Networking Refresher

Layering and Encapsulation, Routing and Forwarding, Important Protocols

Network Security AS 2020

17 September 2020

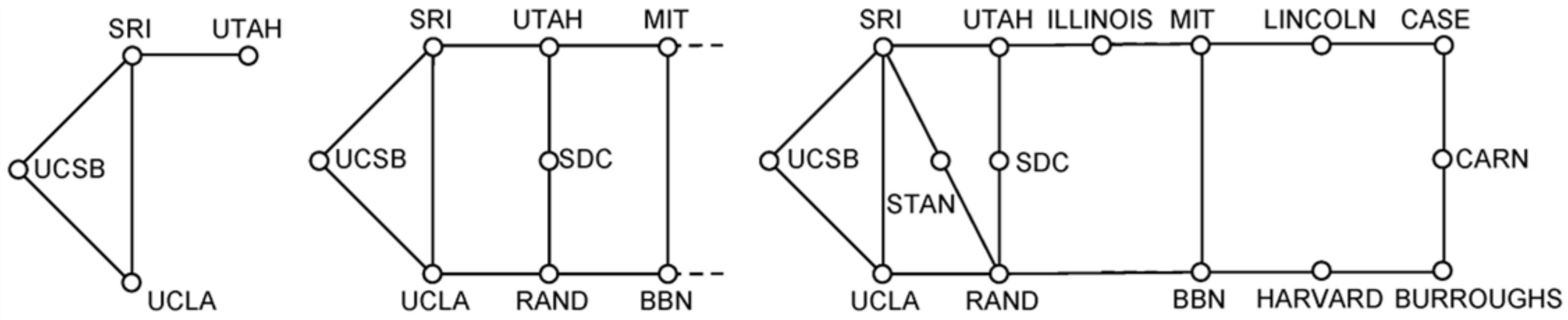
Markus Legner
(based on slides by Adrian Perrig)

Overview

- Brief network summary, based on slides of “Computer Networks” course
  - Website: https://ndal.ethz.ch/courses/networks.html
  - Video recordings: https://video.ethz.ch/lectures/d-infk/2019/spring/252-0064-00L.html
  - High-level summary of networking concepts: last lecture (2019-05-31)
- Slides based on textbook (including chapter references):
  Andrew S. Tanenbaum and David J. Wetherall, *Computer Networks*, 5th Edition, 2011
- Further reading (used in “Computer Networks” course):
  James F. Kurose and Keith W. Ross, *Computer Networking: A Top-Down Approach*

From this experimental network …

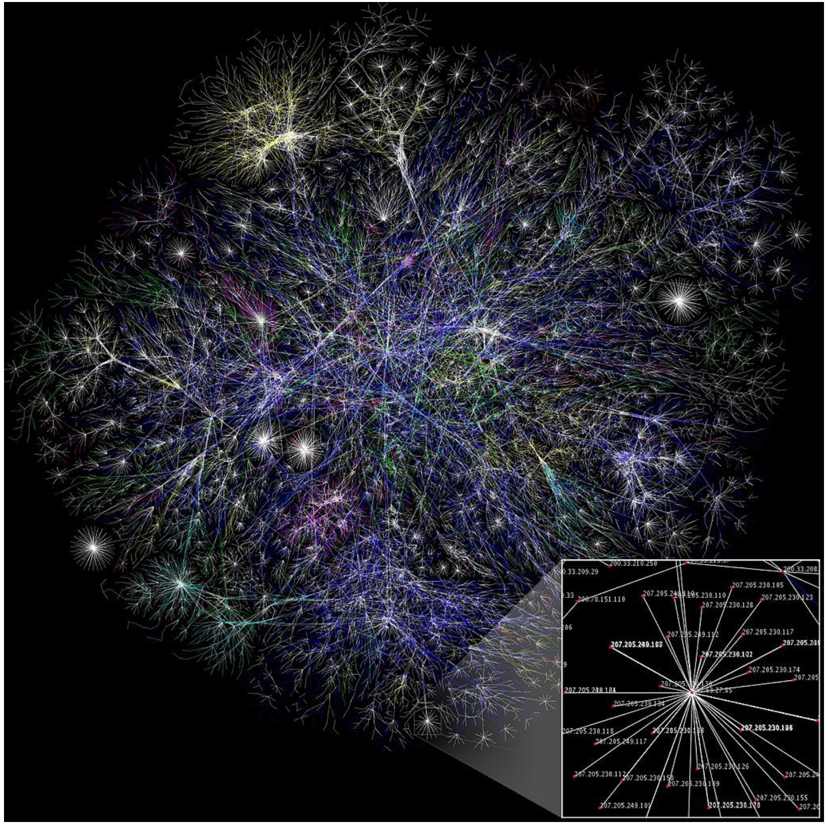
ARPANET ~1970



(a) Dec. 1969. (b) July 1970. (c) March 1971.

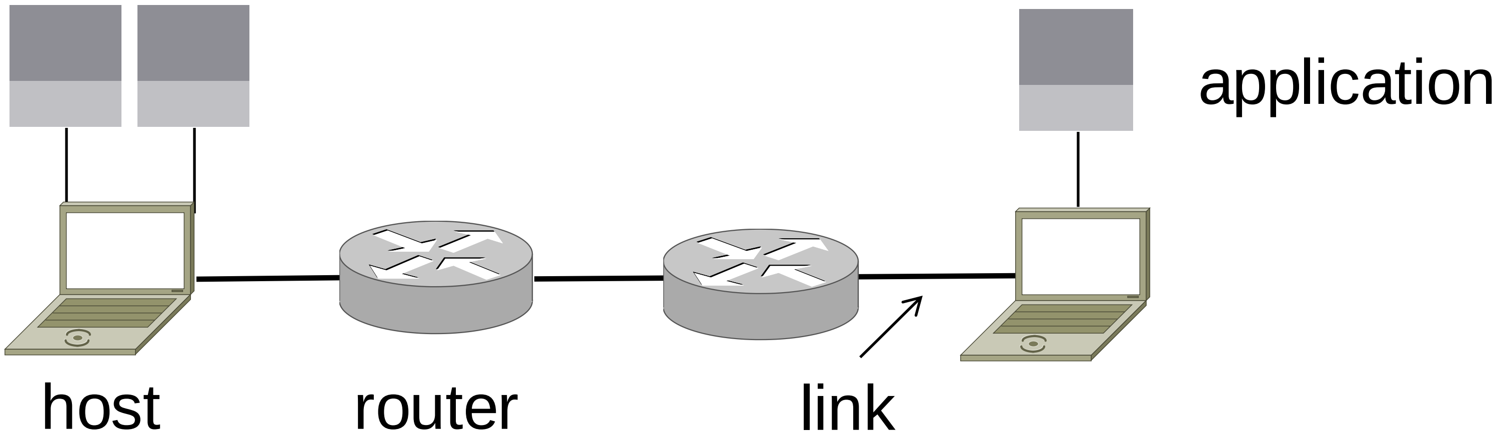
To the current Internet

- An everyday institution used at work, home, and on-the-go
- Visualization (from 2005) contains millions of links



