NSWI090: **Computer Networks**

http://www.ksi.mff.cuni.cz/~svoboda/courses/202-NSWI090/

Lecture 9

Addresses and Addressing I

Martin Svoboda
svoboda@ksi.mff.cuni.cz

3. 5. 2021

Charles University, Faculty of Mathematics and Physics

Lecture Outline

Addresses and addressing

- L2: **MAC** addresses
- L3: **IP** addresses
- L4: **port** numbers
- L7: **URI** identifiers

Addressing at L2

MAC addresses (**hardware addresses**)

- Used at L2
  - More precisely at the **MAC sublayer** (**Media Access Control**)
- Assigned to **network interface controllers**
  - I.e., individual L2 interfaces of end nodes as well as routers
  - Bridges / switches only have one MAC address, <u>if any</u>
    - In case they should explicitly be accessible as ordinary devices
    - Not because of fulfilling their L2 tasks (filtering / forwarding)
- Types of addresses
  - Standard **unicast**, but also **multicast** and **broadcast**
- **Different technologies** may have different mechanisms
  - However, **sharing of address spaces** may be desirable
    - So that **Wi-Fi and Ethernet** can coexist in one network

Addressing at L2

MAC addresses (cont'd)

- Must be **locally unique** within a given network
  - So that senders can **identify** the intended recipients
  - And these recipients are able to **recognize** their frames
    - Since everything is / may be delivered to everyone
    - Because all nodes are mutually visible and reachable
- **Assignment strategies**
  - **Locally** administered addresses would suffice
  - However, **globally unique addresses** simplify everything
    - Otherwise newly connected devices would need to be treated
    - So that they do not potentially use conflicting addresses
- ⇒ **globally** unique **burned-in** addresses **assigned by device manufacturers** are primarily used in practice
  - Exceptions exist, though

EUI Addresses

EUI numbering systems (**Extended Unique Identifier**)

- **EUI-48** (48 bits, 6 bytes, $\approx$ 281 trillion addresses)
  - Formerly denoted as **MAC-48**
    - Which is inappropriate since MAC denotes the entire sublayer
    - And so this label should no longer be used since it is obsolete
  - **Notation**
    - Six hexadecimal numbers separated by hyphens or colons
    - E.g.: `FC-77-74-19-41-1E` or `FC:77:74:19:41:1E`
  - Deployment: **Ethernet**, **Wi-Fi**, **Bluetooth**, **ATM**, …
- **EUI-64** (64 bits, 8 bytes)
  - Newer version with larger address space
  - **Conversion** of EUI-48 possible by adding `FF-FE` in the middle
    - E.g.: `FC-77-74-FF-FE-19-41-1E`
  - Deployment: FireWire (IEEE 1394), …

EUI Addresses

Internal structure

- Organizationally Unique Identifier (**OUI**)
  - **Higher 3 bytes** (24 bits)
  - Describes a particular **vendor or manufacturer**
    - E.g.: `FC-77-74` (Intel), `90-F6-52` (TP-Link), …
- **Interface number**
  - Lower 3 bytes (24 bits) in case of EUI-48
  - Lower 5 bytes (40 bits) in case of EUI-64
  - **Serial number** of a given **network interface controller**

OUI Identifiers

OUI component details

- **M bit** (**I/G bit**) = the least significant bit of the first byte
  - 0 = **individual address** (unicast)
  - 1 = **group address** (multicast, broadcast)
- **X bit** (**U/L bit**) = the second least significant bit of the first byte
  - 0 = **Universally Administered Addresses** (UAA)
    - Intended for globally unique addresses
  - 1 = **Locally Administered Addresses** (LAA)
- Examples
  - **Individual** (M = 0) and **universal** (X = 0)
    - `FC-77-74-19-41-1E`: in binary `11111100-`…
  - **Group** (M = 1) and **local** (X = 1)
    - `FF-FF-FF-FF-FF-FF` (L2 broadcast): in binary `11111111-`…

