse, triggered updates are used in RIPv2. See lecture slides for how they mitigate the issue.

7. *DNS (6 p)*

Study the following excerpt from a DNS zone file on a DNS server.

```
$ORIGIN     firefly.org.
$TTL        86400
@           IN    SOA    ns.firefly.org. hostmaster.firefly.org.(
                         2014102501
                         10800
                         3600
                         604800
                         3600 )

                         IN  NS         ns.firefly.org.
                         IN  NS         ns2.firefly.org.
```

```
ns                      IN  A       198.16.12.5
ns2                     IN  A       212.16.12.2
chicolini               IN  A       198.16.12.139
                        IN  AAAA    2001:6182::beef
rufus                   IN  NS      ns.rufus.firefly.org.
                        IN  NS      ns2.rufus.firefly.org.
ns.rufus                IN  A       198.16.13.9
ns2.rufus               IN  A       192.16.13.23
                        IN  AAAA    2001:6b01::feed
pinky                   IN  CNAME   chicolini
```

a) For which domain is this server an authoritative server? Give the answer as an FQDN (Fully Qualified Domain Name). (1 p)
b) Can you conclude the IP address (or addresses) of the server by studying the zone file? If so, give the address(es). (1 p)
c) You make an attempt to resolve the name `pinky.firefly.org`. What address (or addresses) will the name be translated to, if any? (1 p)
d) The zone file contains a delegation of a sub-zone. What is the sub-zone that is being delegated (answer with an FQDN)? What can you tell about the name server (or servers) for that sub-zone? Give name(s) and IP address(es), if possible. (1 p)
e) The sub-zone is defined by a separate zone file, which resides on a different name server. What can you learn about the content of that zone file, by studying the above zone file? Describe the records you can conclude should be there. (2 p)

Solution

a) firefly.org. (With a dot '.' at the end, otherwise it is not an FQDN).

b) It is likely to be one of the authoritative name servers for the domain, that is, "ns. firefly.org" or "ns2. firefly.org". Hence, according to the zone file, its IP address is 198.16.12.5 or 212.16.12.2. (It is enough to answer with one IP address.)

c) It will finally be translated to chicolini's IP addresses: 198.16.12.139 and 2001:6182::beef .

d) The delegated zone is "rufus.firefly.org." (dot at the end). There are two name servers. Their names are "ns.rufus.firefly.org." (IP address 198.16.13.9) and "ns2.rufus.firefly.org." (IP addresses 192.16.13.23 and 2001:6b01::feed)

e) It has a "SOA" record for the domain "rufus.firefly.org." and should contain "NS" records for the two authoritative name servers ""ns.rufus.firefly.org." and "ns2.rufus.firefly.org.". Moreover, it should contain "A" address records for the IP addresses for the two authoritative name servers. There is probably more as well, but nothing you can infer from the zone file in this problem.

8. *Electronic Mail (3 p)*

Suppose that you are designing a "vacation" message service – a program that will automatically send responses to emails addressed to a certain user. The message service should only respond to emails addressed directly to the user (and not to emails on mailing lists, for instance).

For simplicity, assume that you are implementing this as standalone service. In other words, your program's only purpose is to send automatic responses, and it does not need to be integrated in any way with other mail functions. (It needs to be able to send and receive emails, of course.)

a) Where would your vacation service run? As a user agent (mail client), outgoing mail server or incoming mail server? Explain! (1 p)
b) The course textbook (Forouzan, TCP/IP Protocol Suite) divides protocols that define the format for email into two parts: envelope (SMTP, Simple Mail Transfer Protocol) and message (RFC 5322 (formerly RFC 822), MIME, Multi-purpose Internet Mail Extensions, and more). Your program may need to implement one or more of the three mentioned protocols (SMTP, MIME, and RFC 5322/822). For each of the three protocols, explain whether or not you need to implement it. (2 p)

Solution

a) It would run as an incoming mail server, which is always active. It could potentially also run as a user agent, although then an explanation would be required that it needs to run on a computer that is online so that the program can periodically check the incoming mail server for new mail.

b) Assuming that you implement the service as an incoming mail server, you need to implement SMTP in order to send and receive emails. Furthermore, you need to implement RFC 5322 in order to parse the mail header to find out who the recipient is. You do not need to implement MIME, since MIME is about mail content, which your program does not need to process.

