Advanced Computer Networking (ACN)

Exercise 2 – Solution

Prof. Dr.-Ing. Georg Carle

Sebastian Gallenmüller, Max Helm, Benedikt Jaeger,
Marcel Kempf, Patrick Sattler, Johannes Zirngibl

Chair of Network Architectures and Services
School of Computation, Information, and Technology
Technical University of Munich

Outline

Announcements

Tutorial2 – Problem 1: Wireshark

Tutorial2 – Problem 2: Spanning Tree Protocol

Tutorial2 – Problem 3: Network Topology and Tracing Routes

Announcements

For questions and problems:

- Always use this mail address: `acn@net.in.tum.de`
- If you reply to a mail always use Reply All, usually results in a faster response

Tutorial

- Deadline for tutorial2 was ~15 minutes ago
- If you haven't yet, commit and push your solution **now**

1 a)

Briefly explain which purpose the FCS serves and how it is computed.

1 a)

Briefly explain which purpose the FCS serves and how it is computed.

- Frame Check Sequence
- Bit error detection $\rightarrow$ discard frame if FCS is wrong
- No error correction
- Computed using CRC32

Tutorial2 – Problem 1: Wireshark

1 b)

Write the body of the given function `extend_hexdump()`.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A										

Tutorial2 – Problem 1: Wireshark

1 b)

Write the body of the given function `extend_hexdump()`.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						

FCS (bytes FF FF FF FF at 0x0056–0x0059)

Tutorial2 – Problem 1: Wireshark

1 c)

Write two functions which return the source and destination MAC of a given Ethernet frame.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						

FCS

Tutorial2 – Problem 1: Wireshark

1 c)

Write two functions which return the source and destination MAC of a given Ethernet frame.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
	Destination MAC						Source MAC									
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						
							FCS									

Tutorial2 – Problem 1: Wireshark

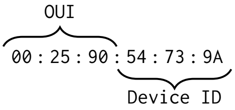
1 d)

Try to identify the hardware vendors, based on MAC addresses given in the IPv6 hexdump.

`00 : 25 : 90 : 54 : 73 : 9A`

1 d)

Try to identify the hardware vendors, based on MAC addresses given in the IPv6 hexdump.



- MAC Address: 48 bit long L2 address
- Organisation Unique Identifier (OUI): first 3 B identify hardware manufacturer
- Online lookup, e.g.: `http://standards-oui.ieee.org/oui.txt`
  - `C4:2A:D0` → Apple, Inc
  - `4C:34:88` → Intel Corporate
  - `04:D3:B0` → Intel Corporate

1 d)

Try to identify the hardware vendors, based on MAC addresses given in the IPv6 hexdump.

OUI

`00 : 25 : 90 : 54 : 73 : 9A`

Device ID

- MAC Address: 48 bit long L2 address
- Organisation Unique Identifier (OUI): first 3 B identify hardware manufacturer
- Online lookup, e.g.: `http://standards-oui.ieee.org/oui.txt`
  - `C4:2A:D0` → Apple, Inc
  - `4C:34:88` → Intel Corporate
  - `04:D3:B0` → Intel Corporate

What about `33:33:FF:D7:6D:A0`?

Tutorial2 – Problem 1: Wireshark

1 d)

Try to identify the hardware vendors, based on MAC addresses given in the IPv6 hexdump.

OUI

`00 : 25 : 90 : 54 : 73 : 9A`

Device ID

- MAC Address: 48 bit long L2 address
- Organisation Unique Identifier (OUI): first 3 B identify hardware manufacturer
- Online lookup, e.g.: `http://standards-oui.ieee.org/oui.txt`
  - `C4:2A:D0` $\longrightarrow$ Apple, Inc
  - `4C:34:88` $\longrightarrow$ Intel Corporate
  - `04:D3:B0` $\longrightarrow$ Intel Corporate

What about `33:33:FF:D7:6D:A0`?

