

Exercise 2

David Koll

dkoll@cs.uni-goettingen.de

CRC checksums

- CRC checksums: Please calculate the CRC R of $D = 0101\ 1100\ 1010\ 0111\ 1110\ 1111$. Please use the 4 bit generator $G = 1101$.
- Please note, R is always of polynom-length (=if G has 4 bit, R is 3 bit long).

$$G = 1 \cdot x^3 + 1 \cdot x^2 + 0 \cdot x^1 + 1 \cdot x^0$$

CRC Checksums

just lines to help:
do not lose the
correct column!

0101110010100111111101111000

1101

0110

1101

1101

0000

001010

1101

01110

1101

001111

1101

001011

always start with the
leading 1

CRC Checksums

01011100101001111110	1111	000
	1011	
	<u>1101</u>	
	01101	
	<u>1101</u>	
	0000	
	1011	
	<u>1101</u>	
	01101	
	<u>1101</u>	
	0000	1000
		<u>1101</u>
		0101

Remainder = CRC = 101

only 3 bits!

Identifiers

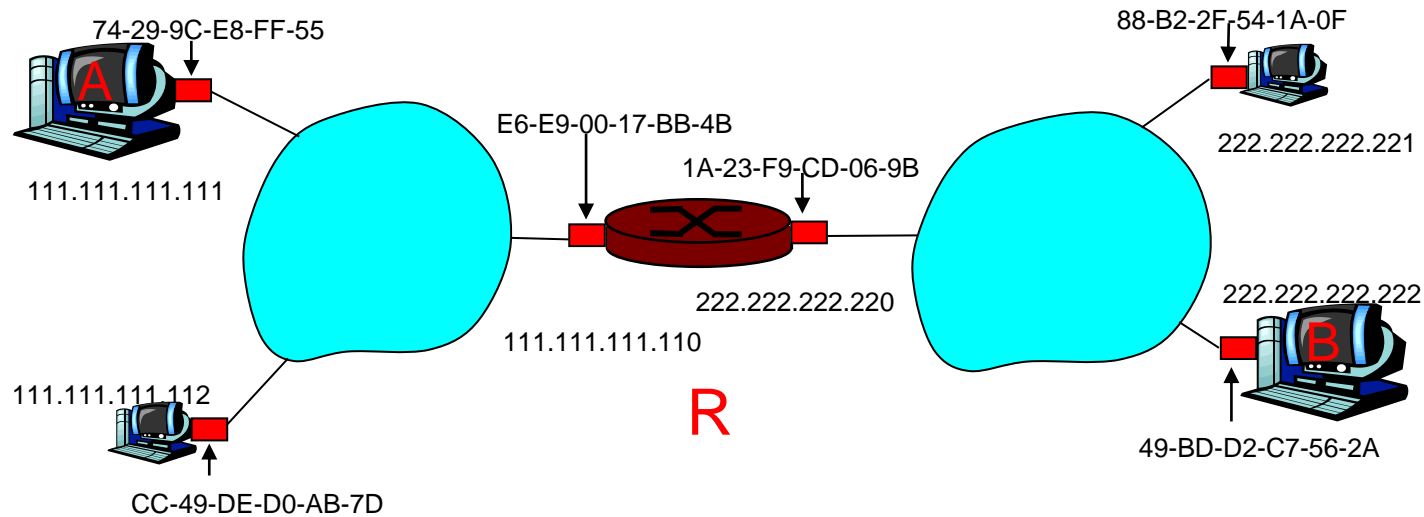
- Why do we need multiple identifiers for one entity such as IP-addresses, MAC addresses etc.?
- Answer: Multiple layers, transparent, nevertheless currently development to split e.g. ID and topological location

ARP

- Please look into the Ethernet frames using Wireshark and, in the best case scenario, observe an ARP request. What happens, if you want to connect to a host that is not in your local area network?
- Was a bit a trick question ;) Remember: ARP is layer 2, routers are layer 3. That is the job of the network layer, to connect different “broadcast domains” where ARP works.
- Answer: By having the router R as a default route in host A, host B is contacted via R so R’s MAC is looked up with ARP! Remember the example page.