9. *Web and HTTP (5 p)*

A Web page consists of multiple objects. To provide a good user experience, Web browsers try to present pages as quickly as possible, and fetch all objects with minimum delay. Consider the case with a Web page where all objects are stored on the same server. You are viewing the page using a regular browser on an ordinary computer. The browser could use different strategies for loading the page:

1. Open multiple TCP connections in parallel, one for each object.
2. Use a single persistent TCP connection, and fetch the objects one at a time over the connection: Send HTTP request for the first object; wait for response; send HTTP request for second object; wait for response; etc.
3. Use a single persistent TCP connection, in combination with HTTP pipelining where multiple HTTP requests are sent after each other, without waiting for the corresponding responses. In other words, the browser sends all HTTP requests, and then waits for the responses.

a) Explain how you would rank the three alternatives according to how fast they will load the entire page, from fastest to slowest. (2 p)
b) Explain how you would rank the three alternatives according to how much load they would put on the Web server, from highest to lowest load. (2 p)
c) HTTP is stateless, which means that an HTTP transfer consists of a single request-response transaction. This creates a problem in combination with HTTP pipelining, since HTTP pipelining involves several HTTP transactions at the same time and therefore there needs to be a way to match HTTP responses with the corresponding requests. How do you think HTTP requests and responses could be matched with each other with HTTP pipelining? Explain! (1 p)

Assume that all objects are immediately available at the server, without any delay. So as soon as the server receives an HTTP request, it can send the HTTP response.

Moreover, the client and the server are regular computers, each with one network port connected to the Internet.

It is important that you explain your solution. You may want to draw time diagrams, for instance.

Solution

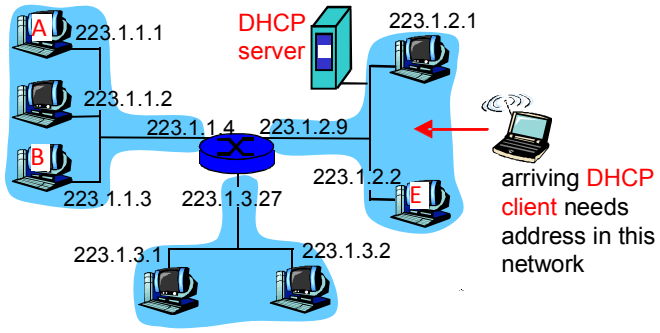
a) The solution is not entirely obvious, so the score depends on your explanation. A realistic answer is that number 3 (Persistent HTTP with HTTP pipelining) is the fastest, then comes number 1 (multiple TCP connections in parallel), and finally number 2. One might think that multiple TCP connections is the fastest, but it requires more packets to be transmitted, and in practice the TCP connections will not be truly parallel, since all packets will take the same path between browser and server. In particular, all packets will go through the same network ports. Therefore, packets will be sent sequentially. Hence, The TCP SYNs will be sent sequentially, and the SYN-ACKs will arrive sequentially. The same goes for HTTP requests, which will be sent sequentially, as well as the responses. There will also be some processing delay at the server for setting up each TCP connection.

b) The alternative with multiple TCP connections puts the highest load on the server, since each TCP connection requires resources and processing time. The two alternatives with persistent HTTP should be equally costly, from a processing time point of view.