Attribution: By The Opte Project [CC-BY-2.5], via Wikimedia Commons

Parts of a Network

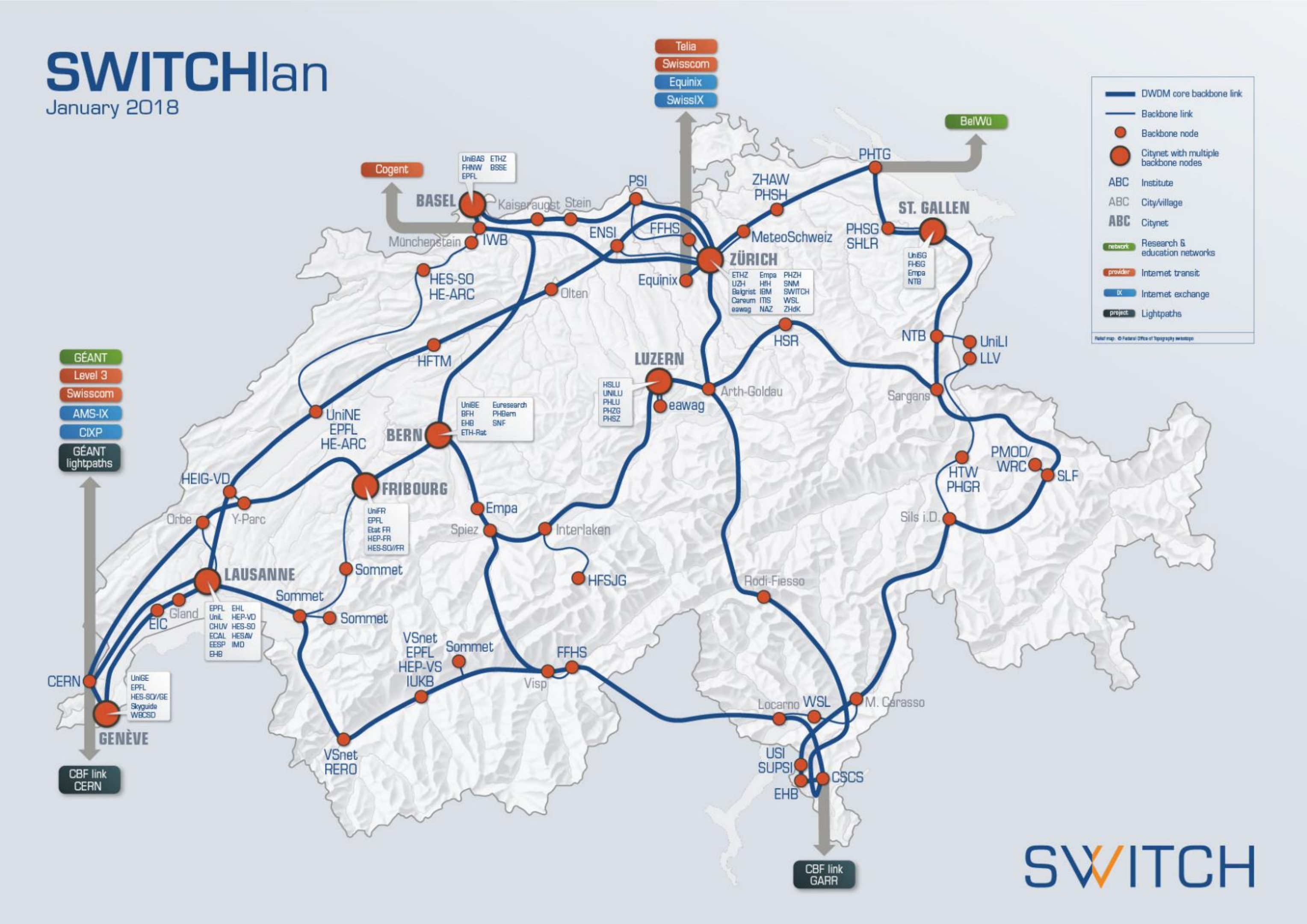


Component Names

Component	Function	Example
application, or app, user	uses the network	Skype, iTunes, Firefox
host, or end-system, edge device, node, source, sink	supports apps	laptop, mobile, desktop
router, or switch, node, hub, intermediate system	relays messages between links	access point, cable/DSL modem, Internet router
link, or channel	connects nodes	wires, wireless

Autonomous Systems (ASes)

- The Internet is a network of networks
  - More than 60’000 autonomous systems (ASes)
    - Internet service providers (ISPs, e.g., Swisscom, Deutsche Telekom)
    - Global backbone networks (CenturyLink, Verizon)
    - Universities, large companies (Google, Cloudflare)
- ASes only exchange information at the edges, internal topology hidden
- Advantage:
  - Internal infrastructure and protocols are independent
  - More efficient routing (see later)


SWITCHlan
January 2018
Telia
Swisscom
Equinix
SwissIX
BelWü
Cogent
DWDM core backbone link
Backbone link
Backbone node
Citynet with multiple backbone nodes
ABC Institute
ABC City/village
ABC Citynet
Research & education networks
Internet transit
Internet exchange
Lightpaths
UniBAS ETHZ FHNW BSSE EPFL
BASEL
Kaiseraugst
Stein
PSI
ZHAW PHSH
PHTG
ST. GALLEN
PHSG SHLR
UniSG FHSG Empa NTB
Münchenstein
IWB
ENSI
FFHS
MeteoSchweiz
ZÜRICH
Equinix
ETHZ Empa PHZH UZH HfH SNM Balgrist IBM SWITCH Careum ITIS WSL eawag NAZ ZHdK
HES-SO HE-ARC
Olten
HSR
NTB
UniLI
LLV
HFTM
LUZERN
HSLU UNILU PHLU PHZG PHSZ
eawag
Arth-Goldau
Sargans
GÉANT
Level 3
Swisscom
AMS-IX
CIXP
GÉANT lightpaths
UniNE EPFL HE-ARC
BERN
UniBE Euresearch BFH PHBern EHB SNF ETH-Rat
HTW PHGR
PMOD/ WRC
SLF
HEIG-VD
FRIBOURG
UniFR EPFL Etat FR HEP-FR HES-SO//FR
Empa
Orbe
Y-Parc
Spiez
Interlaken
Sils i.D.
Sommet
LAUSANNE
HFSJG
Rodi-Fiesso
EPFL EHL UniL HEP-VD CHUV HES-SO ECAL HESAV EESP IMD EHB
Sommet
Gland
EIC
Sommet
VSnet EPFL HEP-VS IUKB
Sommet
FFHS
CERN
UniGE EPFL HES-SO/GE Skyguide WBCSD
Visp
Locarno
WSL
M. Carasso
GENÈVE
VSnet RERO
USI SUPSI
CSCS
EHB
CBF link CERN
CBF link GARR
SWITCH

Important Networking Concepts

Layering

Networks Need Modularity (§1.3)

The network does a lot for applications:

- Make and break connections
- Find a path through the network
- Transfer information reliably
- Transfer arbitrary-length information
- Send as fast as the network allows
- Share bandwidth among users
- Secure information in transit
- Let many new hosts be added
- …

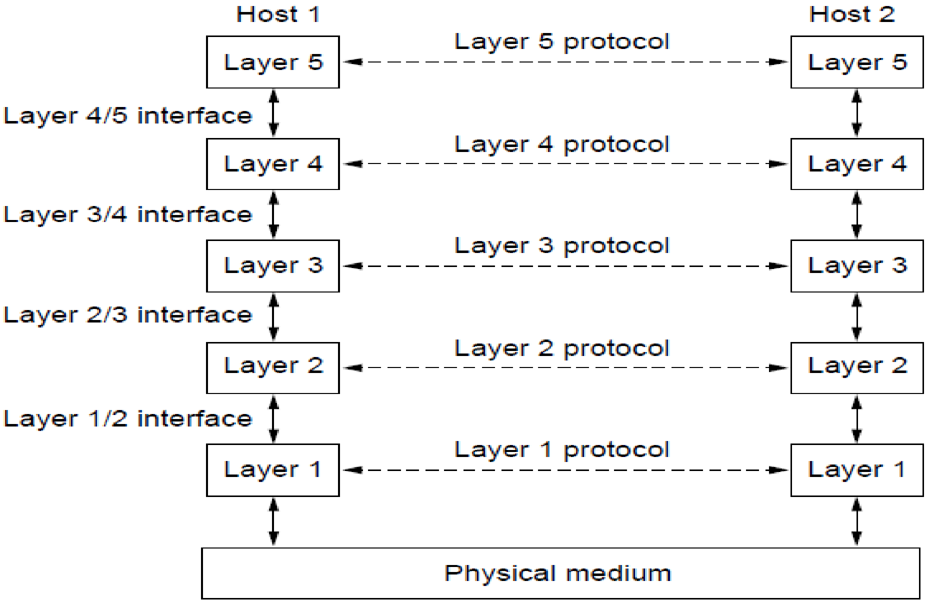
We need a form of modularity, to help manage complexity and support reuse