- IPv6 Multicast Address (RFC 2464, Section 7)
- `33:33:[last 4 octets of the IPv6 address]`

Tutorial2 – Problem 1: Wireshark

1 e)

Write a function that returns the type of the layer 3 payload.
It should return **4** / **6** for IPv4 / IPv6 and None otherwise.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						

Destination MAC

Source MAC

FCS

Tutorial2 – Problem 1: Wireshark

1 e)

Write a function that returns the type of the layer 3 payload.
It should return **4** / **6** for IPv4 / IPv6 and None otherwise.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						

Destination MAC (0x0000: 33 33 FF D7 6D A0)
Source MAC (0x0000: 00 25 90 54 73 9A)
Ethertype (0x0000: 86 DD)
FCS (0x0050: FF FF FF FF)

1 e)

Write a function that returns the type of the layer 3 payload.
It should return **4** / **6** for IPv4 / IPv6 and None otherwise.

	Destination MAC	Source MAC	Ethertype	
0x0000	33 33 FF D7 6D A0	00 25 90 54 73 9A	86 DD	60 00

Ethertype	Protocol
0x0800	IPv4 (Internet Protocol, Version 4)
0x0806	ARP (Address Resolution Protocol)
0x0842	WoL (Wake on Lan)
0x8035	RARP (Reverse ARP)
0x814c	SNMP (Simple Network Management Protocol)
0x86dd	IPv6 (Internet Protocol, Version 6)

1 f)

How can the beginning of the payload be determined for Ethernet frames?

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
	Destination MAC						Source MAC						Ethertype			
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						
							FCS									

1 g)

How can the beginning of the payload be determined for IP packets?
What is the difference between IPv4 and IPv6?

1 g)

How can the beginning of the payload be determined for IP packets?
What is the difference between IPv4 and IPv6?

- Ethernet header → fixed length

IPv4

- Variable length

→ Internet Header Length (IHL)

IPv6

- Fixed length but has extension headers
- Next Header points to Extension Header

→ Extension header has Length field

Tutorial2 – Problem 1: Wireshark

1 h)

Write three functions:

- `cutL2PDU()` shall return a `bytearray` containing the Layer 2 PDU

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
			Destination MAC							Source MAC				Ethertype		
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr						Source Address					
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
			Source Address								Destination Address					
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
			Destination Address				Type	Code								
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
							Target Address									
0x0050	00	25	90	54	73	9A										
			Source Link Address													

Tutorial2 – Problem 1: Wireshark

1 h)

Write three functions:

- `cutL2PDU()` shall return a `bytearray` containing the Layer 2 PDU
- `cutL2SDU()` shall return a `bytearray` containing the Layer 2 SDU

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr		Source Address									
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
	Source Address						Destination Address									
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
	Destination Address						Type	Code								
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
	Target Address															
0x0050	00	25	90	54	73	9A										
	Source Link Address															

Tutorial2 – Problem 1: Wireshark

1 h)

Write three functions:

- `cutL2PDU()` shall return a `bytearray` containing the Layer 2 PDU
- `cutL2SDU()` shall return a `bytearray` containing the Layer 2 SDU
- `cutIPPDU()` shall return a `bytearray` containing the Layer 3 PDU.

0x0000	33	33	FF	D7	6D	A0	00	25	90	54	73	9A	86	DD	60	00
0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr		Source Address									
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
	Source Address						Destination Address									
0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
	Destination Address						Type	Code								
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
	Target Address															
0x0050	00	25	90	54	73	9A										
	Source Link Address															

1 i)

Write a function which helps to identify the type of payload an IPv4 or IPv6 packet is carrying. [. . .]
It should return a bytearray containing the L4 payload identifier or an empty bytearray if neither an IPv4 nor an IPv6 payload was found.

1 i)

Write a function which helps to identify the type of payload an IPv4 or IPv6 packet is carrying. [. . .]
It should return a bytearray containing the L4 payload identifier or an empty bytearray if neither an IPv4 nor an IPv6 payload was found.

0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr		Source Address									
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
	Source Address						Destination Address									

1 j)

Based on your answer of i), what are the protocols contained in the two given hexdumps.

0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr		Source Address									
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
	Source Address						Destination Address									

Tutorial2 – Problem 1: Wireshark

1 j)

Based on your answer of i), what are the protocols contained in the two given hexdumps.

