Department of Informatics
Technical University of Munich (TUM)

Note:

- During the attendance check a sticker containing a unique QR code will be put on this exam.
- This QR code contains a unqiue number that associates this exam with your matriculation number.
- This number is printed both next to the QR code and to the signature field in the attendance check list.

Advanced Computer Networks

Module: IN2097
Examiner: Prof. Dr.-Ing. Georg Carle
Date: 10.02.2016
Exam: Final exam

	P 1	P 2	P 3	P 4
First correction				
Second correction				

Left room from __________ to ______________

from __________ to ______________

Early submission at ______________

Notes ____

Final exam

Advanced Computer Networks

Prof. Dr.-Ing. Georg Carle
Chair for Network Architectures and Services
Department of Informatics
Technical University of Munich (TUM)

Wednesday, 10.02.2016
11:30 – 12:30

- This exam consists of
  - **16 pages** with a total of **4 problems** and
  - a two-sided printed **cheat sheet**.

  Please make sure now that you received a complete copy of the exam.
- Subproblems marked by * can be solved without results of previous subproblems.
- **Answers are only accepted if the approach is documented.** Give a reason for each answer unless explicitly stated otherwise in the respective subproblem.
- Do not write with red or green colors nor use pencils.
- The total amount of achievable credits in this exam is 60.
- Allowed resources:
  - one printed **dictionary** German ⇔ native language **without annotations**
- Physically turn off all electronic devices, put them into your bag and close the bag.

Problem 1 Quiz (7 credits)

The following questions cover multiple topics and can be solved independently of each other.

a)* Name and explain the two basic principles for handling I/O from hardware devices such as NICs.

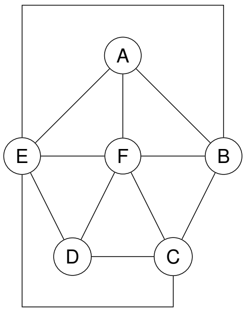
- Polling is done by actively querying the NIC/memory buffer for IO, i.e. program/OS query actively.
- Interrupts are a hardware mechanism for signaling that IO is ready, i.e, the NIC/DMA engine indicates available IO.

b)* Hardware routers often posses a ternary content addressable memory (TCAM) module. This type of memory supports a third state `*` besides the usual binary states `0` and `1`. Explain why this kind of memory is beneficial with respect to implementing lookup tables for hierarchical address structures such as IP addresses.

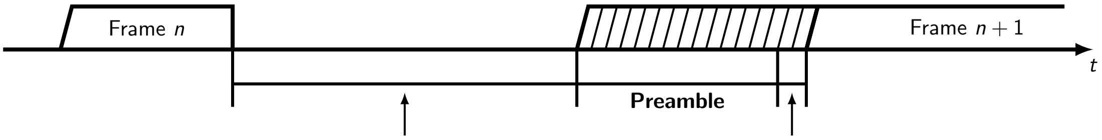
- The hierarchical structure of IP allows for aggregation of address blocks.
- Using the `*` for aggregated address blocks allows a reduced table size.
- Faster lookups due to reduced table size

c)* Determine the k-core such that k is maximized and the core is non-empty. State k and all nodes contained.

- maximum value of k = 3
- all nodes, i. e., A, B, C, D, E & F



d)* The figure below shows the minimum idle time of the medium between two consecutive Ethernet frames. Name the remaining two fields during that time span.



Inter Frame Space (IFS)

Start Frame Delimiter (SFD)

Problem 2 NAT (14 credits)

a)* Explain two different approaches to mitigate or solve the IPv4 address scarcity.

Address trading/market: more efficient use of address space
IPv6 : create more addresses
NAT : address sharing

b)* Explain the difference between private (as defined in RFC 1918) and public IPv4 addresses.

Not routed on the Internet
Can be used by anyone without registration

c)* Explain why NAT causes problems with peer-to-peer applications.

Does not allow inbound connections.

d)* How does an Application Layer Gateway (ALG) operate?

An ALG rewrites application layer protocols that carry IP addresses or ports in their payload.

e) Is an ALG implemented on a host located behind a NAT, on a NAT router, or on a public server on the Internet?

Located on the NAT router

f) Name two examples for protocols that can be handled by an ALG.

Any application layer protocol with IP addresses or ports in the payload, e.g. FTP, SIP, DNS, etc.

g)* Name two different approaches how a host located behind a NAT router can detect the public IP address assigned by the NAT.

