

# Exercise 4

Narisu Tao

[narisu.tao@informatik.uni-goettingen.de](mailto:narisu.tao@informatik.uni-goettingen.de)

# IP as Narrow Waist

Q1: Why is the Internet Protocol (IP) described as the "narrow waist" of the network stack? What are the advantages and disadvantages of such an architecture?

- "IP over anything, anything over IP"
- Single common tie between multiple upper and lower layer protocols
- Innovation in upper/lower layers
- Makes changes to IP difficult (IPv6 ...)

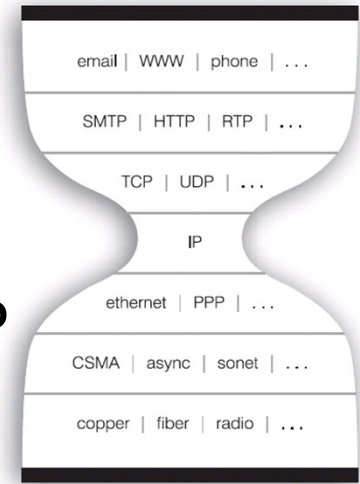


Figure 4.1 Hourglass architecture of the Internet

# Key Functionalities of a Router

Q2: What are the two key functions of the network layer, that each router performs? Please explain the difference between them.

- **Forwarding**: move packets from router's input to appropriate router output
- **Routing**: determine route taken by packets from source to destination.
- A routing protocol determines the forwarding table

# Switching Fabrics

Q3: Which entity residing in a router is responsible for redirecting data from an input port to an output port?  
What are the different types of this entity?

- Switching fabric is responsible

Different types:

- **Memory** used by all ports (limited by memory bandwidth)
- **Bus** used by all ports (limited by bus bandwidth)
- **Crossbar**: Fabric can connect any input port to any output port directly

# Buffering

Q4: What is buffering, where does it occur and what are possible consequences of this situation?

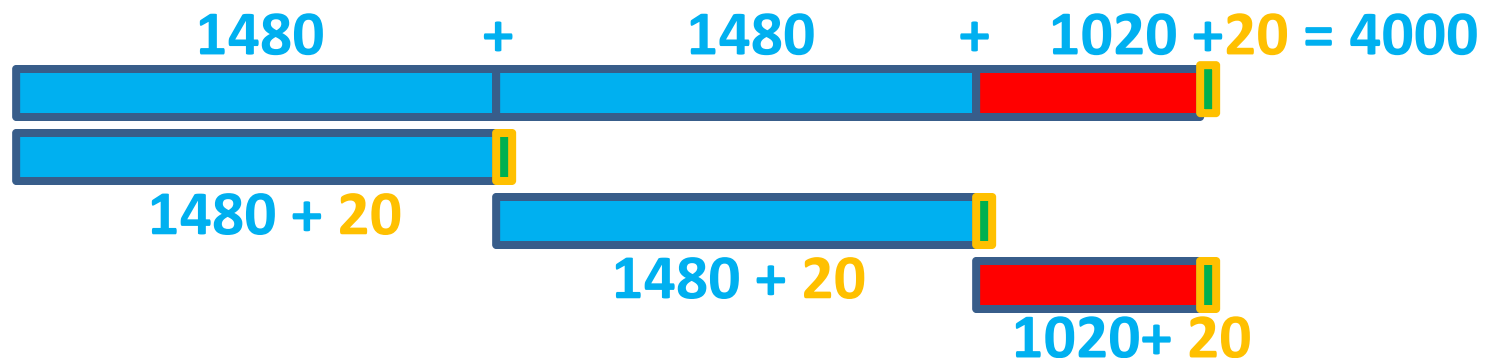
- Required when arrival/departure speed of data **exceeds** fabric speed/transmission rate
- Datagrams that can't be handled directly are stored in a **queue**
- Consequence: **Buffer overflow** if input/output queue are overutilized

# IP Datagram Fragmentation

Q5: Assume you have a 4,000 byte long datagram which needs to be fragmented for a 1,500 bytes MTU. Please fill the following table:

4000 bytes datagram = 20 bytes of IP header + **3980** bytes of IP payload

No.	Length	Frag. Flag	Offset =(MTU-Header / 8)
1	1500 ( <b>1480</b> +20)	1	0
2	1500 ( <b>1480</b> +20)	1	185 (1480 / 8)
3	1040 ( <b>1020</b> +20) =(3980 -1480 - 1480+20)	0	370 (1480*2 / 8)



# IP Address Conversion (Decimal to Binary)

Q6: Convert the following IP addresses into their binary notation:

- TIP: Make yourself a table:

<b>Power</b>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
<b>Value</b>	128	64	32	16	8	4	2	1
<b>Rest</b>								
<b>Bit</b>								

For each octet:

- Put octet number into first “rest” cell
- Bit = (value >= rest ? 1 : 0)
- Restnext = Restprev – Bitprev x Valueprev
- Rinse and Repeat

# IP Address Conversion (Example)

○ First octet of **134**.76.249.227:

<b>Power</b>	$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
<b>Value</b>	128	64	32	16	8	4	2	1
<b>Rest</b>	<b>6</b>	0	0	0	0	<b>2</b>	<b>0</b>	0
<b>Bit</b>	1	0	0	0	0	1	1	0



# Converting IP Addresses

- 134.76.249.227
  - 10000110 01001100 11111001 11100011
- 192.168.0.1
  - 11000000 10101000 00000000 00000001

# IP Address Conversion (Binary to Decimal)

Q7: Convert the following IP address into its decimal notation

Make yourself a table:

<b>Power</b>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
<b>Value</b>	128	64	32	16	8	4	2	1
<b>Bit</b>								
<b>Sum</b>								

For each octet:

- Fill the „Bit“ row with the bits of the octet
- Fill the sum row:  
Sum<sub>next</sub> = Sum<sub>prev</sub> + Bit<sub>prev</sub> x Value<sub>prev</sub>

# IP Address Conversion (Binary to Decimal)

- Octet 11100011:

<b>Power</b>	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>	2 <sup>0</sup>
<b>Value</b>	128	64	32	16	8	4	2	1
<b>Bit</b>	1	1	1	0	0	0	1	1
<b>Sum</b>	128	192	224	224	224	224	226	227

- 11100011100001100000111110101010  
227.134.15.170

# Subnet calculations

- Subnet calculations are used to break a given network into **smaller** pieces
- A (sub-) network mask shows how many bits of an IP address denote the network
- Decimal: /17
- Binary: 11111111.11111111.10000000.00000000
- Hexadecimal: 255.255.128.0

# Subnet calculations (Example)

- Given network: 128.30.0.0/17
- Wanted: Four sub networks
- First step: Find new subnet mask
- To address four networks we need at least two bits ( $2^2 = 4$ ).
- The new subnet mask is  $17+2 = 19$
- Second step: Find new network addresses (see next slide)
- Third step: Calculate data for new networks (see homework)

# Subnet calculations (example)

New netmask: 19 (= 255.255.224.0)

11111111.11111111.11100000.00000000

=> New network 1: 128.30.0.0/19

10000000.00011110.00000000.00000000

=> New network 2: 128.30.32.0/19

10000000.00011110.00100000.00000000

=> New network 3: 128.30.64.0/19

10000000.00011110.01000000.00000000

=> New network 4: 128.30.96.0/19

10000000.00011110.01100000.00000000

Number of hosts:  $2^{13} - 2 = 8,190$

# Subnet calculation (homework)

Q8: A provider has been assigned the network 128.30.0.0/23 and wants to divide it among three customers. Customer A needs to accommodate up to 220 hosts, customer B needs to accommodate up to 110 hosts and customer C needs to accommodate up to 80 hosts. Please fill the following table with the details of the subnetworks that the provider can create to fit its customers' needs.

Subnet No.	Network Address	Netmask	Host Range	No. of Hosts
1 Cust. A	128.30.0.0/24	255.255.255.0	128.30.0.1 – 128.30.0.254	254
2 Cust B	128.30.1.0/25	255.255.255.128	128.30.1.1 – 128.30.1.126	126
3 Cust C	128.30.1.128/25	255.255.255.128	128.30.1.129 – 128.30.1.254	126

# IP Address Allocation - Host

Q9: Consider IP addresses:

1) How does a host get an IP address?

- DHCP
  - Dynamically gets an IP address on joining the network
  - Allows reuse of addresses (address only reserved while online)
- Protocol: DHCP discover → offer → request → ack
  - More details: see lecture slides



# IP Address Allocation - Network

Q9: Consider IP addresses:

2) How does a network get the subnet part of an IP address?

- Allocation of a portion of the providers ISP address space
  - e.g., provider net 200.23.16.0/20
  - Possible allocated subnet: 200.23.30.0/23

# IP Address Allocation - Provider

Q9: Consider IP addresses:

3) How does a provider get a block of IP addresses?

- ICANN (Internet Corporation for Assigned Names and Numbers)
  - Global allocation of addresses to ISPs
  - ISPs then reallocate their addresses to subnets/customers (see previous slides)
  - However: Shortage of IPv4 addresses → Most blocks occupied

# IP Address Allocation -Principle

Q9: Consider IP addresses:

4. What is the principle behind these procedures?

- IP address and IP prefix
- Address aggregation

# Network Address Translation (NAT)

- Q10: What problem is tackled by Network Address Translation (NAT)? Please briefly describe what NAT does and what the NAT traversal problem is.
- IPv4: [Address shortage](#)
- NAT: One network (of an arbitrary number of hosts) has [only one IP address](#) (NAT enabled router) that is accessible from the internet
- The remaining hosts are addressed internally
- Use [port numbers](#) to decide which host the datagram is destined to, mapping inside NAT table
- NAT is often considered a „dirty fix“ to the address shortage issue (→ IPv6)

# IPv4 vs IPv6 - Differences

- Q11: Consider IPv6 What are the main differences between IPv4 and IPv6? What are two approaches towards the transition between IPv4 and IPv6?
- **Address space:** IPv4  $2^{32}$ , IPv6  $2^{128}$
- IPv6: **Fixed header length**, additional information needs to be stored in additional headers
- IPv6: **No packet fragmentation supported**, fragmentation is moved to the sending host
- IPv6: **No header checksum**, error detection on layer 4/2
- ...

# IPv4 to IPv6 - Migration

- There is no “flag day“ on which IPv4 routers are replaced by IPv6 routers.
  - Not all routers can be upgraded **simultaneously**
  - Rather a slow process of transition
  - How to achieve this transition, i.e., a **mixed, concurrent operation** of IPv4 and IPv6 routers?

# IPv4 and IPv6 together

- Two different possibilities
  - **Tunneling**: IPv6 datagram is carried as payload in IPv4 datagram between IPv4 routers; IPv6 routers then decapsulate IPv6 datagram.
  - **Dual Stack**: Routers can do both, IPv4 and IPv6; direct connection between same protocol clients (IPv4 → IPv4, IPv6 → IPv6); can be used together with tunneling