- A creates IP datagram with source A, destination B
- A uses ARP to get R's MAC address for 111.111.111.110
- A creates link-layer frame with R's MAC address as dest, frame contains A-to-B IP datagram
- A's NIC sends frame
- R's NIC receives frame
- R removes IP datagram from Ethernet frame, sees its destined to B
- R uses ARP to get B's MAC address
- R creates frame containing A-to-B IP datagram sends to B

This is a **really** important example – make sure you understand!



Wireshark

- A really nice tool to play with network interfaces
- Can observe all traffic on an interface
- Able to do *packet-level* analysis
- Side note: also very helpful in research, e.g., overhead of a new protocol?
- <http://www.wireshark.org>

Intel(R) 82566DM-2 Gigabit Network Connection: Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: **arp** Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
2	0.013942	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.2.3? Tell 134.76.81.1
3	0.858764	IntelCor_5e:fe:d1	Broadcast	ARP	who has 172.22.255.254? Tell 172.22.0.65
4	1.016046	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.2.3? Tell 134.76.81.1
5	1.674481	Dell_e7:c5:a6	Intel_a5:9b:e5	ARP	who has 172.23.255.254? Tell 172.23.0.9
6	1.674601	Intel_a5:9b:e5	Dell_e7:c5:a6	ARP	172.23.255.254 is at 00:04:23:a5:9b:e5
8	2.016158	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.2.3? Tell 134.76.81.1
19	3.697043	Cadmusco_4b:e4:ad	Broadcast	ARP	who has 172.21.255.254? Tell 172.21.201.1
20	3.852308	Intel_a5:9b:e5	Broadcast	ARP	who has 172.23.100.1? Tell 134.76.81.1
21	3.853751	Fuji-xer_3f:b9:d4	Broadcast	ARP	who has 172.23.255.254? Tell 172.23.100.1
24	4.988436	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.0.19? Tell 134.76.81.1
25	5.988509	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.0.19? Tell 134.76.81.1
27	6.988622	Intel_a5:9b:e5	Broadcast	ARP	who has 172.22.0.19? Tell 134.76.81.1
28	7.772687	Intel_a5:9b:e5	Dell_e7:c5:a6	ARP	who has 172.23.0.9? Tell 172.23.255.254
29	7.772708	Dell_e7:c5:a6	Intel_a5:9b:e5	ARP	172.23.0.9 is at 00:1a:a0:e7:c5:a6
30	8.560753	Intel_a5:9b:e5	Broadcast	ARP	who has 172.21.100.1? Tell 134.76.81.1
31	8.562426	Fuji-xer_3f:f4:ed	Broadcast	ARP	who has 172.21.255.254? Tell 172.21.100.1
39	14.141262	Inventec_59:55:43	Broadcast	ARP	who has 172.20.255.254? Tell 172.20.1.5
164	48.177144	Dell_e7:c5:a6	SamsungE_2e:0d:57	ARP	who has 172.23.100.2? Tell 172.23.0.9
165	48.177351	SamsungE_2e:0d:57	Dell_e7:c5:a6	ARP	172.23.100.2 is at 00:15:99:2e:0d:57

Frame 21 (60 bytes on wire, 60 bytes captured)
 Ethernet II, Src: Fuji-xer_3f:b9:d4 (08:00:37:3f:b9:d4), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
 Address Resolution Protocol (request)

```

0000 ff ff ff ff ff 08 00 37 3f b9 d4 08 06 00 01 ..... 7?.....
0010 08 00 06 04 00 01 08 00 37 3f b9 d4 ac 17 64 01 ..... 7?....d.
0020 00 00 00 00 00 00 ac 17 ff fe ff ff ff ff ff .....
0030 ff ff ff ff ff ff ff ff ff ff ff ff .....
  
```

Intel(R) 82566DM-2 Gigabit Network Connec... Packets: 166 Displayed: 20 Marked: 0 Profile: Default