Networks and Protocols
Jacobs University Bremen
Dr. Jürgen Schönwälder

Course: 320301
Date: 2011-12-10
Duration: 120 minutes

Final Examination

The Jacobs University's Code of Academic Integrity applies to this examination. Please fill in your name (please write readable) and sign below.

Name:	
Signature:	

This exam is **closed book**. In addition, you are not allowed to use any electronic equipment such as computers, smart phones, cell phones, or calculators.

Please answer the questions on the problem sheets. If you need more space, feel free to write on the back of the pages. Please keep the papers stapled.

Problem	Max. Points	Points	Grader
F.1	10		
F.2	20		
F.3	20		
F.4	20		
F.5	10		
F.6	10		
F.7	10		
Total	100		

Good luck and have a relaxing winter break!

Problem F.1: *general questions* (2+2+2+2+2 = 10 points)

Indicate which of the following statements are correct or incorrect by marking the appropriate boxes. For every correctly marked box, you will earn two points. For every incorrectly marked box, you will loose one point. Statements which are not marked or which are marked as both true and false will be ignored. The minimum number of points you can achieve is zero.

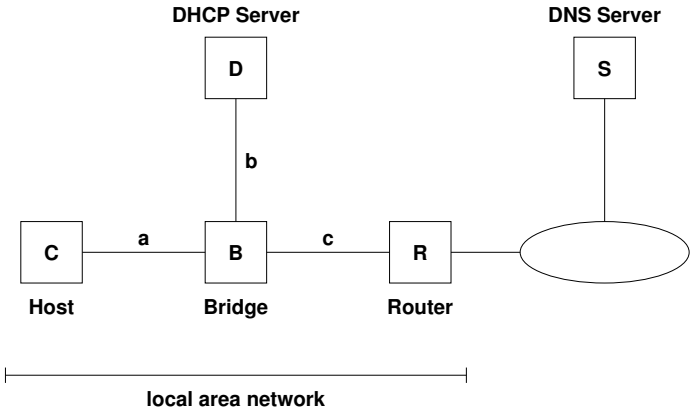
true	false	
☐	☐	Contention-based media access control protocols provide statistical performance guarantees.
☐	☐	Differential encoding schemes allow a receiver to synchronize with the speed of the sender (self-clocking).
☐	☐	Frequency division multiplexing applied to optical fibers is also called wavelength division multiplexing.
☐	☐	The two IPv4 prefixes 203.0.113.13/25 and 203.0.113.130/25 can be aggregated into the prefix 203.0.113.0/24.
☐	☐	The usage of a network address as part of the transport address instead of having a proper address mapping mechanism is an architectural problem making, for example, network mobility support much more complicated than normally needed.

Solution:

true	false	
☒	☐	Contention-based media access control protocols provide statistical performance guarantees.
☐	☒	Differential encoding schemes allow a receiver to synchronize with the speed of the sender (self-clocking).
☒	☐	Frequency division multiplexing applied to optical fibers is also called wavelength division multiplexing.
☒	☐	The two IPv4 prefixes 203.0.113.13/25 and 203.0.113.130/25 can be aggregated into the prefix 203.0.113.0/24.
☒	☐	The usage of a network address as part of the transport address instead of having a proper address mapping mechanism is an architectural problem making, for example, network mobility support much more complicated than normally needed.

Problem F.2: *routers and bridges and dhcp/dns* (4+2+2+12 = 20 points)

Consider the network topology shown below. The host C, which has never been part of this network before, has been connected to the transparent bridge B and it did just initialize. The host C now tries to obtain an IP address from the DHCP server D and once that has been successfully completed, the host C sends a DNS request via the router R to the DNS server S. The DNS server returns a response to C. The spanning tree protocol is not used in the local area network (since the topology is cycle-free). All IP packets exchanged fit into the maximum frame size supported by the links (no fragmentation).



a) What is the DHCP message exchange taking place between the host C and the DHCP Server D? Also mention (for each DHCP message) if it's a broadcast/unicast message.

b) What is the advantage of sending the `DHCP_REQUEST` message as a broadcast?

c) Which configuration information must the exchange include for the client to work correctly?

d) List the frames that are transmitted over the various segments of the local area network. Produce a table like this:

no	segments	eth-src	eth-dst	ip-src	ip-dst	description

Please denote the MAC address of a host H with $mac(H)$ and the IP address of a host H with $ip(H)$. Use $mac(*)$ and $ip(*)$ for layer two and layer three broadcast addresses. In case a node may have different MAC and IP addresses for different network segments, use the notation $mac(H_S)$ and $ip(H_S)$ to refer to the MAC address or the IP address of host H on segment S.