STUN (Session Traversal Utilities for NAT, RFC 5389)
UPnP (Universal Plug and Play)
Web service such as *http://whatismyip.com*

Figure 2.1 (see next page) describes a simiplified 464XLAT setup, consisting of a client (PC1), a customer-side translator (CLAT), a provider-side translator (PLAT), and a public server (SRV). Both PC1 and SRV are legacy devices that do not support IPv6. We assume that a webbrowser on PC1 establishes a connection to a webserver running on SRV. The message (gray line) shown in Figure 2.1 represents the HTTP request sent by the client.

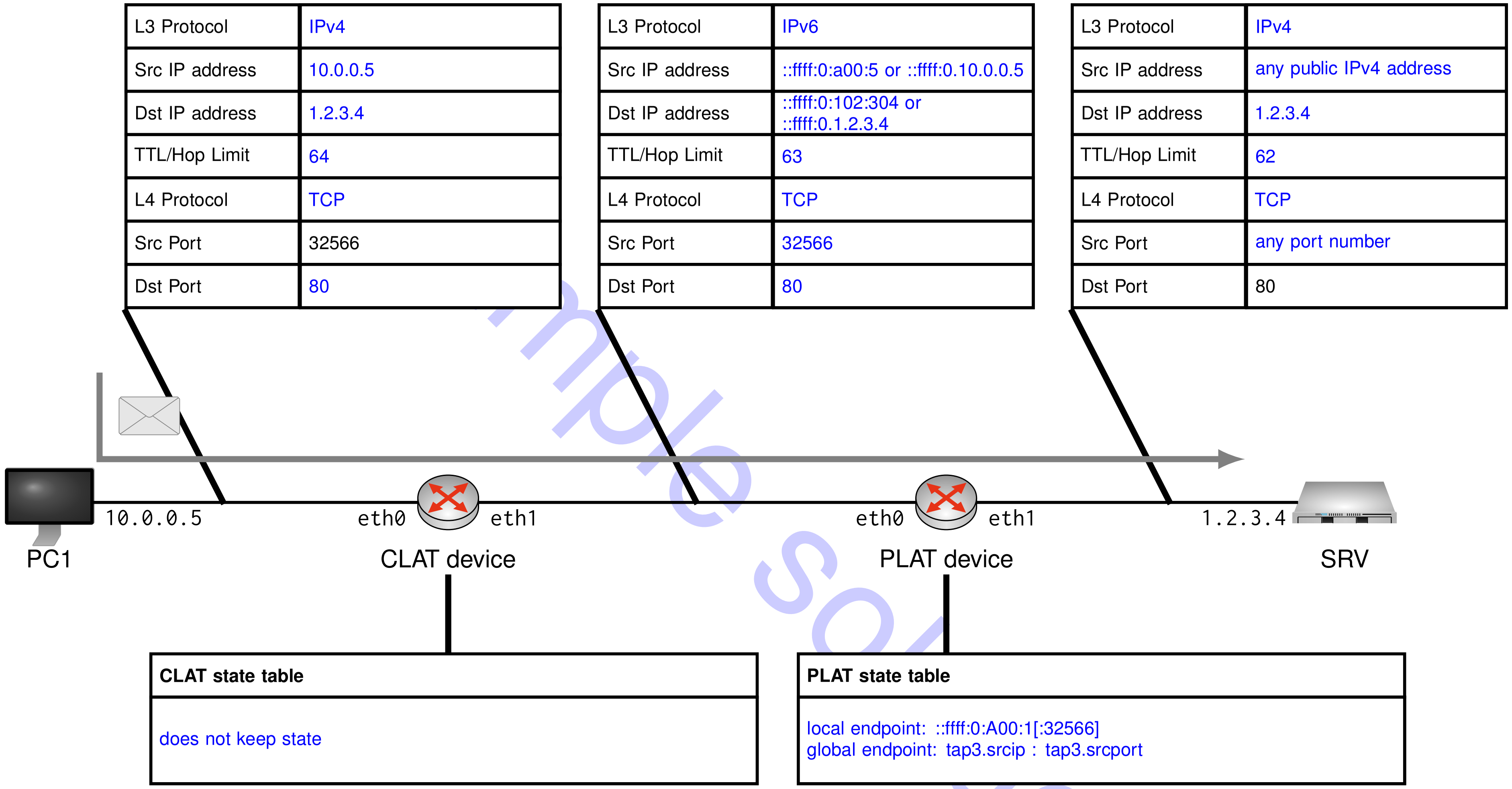
The CLAT and PLAT devices translate IPv4 packets into IPv6 packets and the other way round. The CLAT device performs stateless NAT46, i. e., it translates from the IPv4 address space into a special reserved IPv6 address space (`::ffff:0:0/96`). The PLAT device translates from the IPv6 address space into the IPv4 address space.