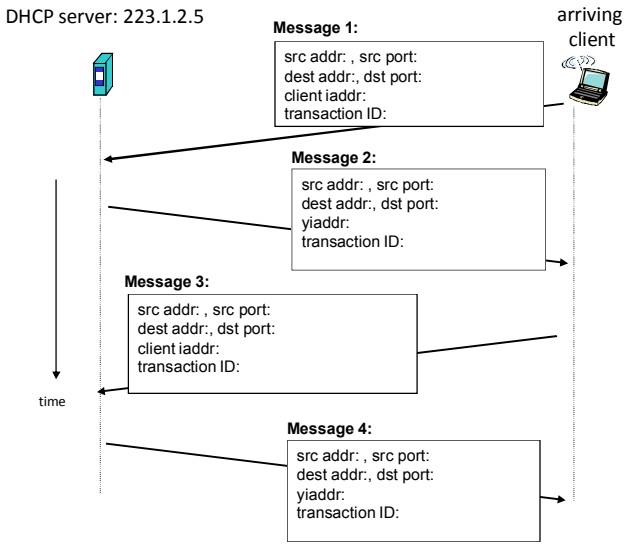
c) An HTTP response does not contain any information that identifies the corresponding request (or the object contained in the response). Hence, the requirement with HTTP pipelining is that a server must return HTTP responses in the same order as the requests arrived.

10. *Autoconfiguration (4p)*

a) Consider the following scenario, where a DHCP client arrives and requests an IP address from the DCHP server.



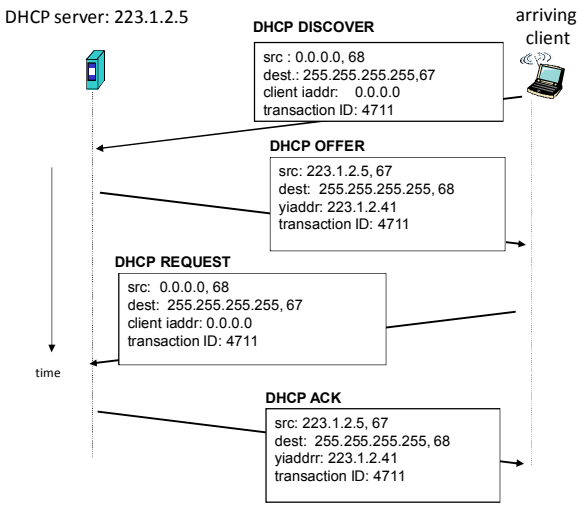
In the simplest case, four DHCP messages will be exchanged according to the figure below. Name these four DHCP messages (message type) and fill in the missing fields in each message. You can assume that the subnet to which the DHCP client arrives is a /24 network and that all addresses below 223.1.2.10 are occupied. Based on that, you can let the DHCP server hand out a suitable IP address. You also have to select reasonable transaction IDs. (2p)



b) Stateless autoconfiguration (SLAAC) was introduced with IPv6. How does the host obtain its global IP address with SLAAC and in what sense is SLAAC stateless? (2p)

SOLUTION

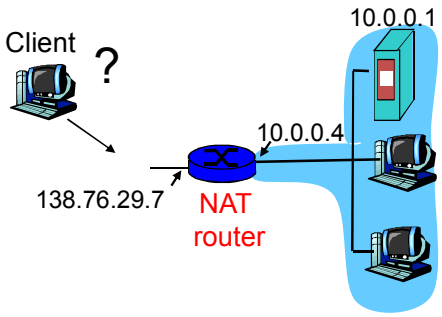
a) Valid IP addresses for yiaddr from the server are 223.1.2.10-223.1.2.254. Transaction ID should be the same value for all four messages. There are some implementations that use a new value in DHCP REQUEST and DHCP ACK, so such a solution to this problem is also OK.



b) It means that the server does not have to keep host-specific information, only non-host state information. The server only keeps track of information like global prefix and subnet prefix. In IPv6, hosts can generate a link local IP address without involving a server. It can then use ICMP (Router Advertisements) to get global and subnet prefixes to form a unique global address.