Solution:

a) The following DHCP exchange takes place:

Source	Destination	Message
C	$*$	DHCP_DISCOVER (broadcast)
D	C	DHCP_OFFER (unicast)
C	$*$	DHCP_REQUEST (broadcast)
D	C	DHCP_ACK (unicast)

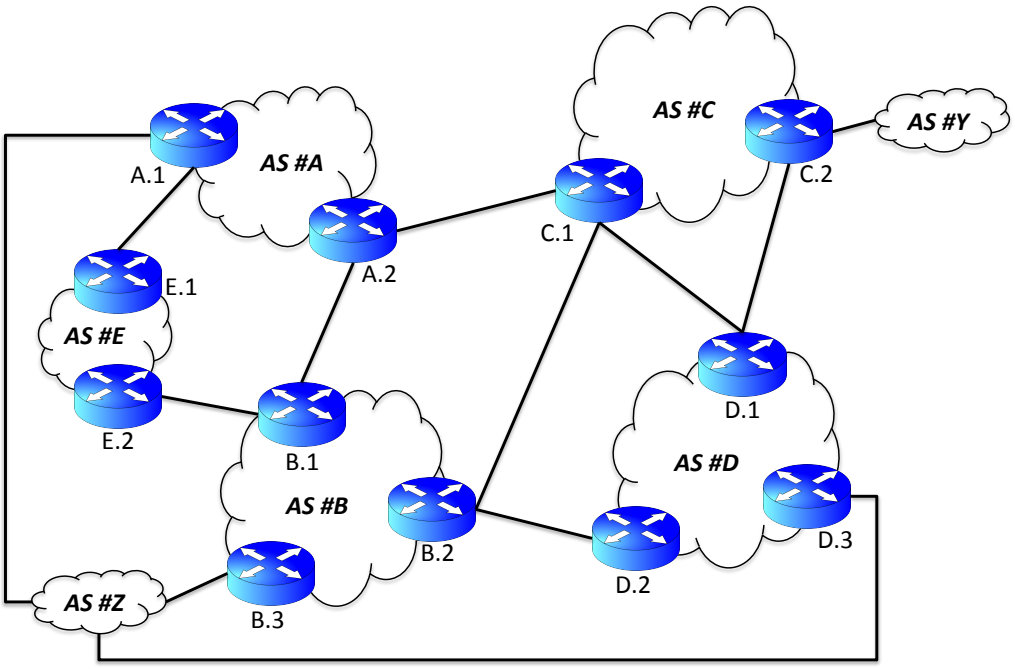
b) The DHCP_REQUEST is sent as a broadcast because (a) the client does not yet have an IP address allocated and (b) by broadcasting the DHCP_REQUEST it is possible to inform other DHCP servers that might have provided an offer that their offer was not chosen.

c) The DHCP exchange must include the router DHCP option so that the host C learns about the default router R. In addition, the DHCP exchange should include the option that conveys the DNS server's IP address to the client C.

d) The following frames are transmitted in the local area network:

no	segments	eth-src	eth-dst	ip-src	ip-dst	description
1	a,b,c	$mac(C)$	$mac(*)$	-	-	DHCP_DISCOVER
2	b,a	$mac(D)$	$mac(C)$	-	-	DHCP_OFFER
3	a,b	$mac(C)$	$mac(*)$	-	-	DHCP_REQUEST
4	b,a	$mac(D)$	$mac(C)$	-	-	DHCP_ACK
5	a,b,c	$mac(C)$	$mac(*)$	-	-	ARP: Who has $ip(R_c)$?
6	c,a	$mac(R)$	$mac(C)$	-	-	ARP: $ip(R_c)$ is at $mac(R_c)$
7	a,c	$mac(C)$	$mac(R)$	$ip(C)$	$ip(S)$	DNS query
8	c,a	$mac(R)$	$mac(C)$	$ip(S)$	$ip(C)$	DNS reponse

Problem F.3: *BGP routing and policies* (6+4+4+6 = 20 points)



Consider the above network scenario with multiple autonomous systems, *AS #A...#E*. Each autonomous system has a number of border routers that are used to peer with neighboring autonomous systems; for example, *AS #A* has two routers *A.1* and *A.2*. The BGP routing protocol is used in order to setup the routing links between each autonomous system.