OUI Identifiers

Observations

- Why M and X bits are the **two least significant bits**?
  - **Ethernet** uses **Big Endian** for the individual **bytes**
  - But **Little Endian** for individual **bits** within these bytes
    - Therefore the first two transmitted bits become M and X
    - And so various address modes can be distinguished easily
- Both M and X bits are **set to zeros** in **OUI identifiers** as such
  - Only when they are exploited in EUI addresses…
    - … they can then be changed as required and so their special meaning activated and utilized

OUI Identifiers

Organizationally Unique Identifier (**OUI**) (cont'd)

- Unique **vendor, manufacturer, or other organization** identifier
  - **Purchased** from the **IEEE Registration Authority**
  - One organization may actually have multiple OUIs at a time
- Various usages
  - Our **EUI-48 and EUI-64 addresses**
  - But also **SNAP Protocol Identifiers** (IEEE 802.2 LLC extension)
  - …
- **Online list**: http://standards-oui.ieee.org/oui/oui.txt
  - Current status (April 2021)
    - Almost 30 thousand OUIs are assigned
    - Huawei $\approx$ 1100, Cisco $\approx$ 1060, Apple $\approx$ 890, Samsung $\approx$ 690, …
    - TP-Link $\approx$ 160, Technicolor $\approx$ 80, D-Link $\approx$ 60, Zyxel $\approx$ 40, …
    - More than 17 thousand organizations only have a single OUI

Addressing at L4

Port numbers

- Assigned to **access points** between L4 and L7 in order to allow **end-to-end** communication of individual **application entities**
  - Within a given end node
  - **Separately for each transport protocol** (**TCP**, SCTP, DCCP, **UDP**)
    - Yet in practice usually the same for all these protocols
- Necessary for both **incoming and outgoing** directions
- Allow for the identification of **transport connections**
  - Tuple (sender IP_1:$port_1$, protocol, recipient IP_2:$port_2$)
    - **Target** of the **outgoing** transmission (IP_2:$port_2$, protocol)
    - **Source** of the **incoming** transmission (IP_1:$port_1$, protocol)
- **Requirements**
  - Ports must be **unique, abstract, implicit and static**

Ports

Port numbers

- **16-bit** long integer numbers: permitted **values 0 – 65535**
- System and registered ports maintained by **IANA**
  - **Internet Assigned Numbers Authority**
- Online table of **port numbers** and **service names**
  - https://www.iana.org/assignments/service-names-port-numbers/

Types of ports

- **System Ports** (**Well Known Ports**) (0 – 1023)
  - **Assigned to one purpose** (usually system-oriented services)
  - Should not be used for any other purpose
  - Examples
    - FTP (21), SSH (22), SMTP (25), DNS (53), HTTP (80), POP3 (110), NTP (123), IMAP (143), HTTPS (443), …

Ports

Types of ports (cont'd)

- **User Ports** (**Registered Ports**) (1024 – 49151)
  - **Assigned to one purpose** again
  - But may freely be used for any other purpose
  - Examples
    - Registered: MySQL (3306), PostgreSQL (5432), Redis (6379), Neo4j (7474), MongoDB (27017), …
    - Not registered: Riak (unasigned 8087 and 8098), Cassandra (assigned 7000), …
- **Dynamic Ports** (**Private Ports**) (49152 – 65535)
  - Not assigned, available for unrestricted usage
  - Usually for **outgoing transmissions**

Addressing at L7

Requirements and expectations

- **Various kinds of objects** need to be identified at L7
  - Web pages, files, e-mail addresses, publications, …
- Two aspects actually need to be covered
  - **Identification**
    - So that objects of one kind can mutually be distinguished
    - At least **locally** (within a given end node) but also **globally**
  - **Location**
    - In terms of a particular node where such objects can be found
  - It make sense to logically decouple both these aspects
- **Each application** may have its own proprietary **naming system**
  - Yet it makes sense to pursue **unification** and **coordination**
  - As well as to recycle approaches that already exist