11. *IP Gateways—NATs and firewalls (6p)*

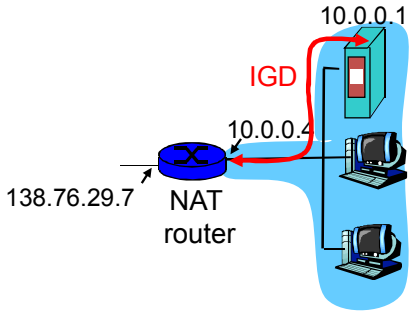
a) Consider the figure below. Assume that you have a server running on host 10.0.0.1 in your private network behind the NAT box and that you want to make it visible to clients outside your private network. Describe how UPnP (Universal Plug and Play) could help you solving this problem. (4p)



b) Assume now that the server you run on 10.0.0.1 uses TCP. Assume further that the NAT router in the figure above also is a packet filtering firewall and that you configure a filtering rule blocking incoming TCP segments with ACK=0. How will that affect the external client's possibility to reach the server? Explain your answer. (2p)

SOLUTION

a) The Internet Gateway Device Protocol (IGD Protocol) is implemented via UPnP. Many routers and firewalls expose themselves as Internet Gateway Devices, allowing any local UPnP control point to perform a variety of actions, including retrieving the external IP address of the device, enumerate existing port mappings, and add or remove port mappings. By adding a port mapping, a UPnP controller behind the IGD can enable traversal of the IGD from an external address to an internal host.



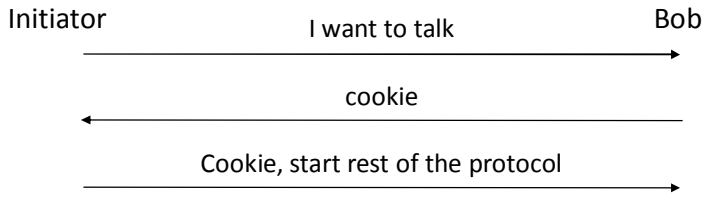
b) This filter will prevent the external client from making TCP connections to the internal server. So even though you have solved the NAT traversal problem, the server will not be reachable from the outside as intended. The TCP ACK flag is set on all but the first packet, the one that establishes the connection so the filter will block the incoming TCP connection establishment segment.

12. *IPsec and IKE (6p)*

a) Briefly summarize the security services that IPsec provides. (2p)

b) The packet below illustrates an IPv6 packet protected using IPsec. Is it tunnel mode or transport mode? Redraw the packet and show how the different parts of the packet can be protected using the services you described in your solution for a) above. (2p)

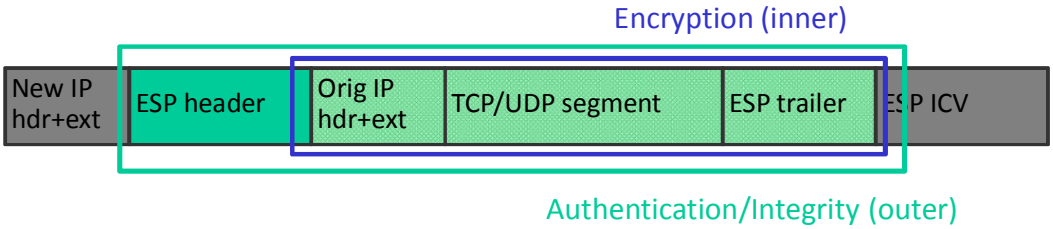
c) The simple illustration below shows how cookies are used in IKE, with the purpose to protect against denial-of-service attacks where an impostor launches packets with forged IP source addresses. The cookies should be *stateless*. What does this mean and how is it achieved in IKE? (2p)



SOLUTION

a) Authentication—IPsec can verify that the sender really is the one it claims to be
Integrity—IPsec can detect if data is modified in transit
Confidentiality—IPsec can encrypt data

b) This is tunnel mode since a new IP header is added outside the original header.



c) When Bob has sent a cookie to the initiator, he will not continue the execution until he receives the same cookie from the initiator. In a DoS attack, there could be a large amount of fake initiators and the cookies should be stateless so that "Bob" doesn't have to keep track of all cookies he has sent. A stateless cookie can be created by doing a hash of the IP address and a single secret number that Bob uses:
hash(IP addr, Bob's secret)