Additionally, *AS #A*, *#B* and *#D* are peering with *AS #Z*. Similarly, *AS #C* is peering with *AS #Y*. Using this information, answer the following questions about this network.

a) Suppose that a BGP router *Z.1* within *AS #Z* has learned the following potential paths to a destination that lies within AS #Y.

Learnt From	Path
A.1	A.1→ A.2 → C.1 → C.2→ Y
B.3	B.3 → B.2 → C.1 → C.2 → Y
D.3	D.3 → D.2 → B.2 → C.1 → C.2 → Y

The following events occur within the network once the above routing information has been received by *Z.1*.

Time	Action
$t = 0s$	Z.1 picks path from A.1 as best path and announces it to Z.2
$t = 5s$	Link A.2 → C.1 fails
$t = 8s$	A.1 withdraws path (A.1→ A.2 → C.1 → C.2→ Y)
$t = 12s$	D.3 withdraws path (D.3 → D.2 → B.2 → C.1 → C.2 → Y)
$t = 22s$	Link B.2 → C.1 fails
$t = 31s$	D.3 announces path (D.3 → D.1 → C.2 → Y)
$t = 34s$	B.3 withdraws path (B.3 → B.2 → C.1 → C.2 → Y)
$t = 47s$	A.1 announces path (A.1 → A.2 → B.1 → B.2 → D.2→ D.1→ C.2 → Y)
$t = 54s$	B.3 announces path (B.3 → B.1 → A.2 → C.1 → C.2→ Y)

Assuming that a 15 second minimum routing advertisement interval (MRAI) timer is used and that *Z.1* always picks the shortest path, list the times at which node *Z.1* sends updates to a node *Z.2* along with the announced paths to AS #Y. You may use the following format and only need to list up to $t = 60s$:

Time	Path to AS #Y announced by Z.1 to Z.2
$t = 0s$	Z.1→ A.1 → A.2 → C.1 → C.2→ Y

b) What effect would the following BGP filtering policy have in the network shown in the image? Explain how the filter directions (in/out) work and the direction of traffic flows relate to BGP announcements.

```
bgp router C.1 peer D.1
  filter out
    add-rule
      match any
      action "metric 10"
      exit
    exit
  exit
```

```
bgp router C.2 peer D.1
  filter out
    add-rule
      match any
      action "metric 5"
      exit
    exit
  exit
```

c) What effect would the following BGP filtering policy have in the network shown in the image? What effect would changing `local-pref` to 200 on link between *A.2* and *C.1* have?

```
bgp router A.2 peer B.1
  filter in
    add-rule
      match any
      action "local-pref 150"
      exit
    exit
  exit
```

```
bgp router A.2 peer C.1
  filter in
    add-rule
      match any
      action "local-pref 100"
      exit
    exit
  exit
```

d) What effect would the following BGP filtering policy have in the network shown in the image? How does it achieve this behavior?

```
bgp router E.2 peer B.1
  filter in
    add-rule
      match any
      action "community add 1"
      exit
    exit
  exit

bgp router E.1 peer A.1
  filter out
    add-rule
      match "community is 1"
      action deny
      exit
    exit
  exit
```

```
bgp router E.2 peer B.1
  filter out
    add-rule
      match any
      action deny
      exit
    exit
  exit

bgp router E.1 peer A.1
   filter out
    add-rule
      match any
      action "community strip"
      exit
    exit
  exit
```

Solution:

a) Path announced by Z.1 to Z.2:

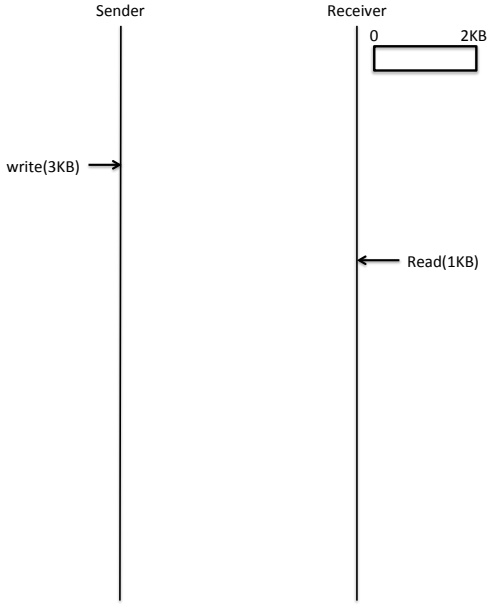
Time	Path to AS #Y announced by Z.1 to Z.2
$t = 0s$	Z.1$\rightarrow$ A.1 $\rightarrow$ A.2 $\rightarrow$ C.1 $\rightarrow$ C.2$\rightarrow$ Y
$t = 15s$	Z.1$\rightarrow$ B.3 $\rightarrow$ B.2 $\rightarrow$ C.1 $\rightarrow$ C.2 $\rightarrow$ Y
$t = 30s$	Z.1$\rightarrow$ B.3 $\rightarrow$ B.2 $\rightarrow$ C.1 $\rightarrow$ C.2 $\rightarrow$ Y
$t = 45s$	Z.1$\rightarrow$ D.3 $\rightarrow$ D.1 $\rightarrow$ C.2 $\rightarrow$ Y
$t = 60s$	Z.1$\rightarrow$ D.3 $\rightarrow$ D.1 $\rightarrow$ C.2 $\rightarrow$ Y

b) The policy causes traffic from AS #D to flow over the link C.2 $\rightarrow$ D.1 since this link has the lower metric. `filter out` is used to specify the direction of the filter. In this case, the filter is an output filter, i.e., it is applied to routes sent to the peer D.1.

c) The policy causes traffic from AS #A to AS #D to flow over the link A.2 $\rightarrow$ B.1 by increasing its preference within the local network. Changing `local-pref` of this link to 200 will cause the link between A.2 $\rightarrow$ C.1 to become the preferred link instead.

d) This policy causes the link E.2 $\rightarrow$ B.1 to only carry traffic originating from AS #E. This is achieved by tagging all routes announced by AS #B with the community 1 (first rule). The second rule ensures that no routes are announced to B.1 and hence AS #B will not move traffic to this link. The last two rules ensure that any routes learned from B.1, that have been tagged with community 1, are not propagated to A.1.

Problem F.4: *tranmission control protocol* (10+10 = 20 points)

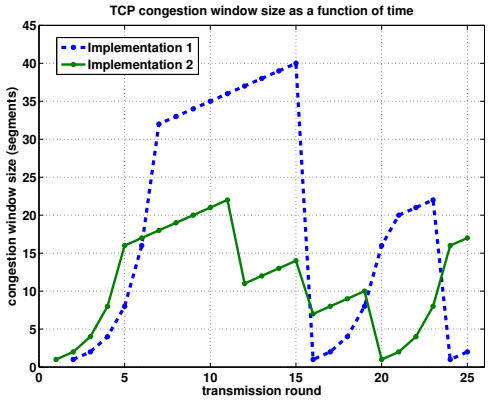


Consider the scenario shown above. The receiver has a 2KB receive buffer. The sender performs a single write operation of 3KB data (1KB = 1024B). The sender sends data as soon as possible, delays are short and there is no congestion between the sender and the receiver.

a) Use the above image to show the TCP exchange that takes place between the sender and the receiver, including the connection establishment and tear-down procedure. For each segment sent, you must show the appropriate segment size, window size and sequence number, if any; for any SYNs, FINs and ACKs sent, you must show their respective number.

   Also mark any period during which the sender might be blocked from sending more data.

b) The following plot shows the TCP congestion window size as a function of time for two different TCP implementations.



Answer the following questions regarding both implementations:

i) What is the initial threshold for the first transmission round? Why?

ii) What is the threshold at the 17th transmission round? Why?