URI Framework

Uniform Resource Identifier (**URI**)

- Generic, federated and extensible **naming system**
  - Allows to identify basically anything
    - Including real-world objects (people, places, concepts, …)
- Types of identifiers
  - **Uniform Resource Locator** (**URL**)
    - **Web resource reference** (web address)
    - Specifies particular location as well as retrieval mechanism
  - **Uniform Resource Name** (**URN**)
    - **Globally unique persistent resource identifier**
    - Does not imply any location, not widely used
  - **Uniform Resource Characteristics** (**URC**)
    - Description of **meta data** about URLs or URNs (citations, …)
    - Never standardized, not even implemented

URI Examples

Sample **URLs**

- `http` scheme (**Hypertext Transfer Protocol**)
  - Addresses of web pages or other resources
  - E.g.: **http**://www.mff.cuni.cz/en/index.php?page=people#5460
- `ftp` scheme (**File Transfer Protocol**)
  - Paths to files or directories accessible using the FTP protocol
  - E.g.: **ftp**://svoboda:password@ulita.ms.mff.cuni.cz/
- `file` scheme
  - Host-specific paths on local or remote **file systems**
  - E.g.: **file**:///home/svoboda/NSWI090/Lecture-03-Layers.pdf
- `mailto` scheme
  - **E-mail addresses** including additional parameters
  - E.g.: **mailto**:svoboda@ksi.mff.cuni.cz?subject=NSWI090

URI Examples

Sample **URLs**

- `tel` scheme
  - **Telephone numbers**
  - E.g.: **tel**:+420-951-554-250
- `sip` scheme (**Session Initiation Protocol**)
  - Participants of multimedia sessions such as **voice calls** (VoIP, …)
  - E.g.: **sip**:martin.svoboda@mff.cuni.cz
- `jdbc` scheme (**Java Database Connectivity**)
  - Connections to **relational databases** from Java applications
  - E.g.: **jdbc**:postgresql://nosql.ms.mff.cuni.cz:5432/database

URI Examples

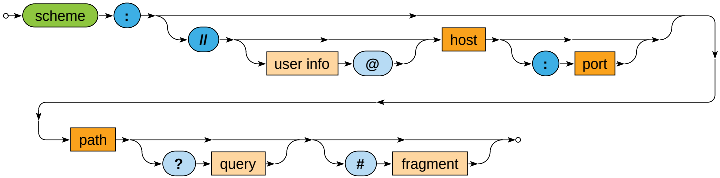
Sample **URNs**

- `isbn` namespace (**International Standard Book Number**)
  - Printed or electronic **books**
  - E.g.: urn:**isbn**:9780132126958
- `issn` namespace (**International Standard Serial Number**)
  - Printed or electronic **serial publications** (journals, …)
  - E.g.: urn:**issn**:0302-9743
- `ietf` namespace (**Internet Engineering Task Force**)
  - IETF family of RFC, STD, FYI, and BCP documents
  - E.g.: urn:**ietf**:rfc:2648
- `iso` (**International Organization for Standardization**)
  - **Standards** and other technical specifications
  - E.g.: urn:**iso**:std:iso-iec:9075:-1:ed-5:en

URI Structure

Generic syntax

- May further be restricted by particular schemes
- **Syntax diagram** (for context-free grammars)



- **Green** boxes: **literals**
  - Expected to be replaced with a particular value
- **Blue** boxes: **constants** (preserved as they are)
- **Orange** boxes: **non-terminals**
  - Unfolded to more complicated fragments

URI Schemes

URI components

- **Scheme**: case-insensitive **scheme name** (usually lower-case)
- **Authority**
  - **User info**: authentication tokens such as **name** and **password**
    - Deprecated for security reasons
  - **Host**: **domain name**, **IP address**, or a different registered name
  - **Port**: transport layer **port**
- **Path**: usually **hierarchical path** with individual segments
- **Query**: usually parameters in a form of **attribute / value pairs**
- **Fragment**
  - Reference to a **secondary resource** related to the primary one
    - E.g.: anchors in HTML pages, classes in OWL ontologies, …