0x0010	00	00	00	20	3A	FF	FE	80	00	00	00	00	00	00	02	25
					Next Hdr		Source Address									
0x0020	90	FF	FE	54	73	9A	FF	02	00	00	00	00	00	00	00	00
	Source Address						Destination Address									

Protocol Number / Next Header	Protocol
0x01	ICMPv4 (Internet Control Message Protocol, Version 4)
0x04	IPv4 encapsulation
0x06	TCP (Transmission Control Protocol)
0x11	UDP (User Datagram Protocol)
0x29	IPv6 encapsulation
0x2f	GRE (General Routing Encapsulation)
0x3a	ICMPv6 (Internet Control Message Protocol, Version 6)
0x84	SCTP (Stream Control Transmission Protocol)

Tutorial2 – Problem 1: Wireshark

1 k)

Explain the header fields (header fields and the content type) and the content of the IPv4 packet identified in j).

1 l)

Explain the header fields (header fields and the content type) and the content type of the IPv6 packet identified in j) and how this relates to the answer in d).

0x0030	00	01	FF	D7	6D	A0	87	00	19	C9	00	00	00	00	20	01
0x0040	4C	A0	20	01	00	11	02	25	90	FF	FE	D7	6D	A0	01	01
0x0050	00	25	90	54	73	9A	FF	FF	FF	FF						

Destination Address (0x0030: 00 01 FF D7 6D A0)

FCS (0x0050: FF FF FF FF)

1 k)

Explain the header fields (header fields and the content type) and the content of the IPv4 packet identified in j).

1 l)

Explain the header fields (header fields and the content type) and the content type of the IPv6 packet identified in j) and how this relates to the answer in d).

```
0x0030  00 01 FF D7 6D A0 87 00  19 C9 00 00 00 00 20 01
0x0040  4C A0 20 01 00 11 02 25  90 FF FE D7 6D A0 01 01
0x0050  00 25 90 54 73 9A FF FF  FF FF
```

- Destination Address: 00 01 FF D7 6D A0
- Type: 87
- Code: 00
- FCS: FF FF FF FF

Type	Code	Description
128 (0x80)	0	Echo Request
129 (0x81)	0	Echo Reply
133 (0x85) – Router Solicitation	0	NDP (Neighbor Discovery Protocol)
134 (0x86) – Router Advertisement	0	NDP (Neighbor Discovery Protocol)
135 (0x87) – Neighbor Solicitation	0	NDP (Neighbor Discovery Protocol)
136 (0x88) – Neighbor Advertisement	0	NDP (Neighbor Discovery Protocol)

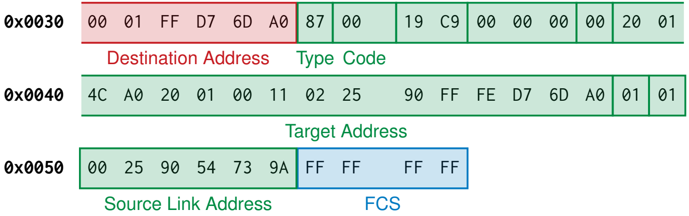
Tutorial2 – Problem 1: Wireshark

1 k)

Explain the header fields (header fields and the content type) and the content of the IPv4 packet identified in j).

1 l)

Explain the header fields (header fields and the content type) and the content type of the IPv6 packet identified in j) and how this relates to the answer in d).



Type	Code	Description
128 (0x80)	0	Echo Request
129 (0x81)	0	Echo Reply
133 (0x85) – Router Solicitation	0	NDP (Neighbor Discovery Protocol)
134 (0x86) – Router Advertisement	0	NDP (Neighbor Discovery Protocol)
135 (0x87) – Neighbor Solicitation	0	NDP (Neighbor Discovery Protocol)
136 (0x88) – Neighbor Advertisement	0	NDP (Neighbor Discovery Protocol)

2 a)

What is the difference between a shortest path tree (SPT) and a minimum spanning tree (MST)?

Tutorial2 – Problem 2: Spanning Tree Protocol

2 a)

What is the difference between a shortest path tree (SPT) and a minimum spanning tree (MST)?