Protocols and Layers

- Protocols and layering is the main structuring method used to divide up network functionality
  - Each instance of a protocol talks *virtually* to its peer using the protocol
  - Each instance of a protocol uses only the services of the lower layer
  - Protocols *should not* look at data from higher protocols
    - Often violated in practice (see, e.g., NAT later)

Protocols and Layers

- Set of protocols in use is called a protocol stack



OSI "7-Layer" Reference Model

- A principled, international standard, to connect systems
  - Influential, but not used in practice… (Woops)

7	Application	– Provides functions needed by users
6	Presentation	– Converts different data representations
5	Session	– Manages task dialogs
4	Transport	– Provides end-to-end delivery
3	Network	– Sends packets over multiple links
2	Data link	– Sends frames of information
1	Physical	– Sends bits as signals

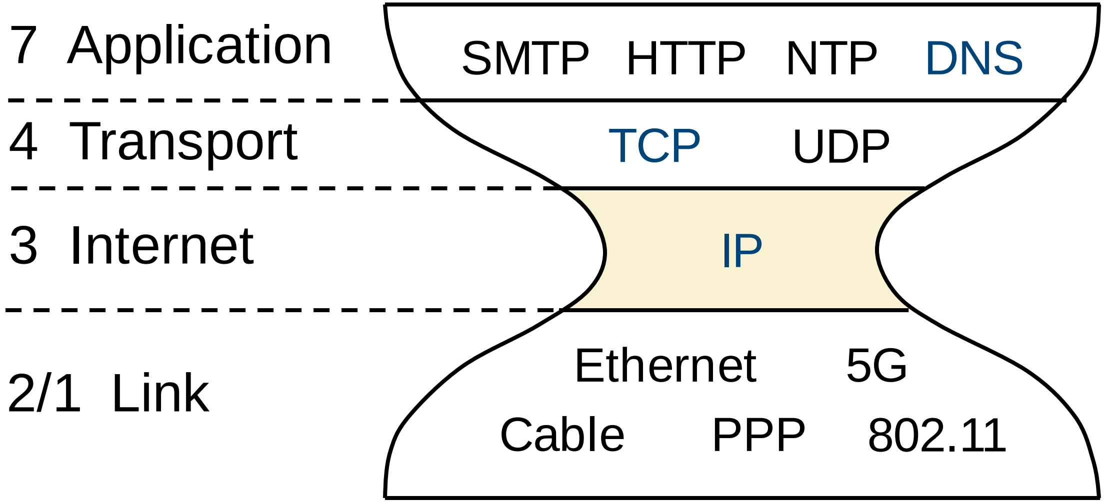
Internet Reference Model

- A four-layer model based on experience
  - Omits/combines some OSI layers and uses IP as the network layer

7: Application	– Programs that use network service
4: Transport	– Provides end-to-end data delivery
3: Internet	– Send packets over multiple networks
1/2: Link	– Send frames over one or multiple links

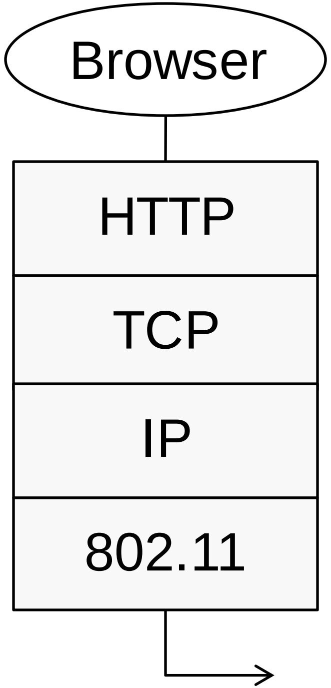
Internet Reference Model

- IP is the “narrow waist” of the Internet
  - Supports many different links below and apps above



Protocols and Layers

- Protocols you've probably heard of:
  - TCP, IP, 802.11 (WLAN), Ethernet, HTTP, SSL/TLS, DNS, … and many more
- An example protocol stack
  - Used by a web browser on a host that is wirelessly connected to the Internet
  - (today, most likely TLS would be used as well → this course)

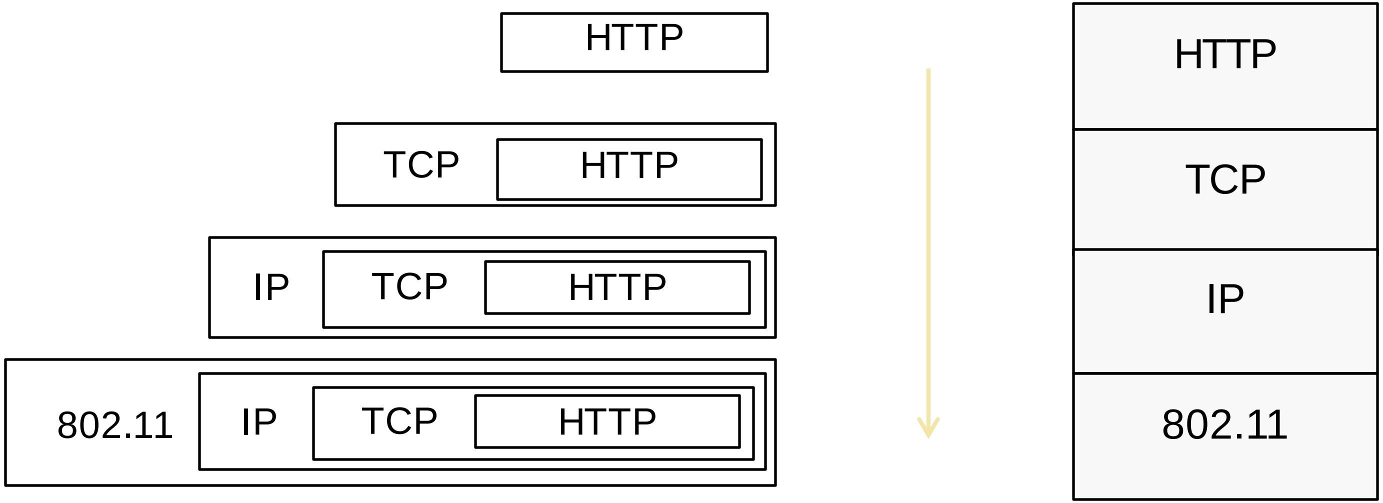


Encapsulation

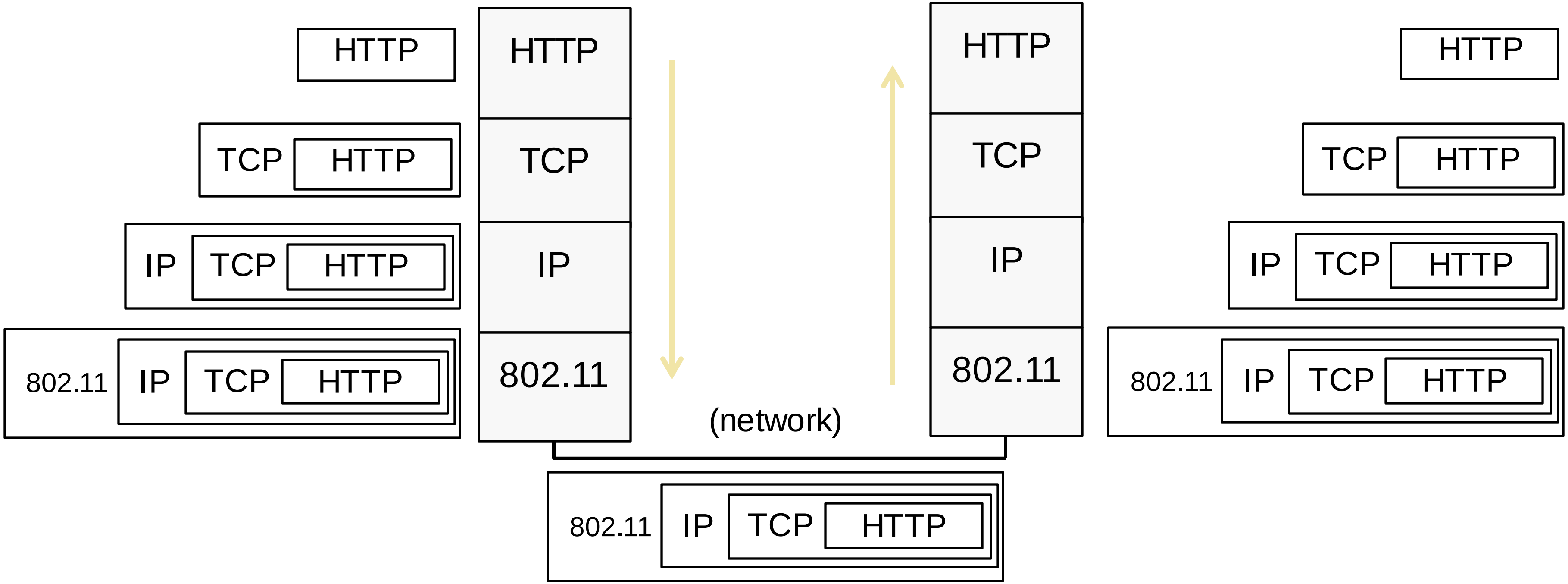
- Encapsulation is the mechanism used to effect protocol layering
- Lower layer wraps higher layer content, adding its own information to make a new message for delivery
- Like sending a letter (*higher* layer) in an envelope (*lower* layer); postal service is not supposed to look inside