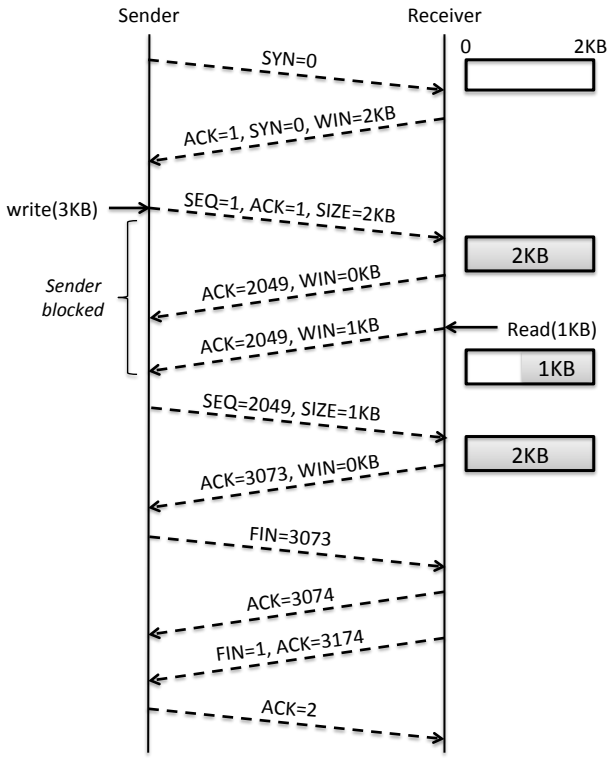
iii) What would the new threshold have been in case congestion was detected at round 8?

iv) During which transmission round is the 50th segment sent? Explain.

v) Which of the two implementations supports fast retransmit / fast recovery? Explain.

Solution:

a) TCP exchange:



b) TCP congestion control:

i) The inital threshold for impementation 1 is 32 and for implementation 2 is 16. This is because it is at these values that slow start stops and congestion avoidance begins.

ii) The threshold for implementation 1 at the 17th round is 20 because congestion was last detected at round 15, when the window size was 40; the window size is halved when a congestion event is detected. Similarly, the threshold for implementation 2 at the 17th round is 7.

iii) If congestion was detected at round 8, the new threshold for implementation 1 would have been 16 since at round 8 the window size is 33. Similarly, for implementation 2 this would have been 9, since the window size at round 8 is 19.

iv) For implementation 1: During the 1st transmission round, no packets are sent but 1 packet is sent in the 2nd transmission round; packets 2-3 are sent in the 3rd round; packets 4-7 are sent in the 4th round; packets 8-15 in the 5th round; packets 16-31 in the 6th round and packets 32-63 are sent in the 7th round. As such, the 50th segment is sent during the 7th transmission round.

For implementation 2: During the 1st transmission round 1 packet is sent and in the 2nd transmission round packets 2-3 are sent; packets 4-7 in the 3rd round; packets 8-15 in the 4th round; packets 16-31 in the 5th round; packets 32-48 in the 6th round and packets 49-66 are sent in the 7th round. As such, the 50th segment is sent during the 7th transmission round.

v) An implementation supports fast retransmit / fast recovery if it reacts to congestion events where ACKs are still flowing by just halving the window size and staying in congestion avoidance mode. This happens at time indexes 11, 15 and 19 in implementation 2. As such, implementation 2 supports fast retransmit / fast recovery.

Problem F.5: *application layer protocol concepts* (2+2+3+2+1 = 10 points)

a) Some application protocols (e.g., HTTP and SMTP) support persistent connections. What are persistent connections and why are they useful?

b) Some application protocols (e.g., HTTP and extended versions of SMTP) support pipelining. What is pipelining and when should we *not* pipeline requests?

c) What is HTTP chunked transfer encoding? How does an HTTP server indicate the end of a message in this case? Would SMTP benefit from a chunked encoding mechanism?

d) Some application protocols provide special commands to upgrade the TCP transport to a transport secured by TLS (e.g., HTTP's CONNECT or IMAP's STARTTLS). In addition, there are often separate port numbers allocated for the usage of TLS (e.g., https (443) or imaps (993)). Discuss the pros and cons of both approaches.

e) Some application layer protocols allow the peers to negotiate who terminates the underlying TCP connection. Why is this useful?