SPT	shortest path between one root node to all other nodes
MST	connect all nodes in a network with the minimum total weight

2 b)

Explain the problem that is being solved by using the spanning tree protocol in a switched network.

2 b)

Explain the problem that is being solved by using the spanning tree protocol in a switched network.

- No TTL or max hops on layer 2
- STP is useful to handle loops on layer 2
- STP recomputes paths and can use backup paths if a bridge fails

2 c)

Explain the purpose of the root bridge and how it is elected.

2 c)

Explain the purpose of the root bridge and how it is elected.

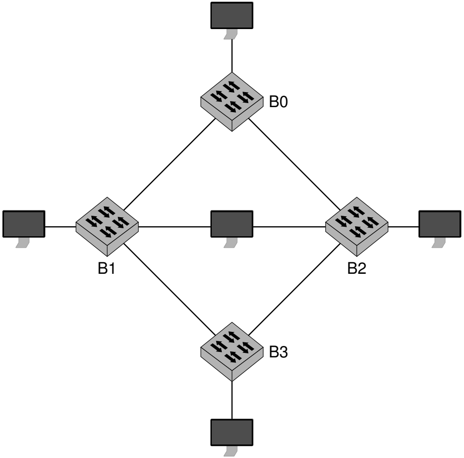
- The root bridge acts as a reference point
- It is determined via an election process:
  - Initially each bridge assumes it is the root bridge
  - Each bridge starts transmitting bridge protocol data units (BPDUs) - bridge ID, root bridge ID, distance to root bridge
  - Bridge ID contains itself a configurable priority parameter
  - Each bridge listens for BDPUs but does not forward them
    - The port with the lowest root bridge ID received will be the new root port
    - If two ports connect to the root bridge the one with the higher cost or if equal the one with the higher id gets blocked
    - The other ports become designated ports

Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

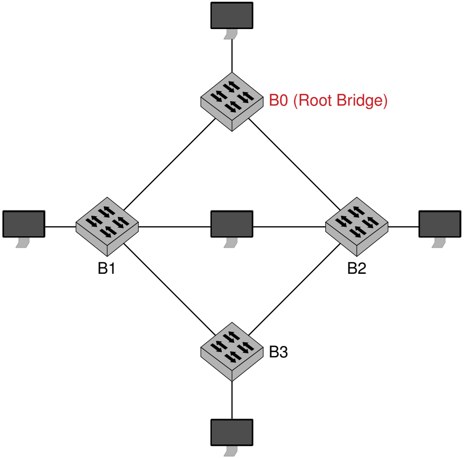


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

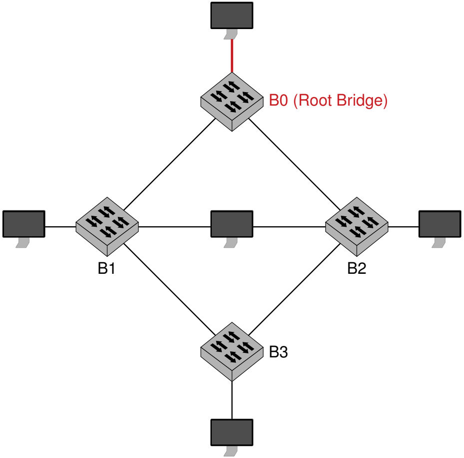


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

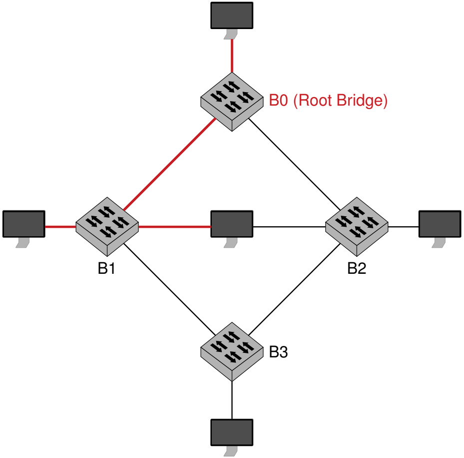


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

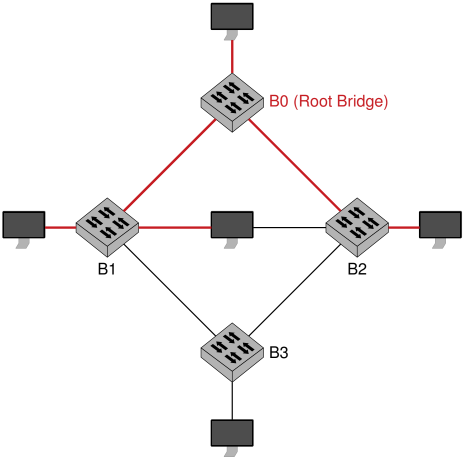


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

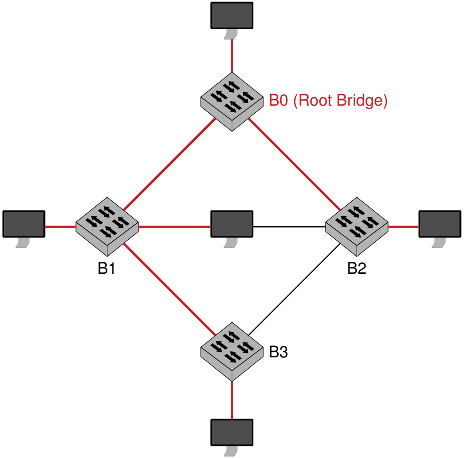


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.

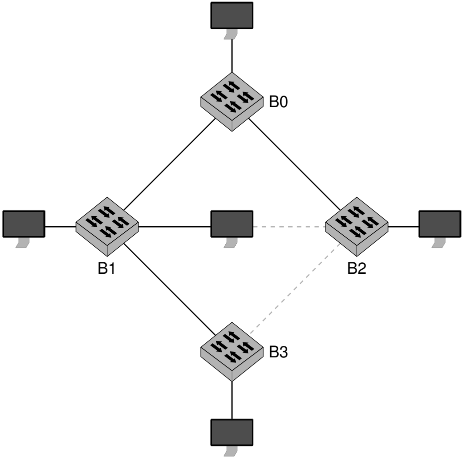


Tutorial2 – Problem 2: Spanning Tree Protocol

2 d)

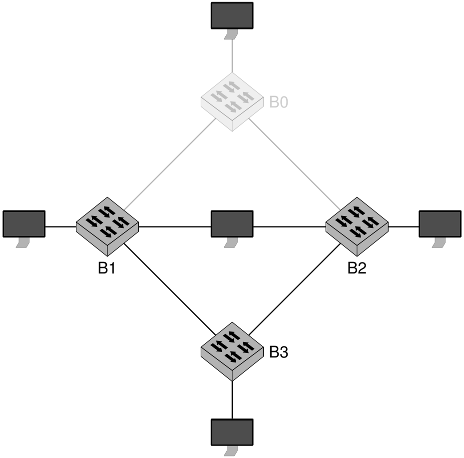
How does the resulting spanning tree look like after the spanning tree algorithm has been applied to the given network topology?

We use another topology here.



2 e)

What happens if bridge B1 (in this case B0) fails?

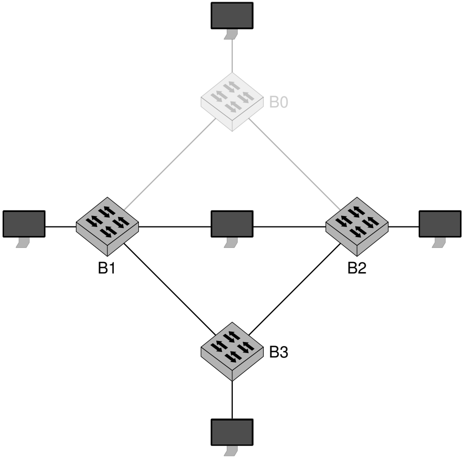


Tutorial2 – Problem 2: Spanning Tree Protocol

2 e)

What happens if bridge B1 (in this case B0) fails?