URI Framework

Both **schemes** and **namespaces** are registered with **IANA**

- **URI schemes**
  - https://www.iana.org/assignments/uri-schemes/
  - Current status (April 2021)
    - $\approx$ 100 permanent, 230 provisional, and 10 historical schemes
- **URN namespaces**
  - https://www.iana.org/assignments/urn-namespaces/
  - Current status (April 2021)
    - $\approx$ 70 namespaces

Internationalized Resource Identifier (**IRI**)

- Just an extended version of the traditional URIs
- Allows to use most of **Unicode characters**
  - And so Chinese, Japanese, Korean or other national characters

Addressing at L3

IP addresses

- Primary objective is **addressing of nodes** as a whole
- In spite of that…
  - IP addresses are actually assigned to their **network interfaces**
  - I.e., individual L3 interfaces of end nodes as well as routers
    - Note that end nodes used to usually have only one IP address
    - But nowadays, **multi-homed** hosts with more IPs are common
- Another observation…
  - **Routing algorithms perceive networks as a whole**
  - $\Rightarrow$ **networks themselves** must also be **uniquely identified**
    - Though two separate network / node identifiers could work
    - **Internally structured atomic IP addresses** are more practical
  - Otherwise hop-to-hop **routing** and **forwarding** would not work

IP Addresses

Requirements on IP addresses

- **Globally unique** within the whole system of networks
- And internally structured…

Internal components

- **Network part** (**Network ID**)
  - **Prefix** of the whole address
    - I.e., certain part from the beginning
  - **Uniquely identifies a given network** as a whole
  - Determines affiliation of nodes to a particular network
- **Relative part** (**Host ID**)
  - Remaining part of the whole address
  - **Uniquely identifies a given node** within a given network

Assignment Principles

Observations = rules that must be followed

- **Two nodes in the same network**…
  - Must have the same network parts
  - And different relative parts
    - So that they can be mutually distinguished
- **Two nodes in different networks**…
  - Must have different network parts
    - So that we can detect they belong to different networks
  - And as for relative parts, they do not matter
    - They may be the same as well as not

Assignment Principles

Block principle

- (1) **network as a whole** must first be assigned with a whole contiguous **block of addresses**
  - I.e., set of IP addresses belonging to a specific range
    - All with the same prefix (network part)
  - **Assignment process** must be **globally coordinated**
    - IANA **and regional providers**
- (2) only than **individual nodes** can be given their addresses
  - Of course, from this range
  - Manually or using DHCP **or similar protocols**

Consequence

- **Unused addresses cannot be used by anyone else!**
  - This may lead / actually led to unacceptable wasting

IPv4 Addresses

IPv4 addresses

- **4 bytes** (32 bits) – potentially $\approx$ 4 billion values
  - Which is certainly not much from today's perspective
    - But was initially thought of as generously sufficient
- **Notation**
  - Four decimal numbers, one for each byte, separated by dots
  - E.g.: 195.113.19.170
- **Internal structure**: **network and relative parts**
  - Where the divide between the parts should be located?
    - 3 possible **divide placements** are possible
    - This gives us **Classes A, B and C** of IP addresses
  - If more options exist, though, how to recognize them?
    - Since it must be possible just from the IP address itself

Classes of IPv4 Addresses

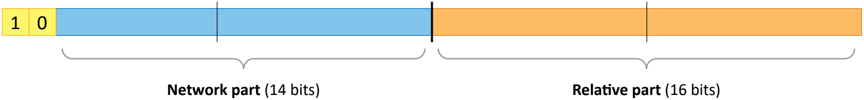
Class A

- Divide is positioned **after the first byte**
  - ≈ 128 networks
  - Each with ≈ 17 million addresses
- Overall **range**: **0**.0.0.0 – **127**.255.255.255
  - Covers 1/2 of the entire space
- **Highest bit** is 0
- Suitable for **very large networks**