Encapsulation

- Message “on the wire” begins to look like an onion
  - Lower layers are outermost

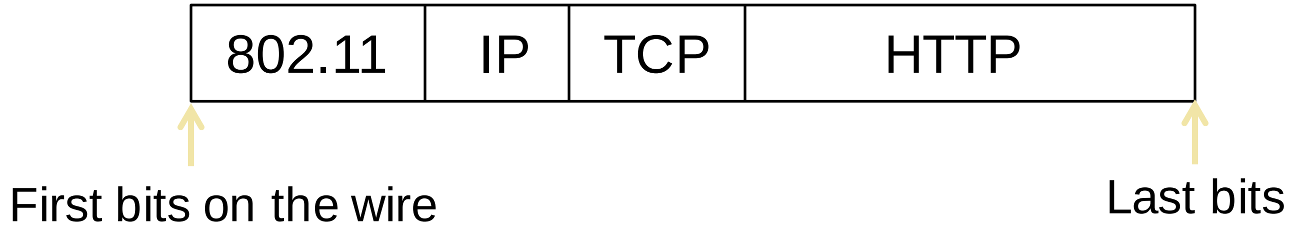


Encapsulation



Encapsulation

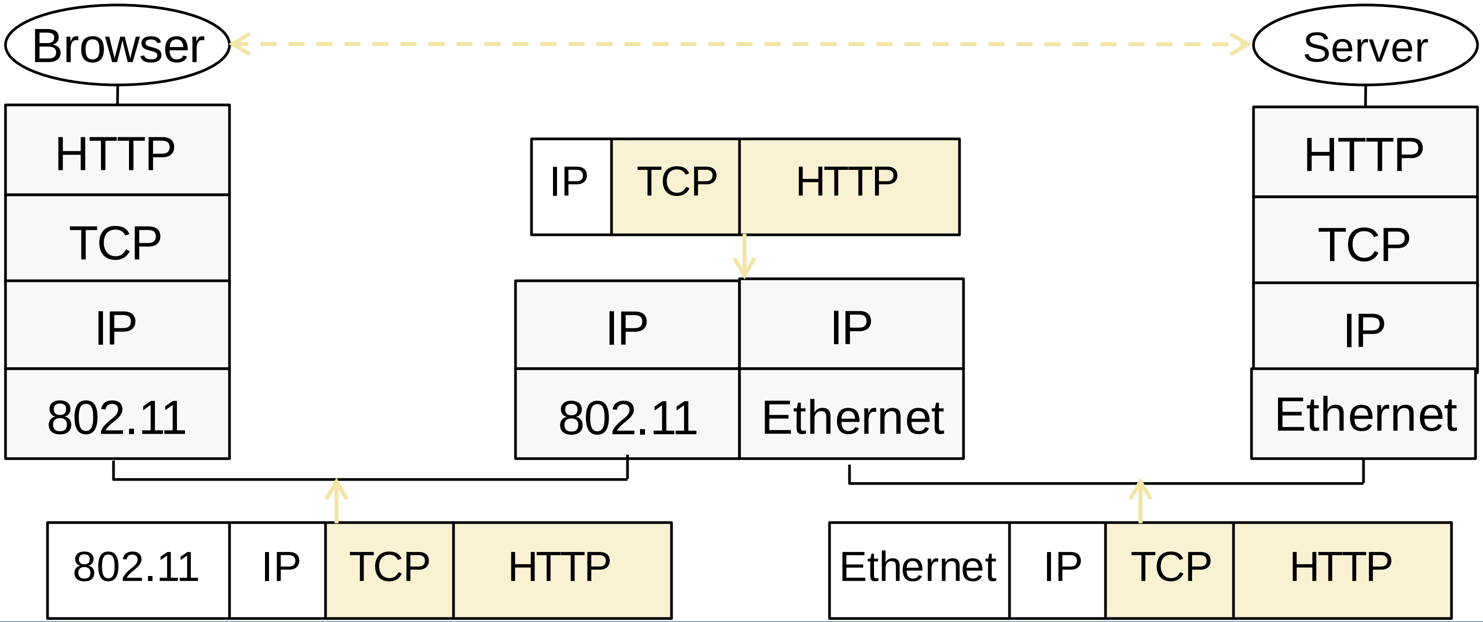
- Normally draw message like this:
  - Each layer adds its own header



- More involved in practice
  - Trailers as well as headers, encrypt/compress contents
  - Segmentation (divide long message) and reassembly

Advantage of Layering

- Higher layers do not need to worry about low-level details
- Possibility to connect different systems



Disadvantage of Layering

- Some optimization potential is lost
- Some protocols do not fit well into a single layer
- → Often layering not strictly followed
  - Network address translation (NAT)
  - TLS
  - QUIC
  - …

Important Networking Concepts

Routing vs. Forwarding

Control Plane vs. Data Plane

- Routing (part of control plane) is the process of discovering best routes in the network (in which direction to send traffic)
  - Network-wide (global), expensive, and (relatively) slow
  - Performed ahead of time and/or in the background
  - Control plane has other components, e.g., key exchange
- Forwarding (data plane) is the process of sending a packet on its
  - Node process (local)
  - Must be very fast: up to billions of packets per second
- Analogy: Training vs. predicting in machine learning

Rules of Routing Algorithms

- Decentralized, distributed setting:
  - All nodes are alike; no controller
  - Nodes only know what they learn by exchanging messages with neighbors, no initial knowledge of topology
  - Nodes operate concurrently
  - There may be node/link/message failures

Distance-Vector (DV) Routing

- Simple, early routing approach
  - Used in ARPANET and RIP (Routing Information Protocol)
- One of two main approaches to routing
- Distributed version of Bellman-Ford algorithm
- Works, but very slow convergence after some failures

Link-State (LS) Routing

- Proceeds in two phases:
  1. Nodes flood topology in the form of link-state packets
     – Each node learns full topology
  2. Each node computes its own forwarding table
     – By running Dijkstra's algorithm (or equivalent)
- Advantage: Faster convergence, support multipath
- Disadvantage: More state and computation required

Path-Vector (PV) Routing

- “Extension” of DV routing
  - Exchanged routing messages include full path
  - Overcomes issues of DV routing
- Example: The Border Gateway Protocol (BGP)

Routing in Today's Internet

- Separate *inter-domain* from *intra-domain* routing
- Intra-domain routing finds paths within one AS
  - Typically link-state routing (OSPF or IS-IS)
- Inter-domain routing finds global paths
  - BGP is the de facto standard
  - We will have a lecture about (non-existent) security of BGP