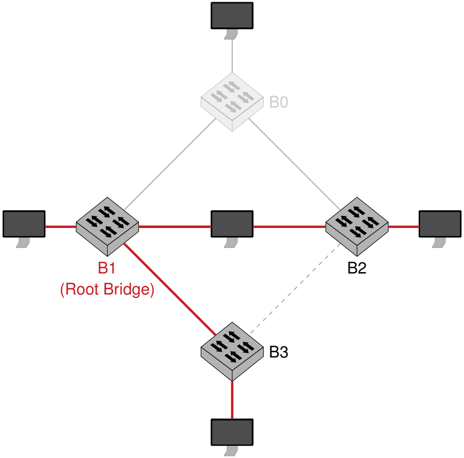
- It stops transmitting BPDUs
- After a specified period other bridge will recognize it is gone
- The algorithm for the root bridge selection starts again
- Any network only connected to B0 will not be reachable



2 e)

What happens if bridge B1 (in this case B0) fails?

- It stops transmitting BPDUs
- After a specified period other bridge will recognize it is gone
- The algorithm for the root bridge selection starts again
- Any network only connected to B0 will not be reachable



3 a)

Explain the basic principle behind traceroute

3 a)

Explain the basic principle behind traceroute

- Send IP packets with increasing TTL values
- When the TTL reaches 0 the router discards the packet and creates an ICMP Time Exceeded / TTL Exceeded, which is returned to the sender
- This error message contains the IP address of the router that discarded the packet (source IP) as well as parts of the original packet
- Traceroute can be used to:
  - detect network problems
  - analyze routing behavior
  - get an approximate path length to a host

3 b)

Discuss the advantages and disadvantages when using packets with different protocols / payload types to generate the traces.

3 b)

Discuss the advantages and disadvantages when using packets with different protocols / payload types to generate the traces.

- ICMP: Sends ICMP echo request packets
  - receives ICMP TTL exceeded error packets
  - requires root privileges
- TCP: Sends TCP SYN packets
  - receives ICMP TTL exceeded error packets
  - requires root privileges
  - uses port 80 to bypass firewalls
- UDP: Sends UDP packets
  - receives ICMP TTL exceeded error packets
  - does not require root privileges
  - uses increasing unlikely port numbers or a fixed port
  - receiving part can be confused by random payload
  - last hop often does not reply

Tutorial2 – Problem 3: Network Topology and Tracing Routes

The following measurements were performed via RIPE Atlas. You can lookup the results [1]

Hop	IP Address	reverse name	ASN	Latencies [ms]		
1	217.10.64.29	hydra-atlasgw.netzquadrat.net	AS15594	3.898ms	0.589ms	0.408ms
2	217.10.64.9	c12-1-e1-3.netzquadrat.net	AS15594	0.771ms	0.741ms	0.795ms
3	213.83.57.97		AS12306	30.387ms	30.362ms	30.662ms
4	82.98.102.6		AS12306	30.691ms	30.617ms	30.604ms
5	212.162.24.57	edge4.frankfurt1.level3.net	AS3356	30.156ms	30.166ms	30.241ms
6				*	*	*
7	4.15.122.46	cenic.ear1.sanjose1.level3.net	AS3356	174.041ms	174.36ms	173.828ms
8	137.164.11.31	dc-svl-agg8–svl-agg4-100ge-2.cenic.net	AS2152	175.205ms		
	137.164.11.29	dc-svl-agg8–svl-agg4-100ge-1.cenic.net	AS2152	180.28ms	186.577ms	
9	137.164.11.0	dc-lax-agg8–svl-agg8-100ge-1.cenic.net	AS2152	180.29ms		
	137.164.11.66	lax-agg8–svl-agg8-100g-3.cenic.net	AS2152	186.731ms	181.244ms	
10	137.164.11.34	dc-lax-agg6–lax-agg8-100ge-3.cenic.net	AS2152	180.057ms	185.605ms	
	137.164.11.6	dc-lax-agg6–lax-agg8-100ge-2.cenic.net	AS2152	180.049ms		
11	137.164.11.61	tus-agg8–lax-agg6-100g-3.cenic.net	AS2152	187.539ms		
	137.164.11.23	dc-tus-agg8–lax-agg6-100ge-1.cenic.net	AS2152	181.976ms	182.336ms	
12	137.164.11.51	sdg-agg4–tus-agg8-3x100ge.cenic.net	AS2152	187.272ms	181.621ms	182.033ms
13	137.164.23.43	dc-sdsc-100ge–sdg-agg4.cenic.net	AS2152	182.134ms	187.637ms	186.656ms
14	192.12.207.46	medusa-mx960.sdsc.edu	AS195	181.975ms	182.095ms	189.205ms
15	192.172.226.78	rommie.caida.org	AS1909	187.69ms	181.883ms	181.526ms