Classes of IPv4 Addresses

Class B

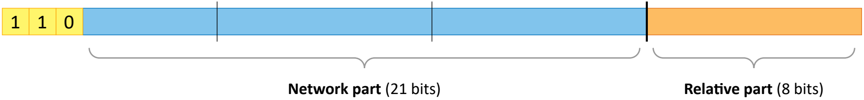
- Divide is positioned **after the second byte**
  - ≈ 16 thousand networks
  - Each with ≈ 66 thousand addresses
- Overall **range**: **128.0**.0.0 – **191.255**.255.255
  - Covers 1/4 of the entire space
- **Highest bits** are 10
- Suitable for **medium-sized networks**



Classes of IPv4 Addresses

Class C

- Divide is positioned **after the third byte**
  - ≈ 2 million networks
  - Each with ≈ 256 addresses only
- Overall **range**: **192.0.0**.0 – **223.255.255**.255
  - Covers 1/8 of the entire space
- **Highest bits** are 110
- Suitable for **very small networks**



Classes of IPv4 Addresses

Apparently the **available space** is not yet fully utilized…

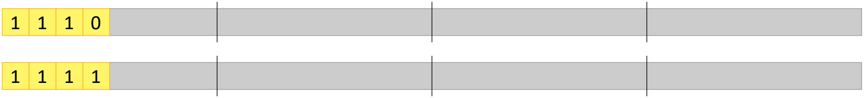
- The rest is covered by Classes D and E
  - They have **specific usage** and they are **internally unstructured**

Class D: **multicast addresses**

- 224.0.0.0 – 239.255.255.255 (1/16 of the entire space)
- Highest bits are 1110

Class E: reserved for **future extensions**, but never used

- 240.0.0.0 – 255.255.255.255 (1/16 of the entire space)
- Highest bits are 1111



Special Addresses

Special addresses

- Certain addresses (even whole ranges) have **dedicated usage**
  - Basic principle: bit 0 **= this**, bit 1 **= all**

Nodes

- Self node: **0.0.0.0**
  - Useful when standard unicast address is not yet known
- Node in a local network: **0**.X.Y.Z, **0.0**.X.Y, or **0.0.0**.X

Networks

- Network as a whole: X.**0.0.0**, X.Y.**0.0**, or X.Y.Z.**0**

Broadcasts

- **Targeted** broadcast: X.**255.255.255**, X.Y.**255.255**, or X.Y.Z.**255**
- **Limited** broadcast: **255.255.255.255**

Special Addresses

Loopback

- 1 A-block: **127**.X.Y.Z
  - **127.0.0.1** is in particular usually used
  - But the whole block is in fact available

Link local addresses

- 1 B-block: **169.254**.X.Y
  - Auto-configuration when standard address cannot be obtained

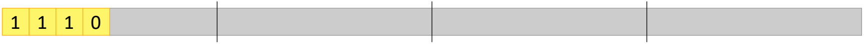
Private addresses

- 1 A-block: **10**.X.Y.Z
- 16 B-blocks: **172.16**.X.Y – **172.31**.X.Y
- 256 C-blocks: **192.168.0**.X – **192.168.255**.X

Multicast Addresses

Multicast transmissions

- Intended recipients are all nodes in a given group
  - This group is predefined or created dynamically on demand
    - **IGMP** (**Internet Group Management Protocol**)
- Certain blocks are assigned and approved by **IANA**
  - https://www.iana.org/assignments/multicast-addresses/
  - Current status (April 2021)
    - $\approx$ 60 static multicast addresses (beside other)
- **Class D addresses** are used
  - Start with 1110 and they are **internally unstructured**
    - I.e., there is no network / relative part