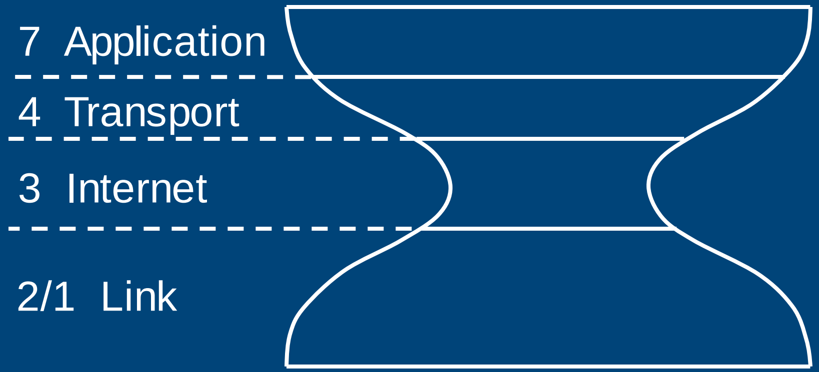
Important Networking Concepts

Reliability ≠ Security

Reliability ≠ Security

- General concept, does not only apply to networking
- Checksums and error-correcting codes can detect accidental errors (reliability)
  - Can be corrected through error correction or retransmission
- Additional mechanisms are necessary to prevent malicious modifications (security)
  - MACs, signatures
- Most original networking protocols only had reliability but not security mechanisms built in

Important Protocols

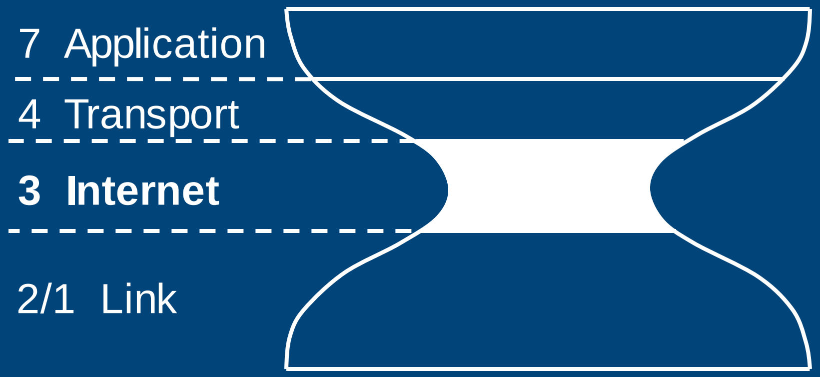


Important Protocols

The Physical/Link Layer Is Less Relevant for This Lecture

- Many different protocols
- Wired connection → physical security
- Wireless connection → lecture “Security of Wireless Networks”: https://syssec.ethz.ch/education/sown/sown_AS20.html

Important Protocols

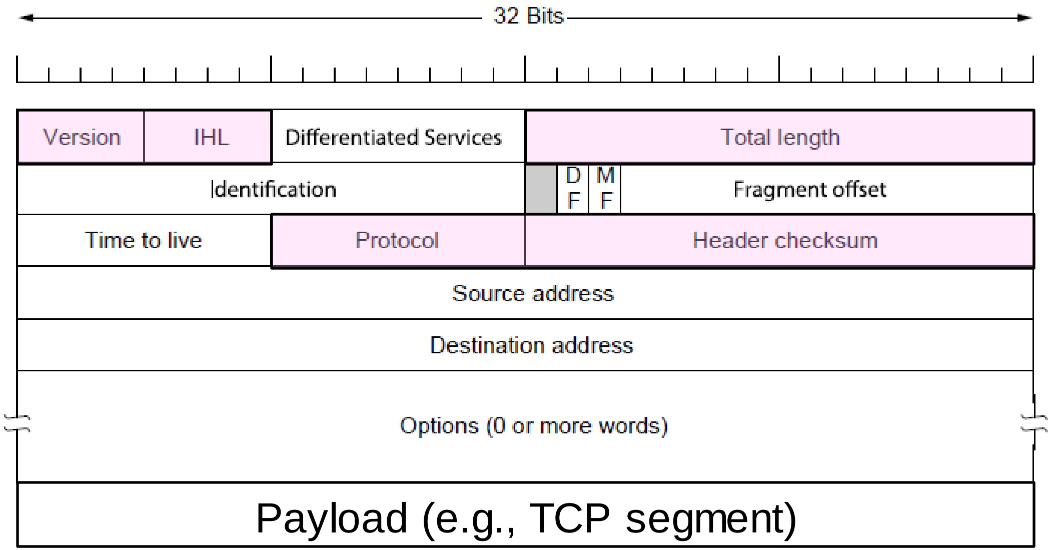


The Internet Protocol

- Internet Protocol (IPv4/IPv6) for packet forwarding (data plane)
  - End hosts specify source and destination
  - No control over paths
  - No guarantees: best-effort traffic
- Internet Control Message Protocol (ICMP) for error messages
- Additional helper protocols:
  - Address resolution protocol (ARP) translates between IP and MAC addresses
  - Dynamic Host Configuration Protocol (DHCP)
  - Network address translation (NAT)
- Border Gateway Protocol (BGP) for global routing (control plane)

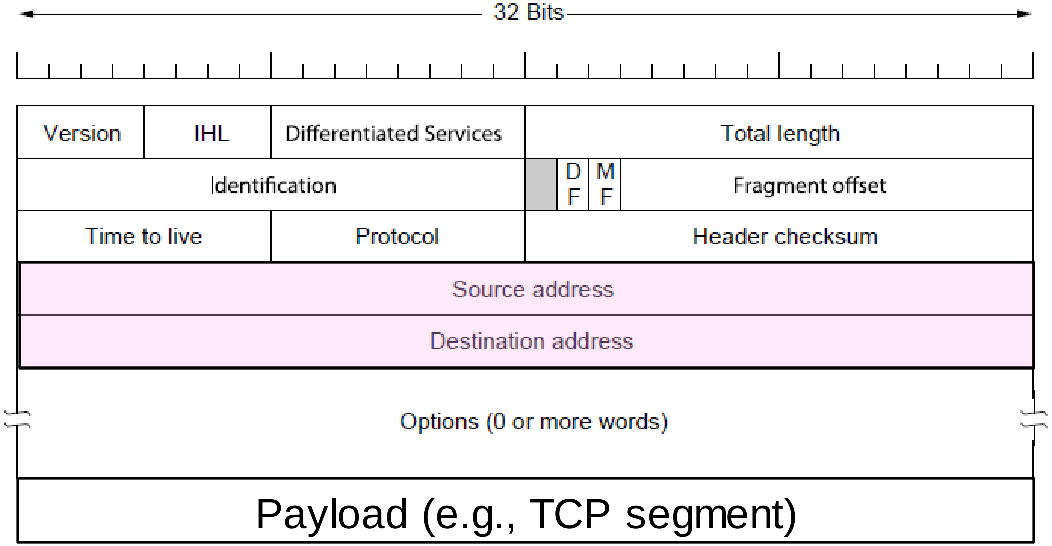
IPv4

- Various fields to meet straightforward needs
  - Version, Header (IHL) and Total length, Protocol, and Header Checksum



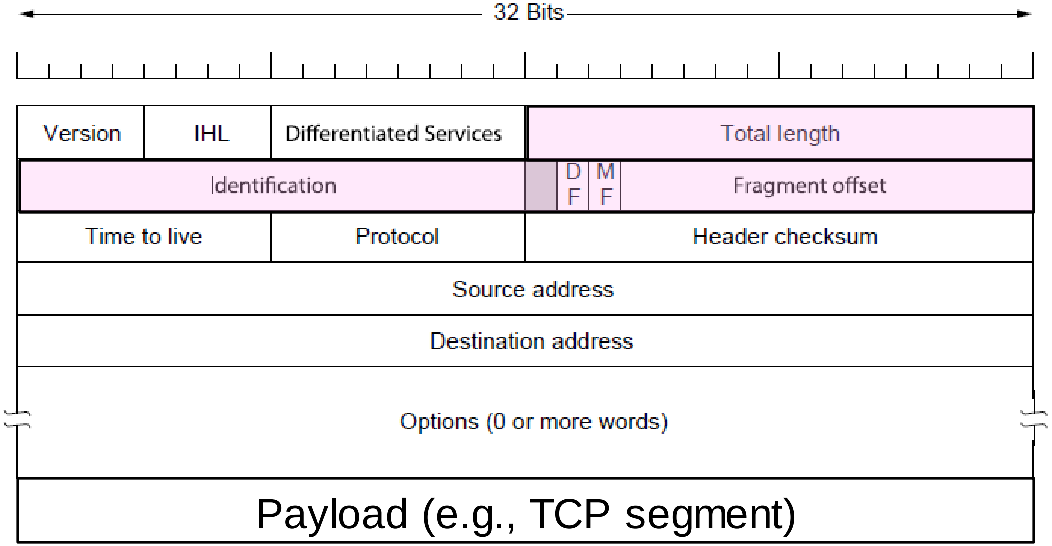
IPv4

- 4-byte IP addresses (independent from link layer)