[1] Measurements from DE `https://atlas.ripe.net/measurements/23212460`

Measurements from GB `https://atlas.ripe.net/measurements/23212455`

3 c)

What does the line * * * mean?

```
6 * * *
```

3 c)

What does the line * * * mean?

```
6 * * *
```

- The asterisks signal that no answer was received

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 d)

Compare the command results to the given trace and explain what you find.

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 d)

Compare the command results to the given trace and explain what you find.

Given command executes an IPv4 traceroute

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 d)

Compare the command results to the given trace and explain what you find.

Given command executes an IPv4 traceroute

	IPv6		IPv4	
Hop	Domain	RTT	RTT	Domain
2	2a00:4700:0:8::1	1.512 ms	0.497 ms	nz-csr1-kw5-bb1.rbg.tum.de
3	2a00:4700:0:1::3	1.458 ms	0.946 ms	vl-3010.csr1-2wr.lrz.de
4	-	-	9.302 ms	cr-fra2-be14.x-win.dfn.de
5	cr-fra2-be14.x-win.dfn.de	10.762 ms	9.299 ms	cr-fra2-be14.x-win.dfn.de
6	dfn.mx1.fra.de.geant.net	10.881 ms	9.206 ms	dfn.mx1.fra.de.geant.net
7	ae7.mx1.ams.nl.geant.net	16.758 ms	15.709 ms	ae7.mx1.ams.nl.geant.net
8	internet2-gw.mx1.ams.nl.geant.net	111.152 ms	103.953 ms	internet2-gw.mx1.ams.nl.geant.net
9	ashb.net.internet2.edu	166.537 ms	-	-
10	ashb.net.internet2.edu	167.301 ms	-	-
11	clev.net.internet2.edu	166.439 ms	-	-
12	eqch.net.internet2.edu	167.631 ms	-	-
13	eqch.net.internet2.edu	167.947 ms	-	-
14	fchic.net.internet2.edu	166.520 ms	-	-
15	kans.net.internet2.edu	167.196 ms	-	-
16	denv.net.internet2.edu	167.063 ms	-	-
17	salt.net.internet2.edu	167.973 ms	-	-
18	losa.net.internet2.edu	165.542 ms	-	-
19	hpr-lax-agg10–i2.cenic.net	164.806 ms	168.401 ms	hpr-lax-agg10–i2.cenic.net
20	-	-	168.709 ms	hpr-sdsc-100ge–sdg-hpr3.cenic.net
21	hpr-sdsc-100ge–sdg-hpr3.cenic.net	167.048 ms	166.069 ms	hpr-sdsc-100ge–sdg-hpr3.cenic.net
22	2001:48d0:101:501::122	167.055 ms	169.137 ms	rommie.caida.org

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 e)

Plot the average RTT deltas between consecutive hops from the given traceroute output

3 e)

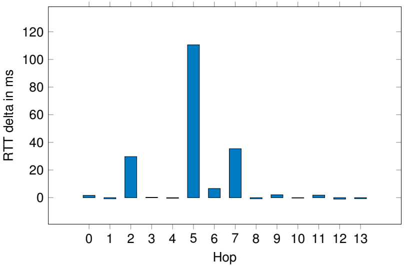
Plot the average RTT deltas between consecutive hops from the given traceroute output

- read values per hop
- average per hop values
- compute diff between hops
- plot bar graph with diffs