h)* Complete the header fields of the HTTP request that can be observed at the three indicated links in Figure 2.1. Assume that the PLAT device uses a random port binding strategy. If the content of a field is not uniquely defined from the given information, make a **meaningful** choice. **Important: Do not abbreviate your addresses. Use real values!**

i) Fill out the PLAT state table in Figure 2.1 after the HTTP connection was established. Use the same addresses and ports as specified in Problem 2h).

j) Why can the CLAT device operate statelessly while the PLAT device needs to keep state?

The CLAT device performs stateless translation from IPv4 to IPv6, which NAT44 identifies a whole subnet via port numbers.
Keeping state is not necessary, since the target address space is much larger than the source address space, so all IPv4 addresses can be mapped into a /96 subnet in IPv6.



L3 Protocol	IPv4
Src IP address	10.0.0.5
Dst IP address	1.2.3.4
TTL/Hop Limit	64
L4 Protocol	TCP
Src Port	32566
Dst Port	80

L3 Protocol	IPv6
Src IP address	::ffff:0:a00:5 or ::ffff:0.10.0.0.5
Dst IP address	::ffff:0:102:304 or ::ffff:0.1.2.3.4
TTL/Hop Limit	63
L4 Protocol	TCP
Src Port	32566
Dst Port	80

L3 Protocol	IPv4
Src IP address	any public IPv4 address
Dst IP address	1.2.3.4
TTL/Hop Limit	62
L4 Protocol	TCP
Src Port	any port number
Dst Port	80

CLAT state table
does not keep state

PLAT state table
local endpoint: ::ffff:0:A00:1[:32566] global endpoint: tap3.srcip : tap3.srcport

Figure 2.1: 464XLAT topology

Problem 3 Load Balancing & Traceroute (22 credits)

IP-based networks allow for complex network topologies. A popular tool to discover topologies for such networks is `traceroute`. However, one has to know the impact of techniques such as load balancing on traceroute to make use of this tool.

a)* Explain the purpose of load balancing for network traffic.

- Distribution of traffic do different servers/routers/devices
- Distribute load to several devices to avoid overload scenarios/ increase robustness / higher availability

b) Given redundant paths; explain which problems might arise if packets are routed independently of each other.

Packets might arrive out of order at the destiantion due to different one-way delays along redundant paths.

c) Explain the impact on transport layer protocols such as TCP.

TCP expects that segments arrive in order. Segments arriving out of order interfere with normal ACK mechanisms since the destination interprets a gap in the data stream as packet loss and acknowledges the next Byte expected. This in turn leads to duplicate ACKs at the source node, which may initiate congestion avoidance and starts retransmitting what it believes to be lost.

d)* State the header fields that are needed to uniquely identify a flow in the context of transport layer protocols such as TCP or UDP.

1. Source IP address
2. Destination IP address
3. Source port
4. Destination port
5. L4 Protocol

e) Briefly explain how load balancing over redundant links can be accomplished, while avoiding the problems discussed in Subproblems 3b) and 3c).

Differentiate by flows and forward packets belonging to the same flow along the same path. This may be done by hashing the 5-tuple of Subproblem 3d).

f)* Explain how traceroute works. (Give a detailed explanation of the basic principle. You do not have to explain advanced variants of traceroute.)

- Send IP packets with increasing TTL values starting at a value of 1.
- Routers decrease the TTL on forwarding if TTL is not 0.
- When the TTL reaches 0 (or conversely if a router receives a packet with TTL=1 that shall be forwarded), the respective router discards the packet and creates an ICMP Time Exceeded / TTL Exceeded , which is returned to the original sender of the packet.
- This error message contains the IP address of the router that discarded the packet (source IP) as well as the L3 header and the first 16 B following the L3 header of the packet that has been discarded.

g) How could load balancing be detected in the output of traceroute?