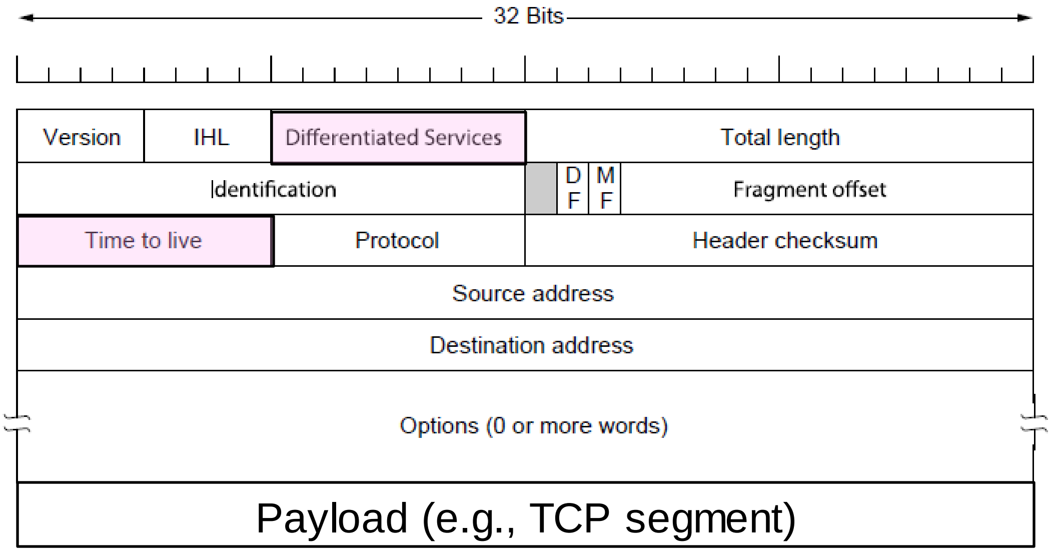
IPv4

- Some fields to handle packet size differences
  - Identification, Fragment offset, Fragment control bits



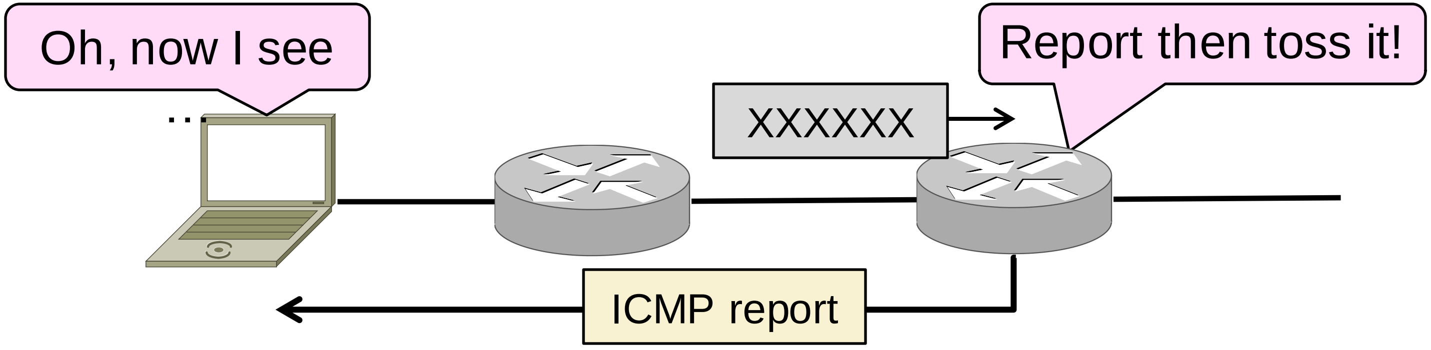
IPv4

- Other fields to meet other needs
  - Differentiated Services, Time to live (TTL)



ICMP Errors

- When router encounters an error while forwarding:
  - It sends an ICMP error report back to the IP source address
  - It discards the problematic packet; host needs to rectify



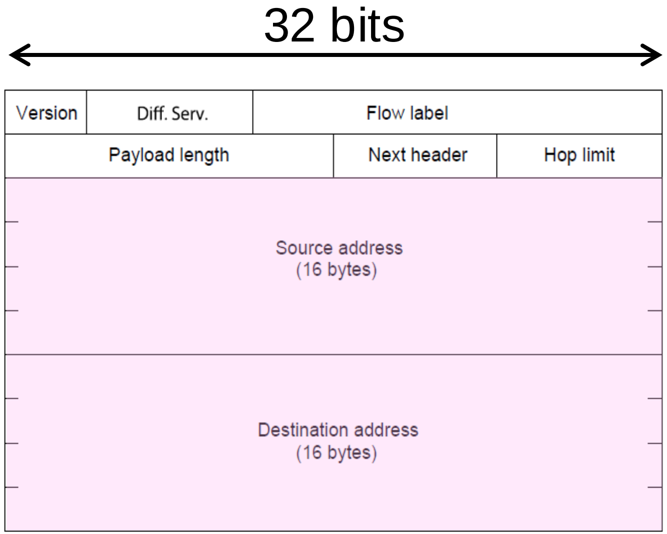
Example ICMP Messages

Name	Type / Code	Usage
Dest. Unreachable (Net or Host)	3 / 0 or 1	Lack of connectivity
Dest. Unreachable (Fragment)	3 / 4	Path MTU Discovery
Time Exceeded (Transit)	11 / 0	Traceroute
Echo Request or Reply	8 or 0 / 0	Ping

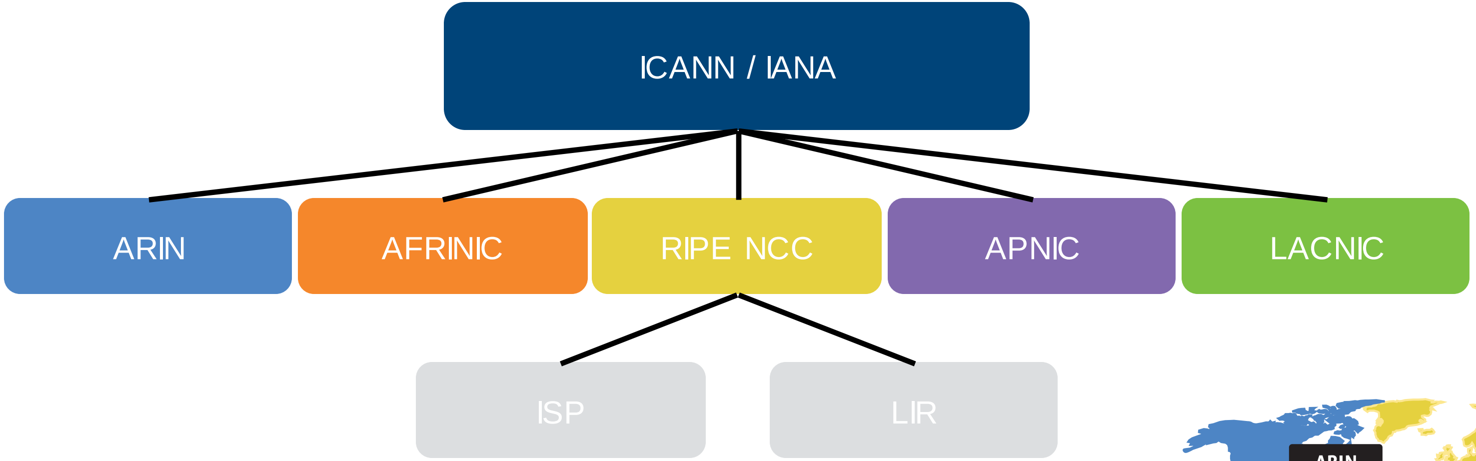
Testing, not a forwarding error: host sends Echo Request, and destination responds with an Echo Reply