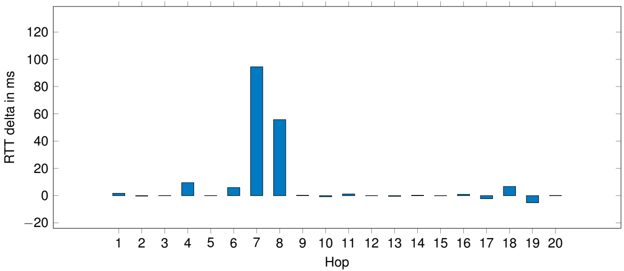
3 e)

Plot the average RTT deltas between consecutive hops from the given traceroute output

- read values per hop
- average per hop values
- compute diff between hops
- plot bar graph with diffs



RIPE traceroute measurements from Germany



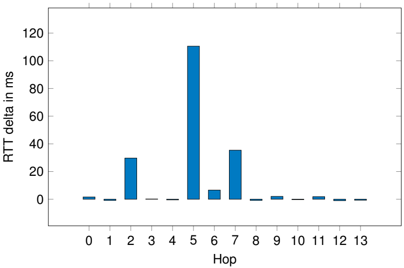
Traceroute measurement from exercise
(this is how your solution should look like)

3 f)

Try to find an explanation for the largest RTT difference calculated in the previous subproblem

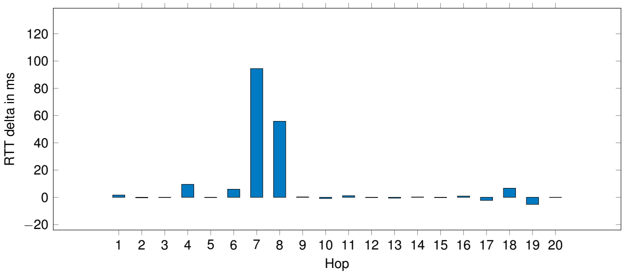
3 f)

Try to find an explanation for the largest RTT difference calculated in the previous subproblem



RIPE traceroute measurements from Germany

- between hop 5 and 7 most probably the transatlantic link is located



Traceroute measurement from exercise
(this is how your solution should look like)

3 g)

Explain why the RTT differences should be always positive in theory. Why does this assumption not always hold in practice?

3 g)

Explain why the RTT differences should be always positive in theory. Why does this assumption not always hold in practice?

- A negative difference means, that the next hop is sending out ICMP responses faster than the previous hop
- There are different possible reason for this seemingly counter intuitive behavior:
  - the slower router performs ICMP throttling
  - Packets generating an ICMP error response have lower priority

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 h)

Try to find out different ways to determine the approximate geographical location of the hops in the given traceroute.

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 h)

Try to find out different ways to determine the approximate geographical location of the hops in the given traceroute.

- GeoIP location services, e.g. Maxmind [1]
- location hints in DNS names - however this information may be wrong
- RTT triangulation
- Crowd sourced data [2]
- Some providers offer additional information: e.g., GEANT [3]
- Some operators offer rough maps of their network, e.g., RedCLARA [4]

[1] https://www.maxmind.com/en/geoip-demo
[2] https://ipmap.ripe.net/
[3] https://www.geant.org/Networks/Network_Operations/PublishingImages/Pages/GEANT_Operations_Centre/GOC-2019013102-GEANT%20ticket%20city%20codes%20table.pdf
[4] https://www.redclara.net/index.php/en/red/redclara/topologia-actual-de-la-red

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 i)

Examples:

- rDNS location hints:
  - `dfn.mx1.fra.de.geant.net`

3 i)

Examples:

- rDNS location hints:
  - `dfn.mx1.`**`fra.de`**`.geant.net`
  - FRA is the IATA airport code for Frankfurt - DE
  - Geant table can also be used

Tutorial2 – Problem 3: Network Topology and Tracing Routes

3 i)

Examples:

- rDNS location hints:
  - `dfn.mx1.`**`fra.de`**`.geant.net`
  - FRA is the IATA airport code for Frankfurt - DE
  - Geant table can also be used
- Maxmind GeoIP:
  - Freely provided data for IPv6 is coarse and only available on a country basis