Solution:

a) The idea behind HTTP persistent connections is to use the same TCP connection to send and receive multiple HTTP requests/responses, as opposed to opening a new connection for each individual request/response interaction. This has several advantages:

   - Less CPU and memory usage (because fewer connections are open simultaneously)
   - Reduced network congestion (fewer TCP connections)
   - Reduced latency in subsequent requests (no handshaking)

b) In HTTP (but similar in SMTP), pipelining is a technique in which multiple HTTP requests are sent on a single HTTP connection without waiting for the corresponding responses. Clients should *not* pipeline requests using non-idempotent methods (`POST`, `PUT`). Otherwise, a premature termination of the transport connection could lead to indeterminate results.

c) Chunked transfer encoding is a data transfer mechanism in which an HTTP server serves content in a series of chunks. The server sends each chunk with its length prepended; it marks the end of the message with a zero-length chunk to allow clients to recognize the end of the message.

   The SMTP protocol would not benefit much from a chunked encoding mechanism since it determines the end of a message using a special end of message marker and hence SMTP peers can exchange data in chunks already (although the receiver has to parse the data for the end of message marker).

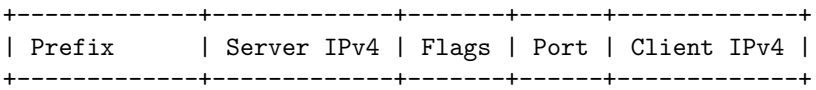
d) An upgrade of an unsecure TCP connection to a TLS-secured connection has the advantage that the protocol endpoints can first negotiate the availability of security mechanisms. Furthermore, in-band security upgrades do not require the allocation of additional port numbers.

   The downside of in-band security upgrades is that in-band upgrades make it more complicated to filter traffic. With separate port numbers, firewalls can easily disallow unprotected traffic by only allowing the "secure" ports of a protocol to be used. (Of course this does not necessarily imply that *all* traffic running over a secure port is actually TLS secured but the likelihood is high with well managed servers.)

e) The TCP endpoint initiating the TCP teardown procedure ends up in the `TIME_WAIT` state. In order to scale well, it is desirable that clients take costs associated with the `TIME_WAIT` state and not the servers.

Problem F.6: *teredo - dynamic IPv6 tunnels crossing NATs* (4+2+1+2+1 = 10 points)

Teredo (defined in RFC 4380) is a tunneling mechanism that provides IPv6 access to hosts behind Network Address Translators (NATs). A host behind a NAT running a Teredo client sends a UDP/IPv4 request to a Teredo server running on the public Internet. The server determines the global IPv4 address and port number assigned to packet by the NAT and returns that information to the Teredo client. The client then forms an IPv6 address which has the following format:

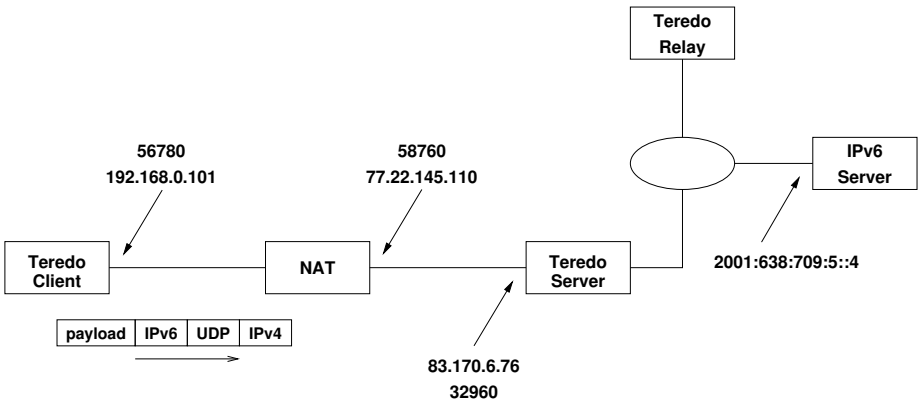


Field	Width	Description
Prefix	32 bit	Teredo service prefix (2001:0000::/32)
Server IPv4	32 bit	IPv4 address of a Teredo server
Flags	16 bit	flags indicating type of address and NAT
Port	16 bit	obfuscated "mapped UDP port" of the client
Client IPv4	32 bit	obfuscated "mapped IPv4 address" of the client

The client's port number is obfuscated by performing a bit-wise exclusively OR of the port number with 0xffff and the client's IPv4 address is obfuscated by performing a bit-wise exclusive OR of the address with 0xffffffff.