IPv6

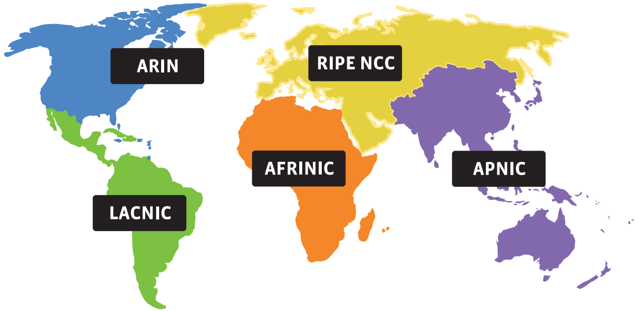
- Features large addresses
  - 128 bits, most of header
- New notation
  - 8 groups of 4 hex digits (16 bits)
  - Omit leading zeros in each group
  - Omit one continuous sequence of zeros
- Example:
  2001:0db8:0000:0000:0000:ff00:0042:8329
  = 2001:db8::ff00:42:8329



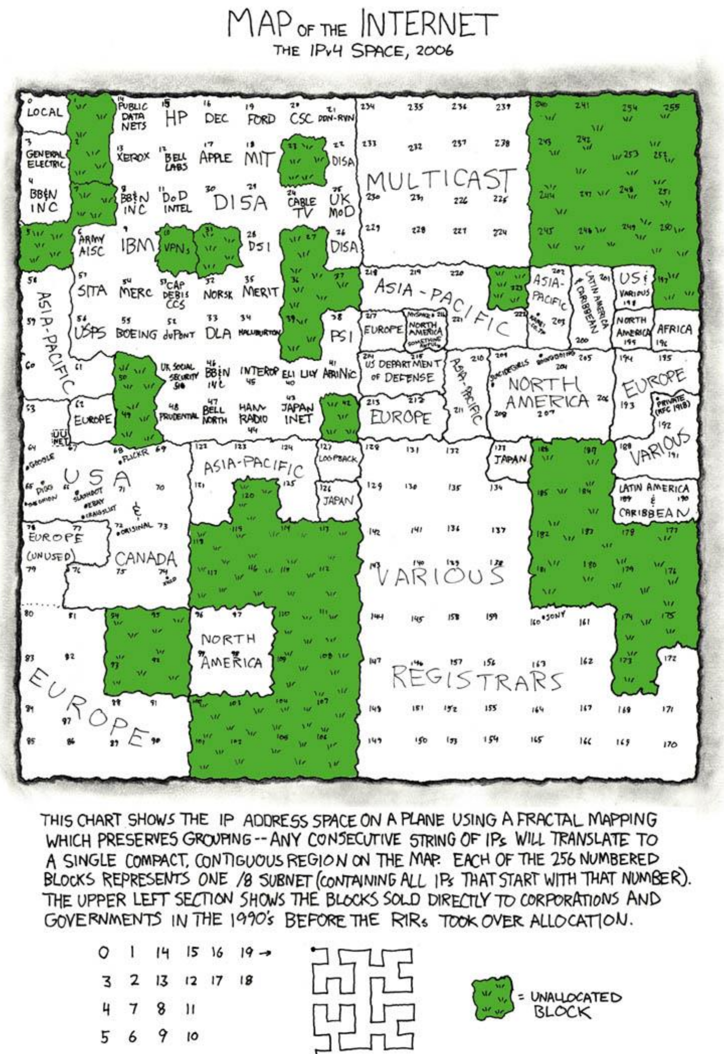
Allocation and Ownership of IP Addresses



Regional Internet Registries (RIRs)