Several different IPs having a common predecessor , e.g.

```
[...]
7  (80. 81.192.164)  9.308 ms  9.386 ms  9.330 ms
8 (213.239.245.249) 12.561 ms 12.537 ms
  (213.239.245.253) 12.561 ms
[...]
```

That predecessor is the load balancer.

Figure 3.1 shows the actual topology of a network. On node *Src* traceroute was executed several times. The destination of the traceroute calls was node *Dst*. Every node in this topology only has a single unique IP address. No loopback links are present in this topology. Assume that routers are well-behaved and standards compliant.

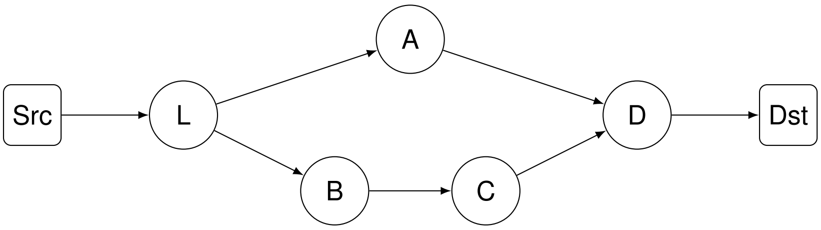


Figure 3.1: Actual network topology

```
[TTL] [IP]                [Delay]
1         80.181.192.164   9.4 ms    9.4 ms    9.3 ms
2        213.239.245.249  12.5 ms
         213.239.245.253  12.3 ms   12.4 ms
3        213.239.245.208  14.1 ms   13.9 ms
         213.239.245.249  13.9 ms
4        213.239.245.207  16.3 ms   16.6 ms
         213.239.245.210  16.4 ms
```

Listing 1: Output of traceroute

h)* Argue whether or not Listing 1 shows a valid output of traceroute for the topology depicted in Figure 3.1.

Listing 1 is not a valid traceroute output for the topology in Figure 3.1: The node with the IP address 213.239.245.249 shows up twice at different hops but according to Figure 3.1 there is no node that can be reached at TTL=2 *and* TTL=3.

i) Complete the figure in the solution box with a minimal number of arcs such that it represents a valid topology for the output of Listing 1. Hint: Additional load balancers may be used.

Add arc between *B* and *A* to the topology in Figure 3.1 .

j) Give a valid mapping between the IP addresses of Listing 1 and node names of your topology in Problem 3i).

IP address	Node name
80.181.192.164	*L*
213.239.245.249	*A*
213.239.245.253	*B*
213.239.245.208	*C*
213.239.245.207	*D*
213.239.245.210	*Dst*

From another run of traceroute someone inferred that the network's topology should be drawn as follows:

Src → L → B → D → D → Dst

k)* State the paths that probes with TTL $\leq$ 4 sent from Src to Dst must have taken such that the topology above is valid. Base the answer either on your topology of Subproblem 3i) or on the topology given in Figure 3.1.

TTL	Path (according to Figure 3.1)	Path (according to Subproblem 3i))
1	$Src \rightarrow L$	$Src \rightarrow L$
2	$Src \rightarrow L \rightarrow B$	$Src \rightarrow L \rightarrow B$
3	$Src \rightarrow L \rightarrow A \rightarrow D$	$Src \rightarrow L \rightarrow A \rightarrow D$
4	$Src \rightarrow L \rightarrow B \rightarrow C \rightarrow D$	$Src \rightarrow L \rightarrow B \rightarrow C \rightarrow D$ or $Src \rightarrow L \rightarrow B \rightarrow A \rightarrow D$

Assume that the load balancer *L* in Figure 3.1 balances traffic on a per-flow basis. Furthermore, assume that traceroute sends TCP probes.

l)* Explain why traceroute fails to discover the correct topology under these circumstances.

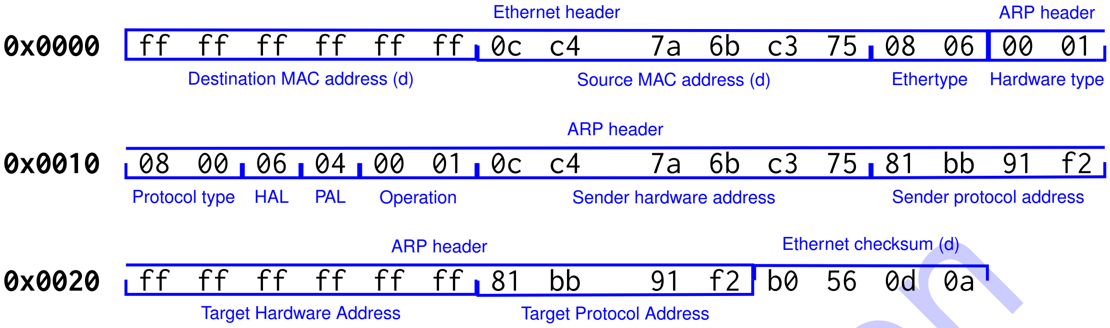
The load balancer classifies all probes as one flow and routes them along the same path, i. e., only one path is being detected.