Teredo encapsulates IPv6 data packets in IPv4/UDP packets. The IPv6 packet uses the calculated Teredo IPv6 address of the client as the source address and a global IPv6 address of the destination as the destination address. Data packets are sent to a Teredo relay, which removes the outer header and forwards the IPv6 packet to its destination.

Consider the following scenario:



a) Calculate the Teredo IPv6 address used by the Teredo client (with the flags set to 0x1809).

b) What is a likely consequence of restarting the Teredo client on the host? What is a likely consequence of restarting the NAT devices?

c) Assume that the Teredo client successully tunneled an IPv6 packet to the IPv6 server at the address 2001:638:709:5::4. The server (not knowing anything about Teredo) generates a response packet. What is the IPv6 destination address of the response packet?

d) What needs to happen on the routing plane so that this response packet gets delivered to a Teredo relay (which then encapsulates the IPv6 packet in an IPv4/UDP packet)? Hint: Consider which IPv6 prefix needs to be announced in the BGP routing protocol.

e) What can happens if the Teredo relay to which the response packet is delivered is not the Teredo server used by the Teredo client? Hint: Consider what happens at the NAT.

Solution:

a) The resulting Teredo IPv6 address is 2001:0:53aa:64c:1809:1a77:b2e9:6e91:

Field	Width	Value	Description
Prefix	32 bit	2001:0000	Teredo service prefix
Server IPv4	32 bit	53aa:064c	IPv4 address 83.170.6.176 in hexadecimal notation
Flags	16 bit	1809	flags indicating type of address and NAT
Port	16 bit	1a77	obfuscated ”mapped UDP port” of the client
Client IPv4	32 bit	b2e9:6e91	obfuscated ”mapped IPv4 address” of the client

```
 77 = 4d = 0100 1101  =>  1011 0010 = b2
 22 = 16 = 0001 0110  =>  1110 1001 = e9
145 = 91 = 1001 0001  =>  0110 1110 = 6e
110 = 6e = 0110 1110  =>  1001 0001 = 91
```

b) If the Teredo client restarts, it will likely obtain a different NAT binding and hence it will obtain a different IPv6 address. Thus, all existing IPv6 connections will stall. Similarly, if that NAT restarts, it will loose the NAT binding and hence all existing IPv6 connections will break.

c) Since the source address of the original IPv6 packet is the Teredo IPv6 address of the client, the response packet is send to the IPv6 address 2001:0:53aa:64c:1809:1a77:b2e9:6e91.

d) In order to deliver this packet to a Teredo relay, a network running Teredo relay must announce the Teredo service prefix (2001:0000::/32) in the BGP routing system.

e) If the routing system delivers the response packet to a Teredo server that is not the Teredo server used by the Teredo client, then the encapsulated IPv4/UDP packet will likely be dropped by the NAT since the NAT does not maintain a binding for the IPv4 address this packet is originating from. Hence, in this case, the Teredo relay needs to ask the Teredo server to ask the Teredo client to send an IPv4/UDP to the Teredo relay in order to establish a suitable binding in the NAT.

Problem F.7: *domain name system* (2+2+2+2+2 = 10 points)

Indicate which of the following statements are correct or incorrect by marking the appropriate boxes. For every correctly marked box, you will earn two points. For every incorrectly marked box, you will loose one point. Statements which are not marked or which are marked as true and false will be ignored. The minimum number of points you can achieve is zero.

true	false	
☐	☐	Information obtained from the DNS is always consistent.
☐	☐	Every resource record has a time-to-live attribute that controls how many seconds information from a resource record can be stored in a local cache.
☐	☐	A server may include additional information in his response to a query in the hope that the client will find the information useful.
☐	☐	DNS reverse mappings can map an IP address to multiple names.
☐	☐	Application protocols (e.g., DKIM) sometimes lookup information stored in TXT records that follow a certain naming convention.

Solution:

true	false	
☐	☒	Information obtained from the DNS is always consistent.
☒	☐	Every resource record has a time-to-live attribute that controls how many seconds information from a resource record can be stored in a local cache.
☒	☐	A server may include additional information in his response to a query in the hope that the client will find the information useful.
☐	☒	DNS reverse mappings can map an IP address to multiple names.
☒	☐	Application protocols (e.g., DKIM) sometimes lookup information stored in TXT records that follow a certain naming convention.