Multicast Addresses

Static groups (**well known** groups)

- Node membership is fixed and given in advance
- Range: 224.0.0.0 – 224.0.0.255 (≈ 256 addresses)
  - Reserved for **routing and other low-level protocols**
    - Topology discovery, maintenance, …
  - **224.0.0.1**: **all hosts** in a given network
  - **224.0.0.2**: **all routers** in a given network
  - …

Dynamic groups

- **Global**: 224.0.1.0 – 238.255.255.255 (≈ 252 million)
  - Group scope can span multiple different networks
- **Local**: 239.0.0.0 – 239.255.255.255 (≈ 17 million)
  - Group scope is limited to a particular network

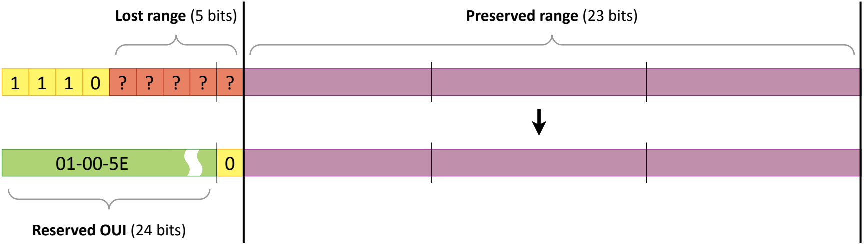
Multicast Addresses

Translation of multicast addresses (**IP** $\rightarrow$ **EUI-48**)

- `L3` multicast **IP address**
  - 4 bits at the beginning are fixed $\Rightarrow$ only **28 bits are relevant**
  - Unfortunately, **only the last 23 bits** can be considered
    - And so the remaining 5 bits in the middle must be truncated
    - $\Rightarrow$ 32 different addresses are mapped to the same value!
- `L2` multicast **EUI-48 address**
  - `00-00-5E` reserved **OUI** is used (24 bits)
    - Of course, altered to `01-00-5E` so that M bit = 1
  - The 25th bit is set to `0`
  - The remaining 23 bits are taken from the original IP address

Multicast Addresses

Translation of multicast addresses (**IP** → **EUI-48**) (cont'd)



Allocation Strategies

Basic principle

- **Whole blocks** of addresses are / must be assigned to networks
- These blocks correspond to our **classes**
  - One **Class A** block $\approx$ 17 million individual addresses
  - One **Class B** block $\approx$ 66 thousand individual addresses
  - One **Class C** block $\approx$ 256 individual addresses

Strategies

- Formerly: **one closest larger block principle**
  - E.g.: 1 Class B address for 1000 requested individual addresses
    - Extremely wasteful approach…
- Later: **multiple closest smaller blocks principle**
  - E.g.: just 4 Class C addresses for the same requested number
    - Better, but still not enough…

Lack of IPv4 Addresses

Temporary **mitigating solutions**

- 1985: **Subnetting**
  - One larger network is divided into separate sub-networks
- 1988: Allocation mechanism
  - **One larger block** $\rightarrow$ **more smaller blocks** principle
- 1993: **CIDR** (**Classless Inter-Domain Routing**)
  - Original concept of IP address classes is entirely dropped
- 1994: **Private addresses**
  - Usage of private IPv4 addresses instead of globally unique ones
  - Requires **NAT** (**Network Address Translation**)

Permanent solution

- 1995: IPv6 protocol and its **IPv6 addresses**
  - **6 bytes** instead of 4 bytes $\Rightarrow$ significantly **larger address space**

Lecture Conclusion

L2: **MAC** addresses

- **EUI-48** and EUI-64 numbering systems, **OUI**, M and X bits

L3: **IP** addresses

- **Network and relative** parts
- **Classes** A, B, C, D, and E
- Special and **multicast** addresses

L4: **port** numbers

- System / User / Dynamic ports

L7: **URI** identifiers

- URI (IRI) framework: **URL** locators, **URN** names