m)* Suggest an improvement to traceroute such that the correct network topology can be discovered.

traceroute has to send probes with different port values to avoid that the load balancer classifies all probes as one flow.

Problem 4 Wireshark (17 credits)

We consider the Ethernet frame (including link layer checksum) depicted in Figure 4.1 as hexdump in network byte order.



```
0x0000  ff ff ff ff ff ff 0c c4 7a 6b c3 75 08 06 00 01
0x0010  08 00 06 04 00 01 0c c4 7a 6b c3 75 81 bb 91 f2
0x0020  ff ff ff ff ff ff 81 bb 91 f2 b0 56 0d 0a
```

Figure 4.1: Hexdump of an Ethernet frame (including link layer checksum) in network byte order

Note: To solve this problem use the cheat sheet that is handed out separately.

a)* Explain the difference between network and host byte order.

Network byte order is commonly defined as big endian, i.,e., most significant byte first, while host byte order may be either big or (more common today) little endian depending on the processor architecture.

b)* Reason whether or not there is a difference between network and host byte order for single byte values.

Ordering only applies to bytes within a multi-byte value, i. e., given a 16 bit value it determines which octet is read first from memory. It has no effect on values consisting of a single byte only.

c)* Explain the difference between *protocol data unit (PDU)* and *service data unit (SDU)*.

The PDU of layer N consists of layer-specific information (header/trailer) called *protocol control information (PCI)* destined for the same layer on the receiving node and an SDU, which is the payload of layer N and itself a PDU of layer $N+1$. .

d)* Mark and name *all* parts of the protocol specific information for layer 2 in Figure 4.1.
Note: Put your solution directly in Figure 4.1.

e) Briefly explain the purpose of **each** part.

- Destination and source MAC addresses identify the receiver and transmitter on the local network , respectively. They are used to direct frames to the next hop.
- The Ethertype identifies the type of the L2-SDU.
- The checksum (CRC32) is used to detect (not correct) transmission errors.

To make it a bit easier, we let you know that the payload of the Ethernet frame in Figure 4.1 is ARP.

f)* How does a receiving node determine that the frame indeed carries an ARP payload?

Ethertype `0x0806` identifies the payload as ARP.

g)* Explain the main purpose of ARP.

Given the network layer address of a node on the local network, determine its link layer address.

h) Reason which network layer protocol is being served.

IPv4 as indicated by the protocol type field (`0x0800`) in the ARP header.

i) Determine the opcode **and** meaning of the ARP packet.

It is a request (Operation: `0x0001`).

j) Determine the source and target link layer addresses of the ARP packet.

Destination is `ff:ff:ff:ff:ff:ff` , source `0c:c4:7a:6b:c3:75`.

k) State the source and destination network layer addresses of the ARP packet in the manner customary for that network layer protocol.

Both source and destination are 129.187.145.242.

According to Subproblem 4k), you may be a bit skeptical how meaningful that frame is. However, the frame was captured on a real network as is and it makes perfectly sense as it is a *gratuitous* ARP packet.

l)* Explain what gratuitous ARP is being used for **and** how it works.

Gratuitous ARP requests are being used to determine whether or not a self-assigned network layer address is already in use.

A node assigns a random network layer address from an eligible block to itself. Before it starts using that address, it sends a gratuitous ARP request, i. e., asking the local network for the link layer address associated with that network layer address. That packet contains the randomly assigned network layer address of the sender. If another node is already using that address, it will respond to that request just like to any other ARP request by copying the payload, switching the sender/receiver fields and filling in its own network layer address, which is now contained twice in the reply. If that reply is received by the originator, it knows for sure that someone else is using the self-assigned address.

Additional space for solutions–clearly mark the (sub)problem your answers are related to and strike out invalid solutions.