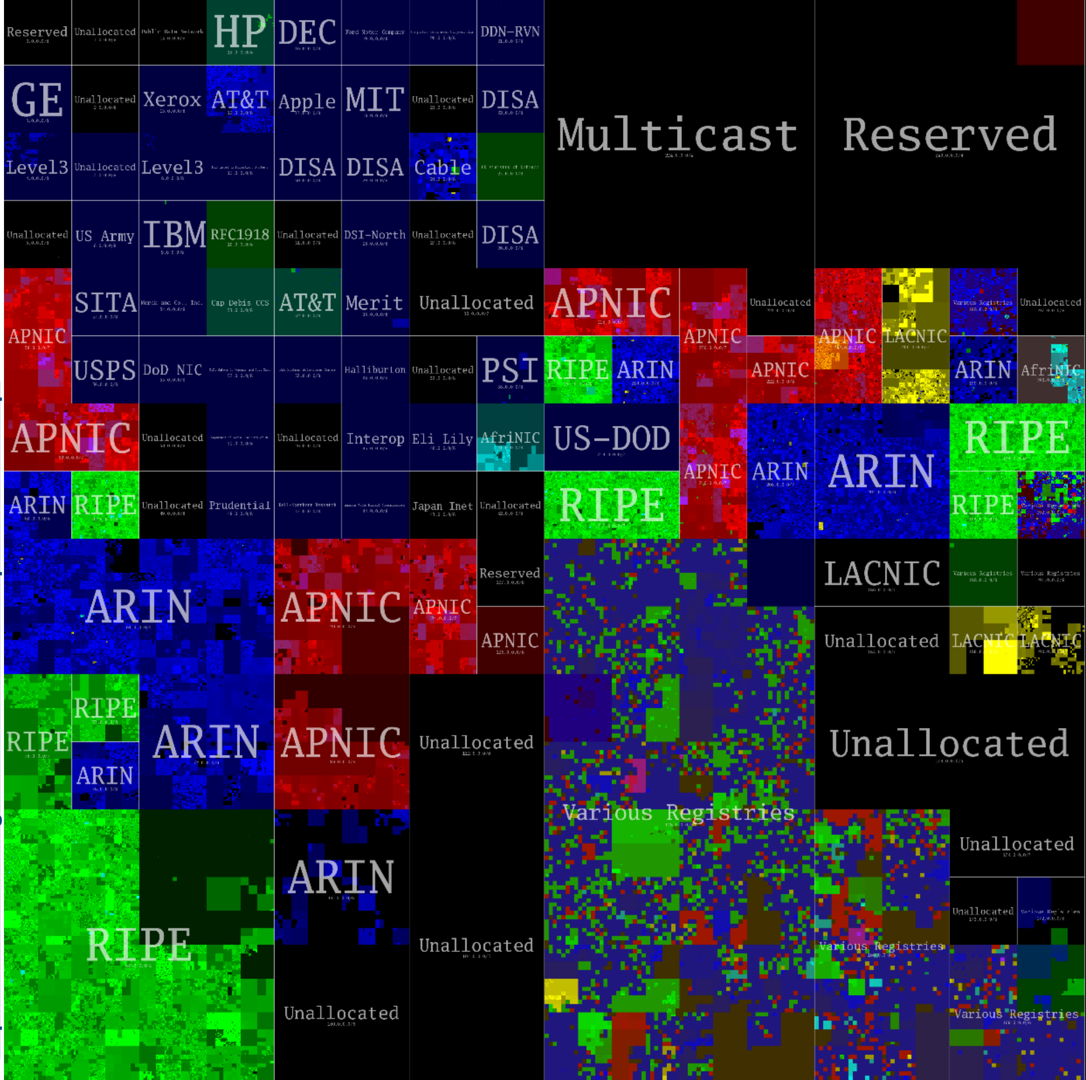


Source: https://www.iana.org/numbers



https://xkcd.com/195/

https://www.caida.org/research/id-consumption/whois-map/

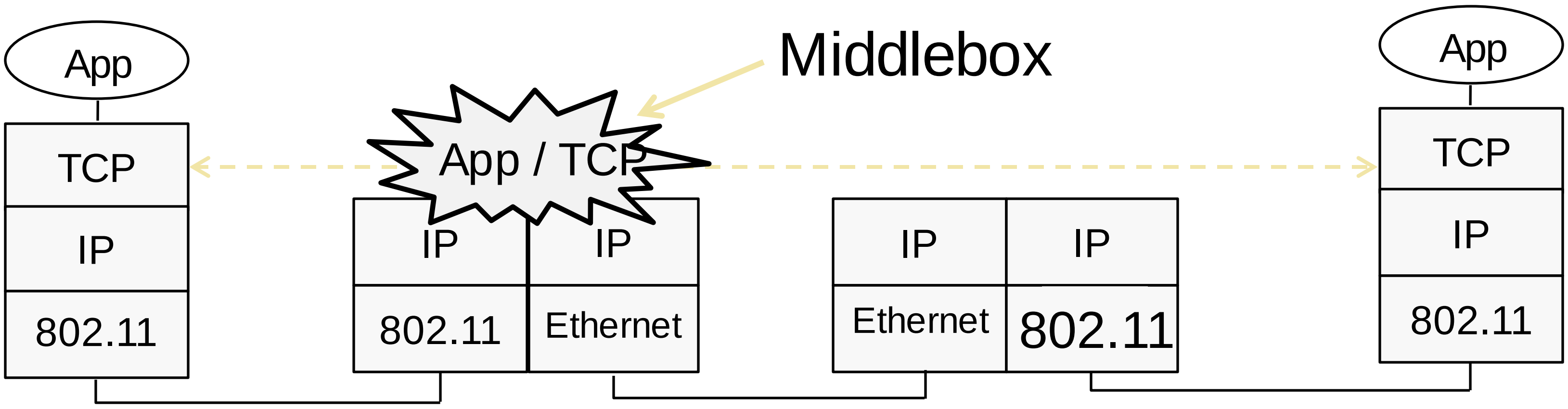


Allocation and Ownership of IP Addresses

- Highest authority: Internet Corporation for Assigned Names and Numbers (ICANN), Internet Assigned Numbers Authority (IANA)
- IP address ownership:
  - ICANN assigns address space to regional Internet registries (RIRs)
  - RIRs assign address space to ISPs or local internet registries (LIRs)
  - LIRs and ISPs assign individual addresses to end customers
- IP address space is allocated in prefixes: "<address>/<prefix length>"
  - 129.132.0.0/16: all IP addresses that start with 129.132 (first 16 bits)
  - Longer prefixes are *more specific* and contain *fewer addresses*

Middleboxes

- Sit “inside the network” but perform “more than IP” processing on packets to add new functionality
  - NAT box, Firewall / Intrusion Detection System



NAT (Network Address Translation)

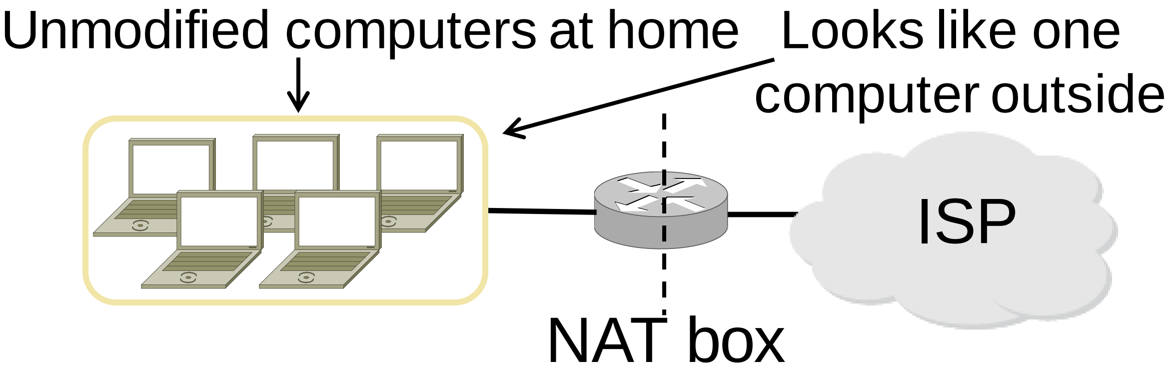
- NAT box connects an internal network to an external network
  - Many internal hosts are connected using few external addresses
  - Middlebox that “translates addresses” (and ports)

- Motivated by IPv4 address scarcity
  - “Today, at 15:35 UTC+1 on 25 November 2019, we made our final /22 IPv4 allocation from the last remaining addresses in our available pool. **We have now run out of IPv4 addresses.**” [RIPE NCC]

- Controversial at first, now widely used and accepted

NAT

- Common scenario:
  - Home computers use “private” IP addresses (RFC 1918, e.g., 192.168.0.0/16)
  - NAT (in home router) connects home to Internet service provider (ISP) using a single external IP address



How NAT Works

- Keeps an internal/external table
  - Typically uses IP address + TCP port (transport layer)
  - This is address and port translation

What host thinks	What ISP thinks
Internal IP:port	**External IP:port**
192.168.1.12 : 5523	44.25.80.3 : 1500
192.168.1.13 : 1234	44.25.80.3 : 1501
192.168.2.20 : 1234	44.25.80.3 : 1502

- Need ports to make mapping 1-to-1 since there are fewer external IPs

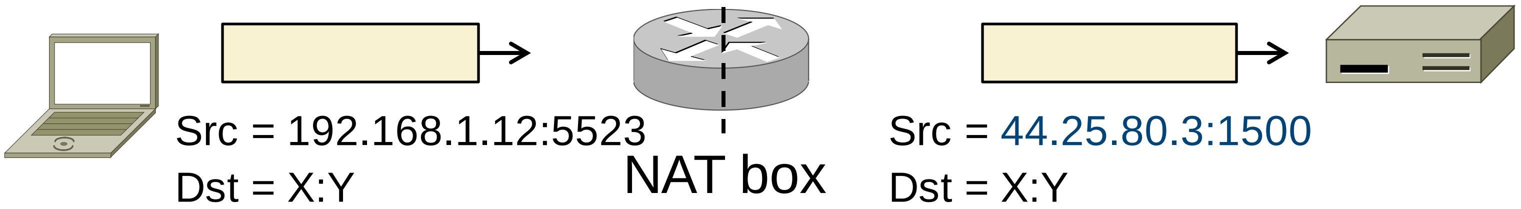
How NAT Works

- Internal → External:
  - Look up and rewrite source IP/port
  - Create new entry if none exists

Internal Source
IP=192.168.1.12, port=5523

Internal IP:port	External IP:port
192.168.1.12:5523	44.25.80.3:1500

External Destination
IP=X, port=Y



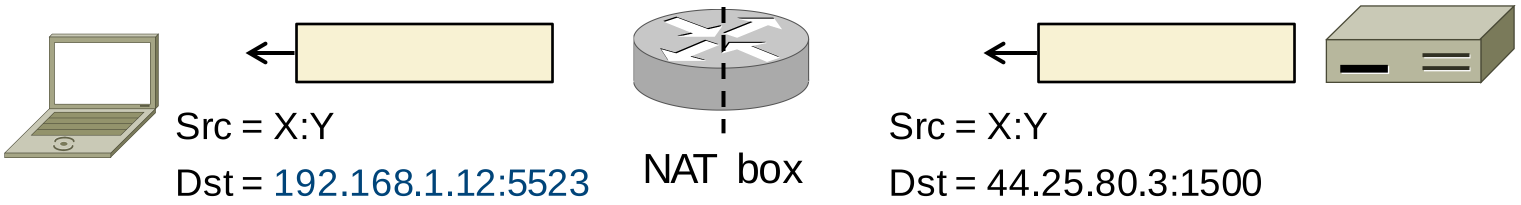
How NAT Works

- External → Internal:
  - Look up and rewrite destination IP/port
  - Block traffic if no matching entry exists
  - Can manually configure port forwarding

Internal Destination
IP=192.168.1.12, port=5523

Internal IP:port	External IP:port
192.168.1.12:5523	44.25.80.3:1500

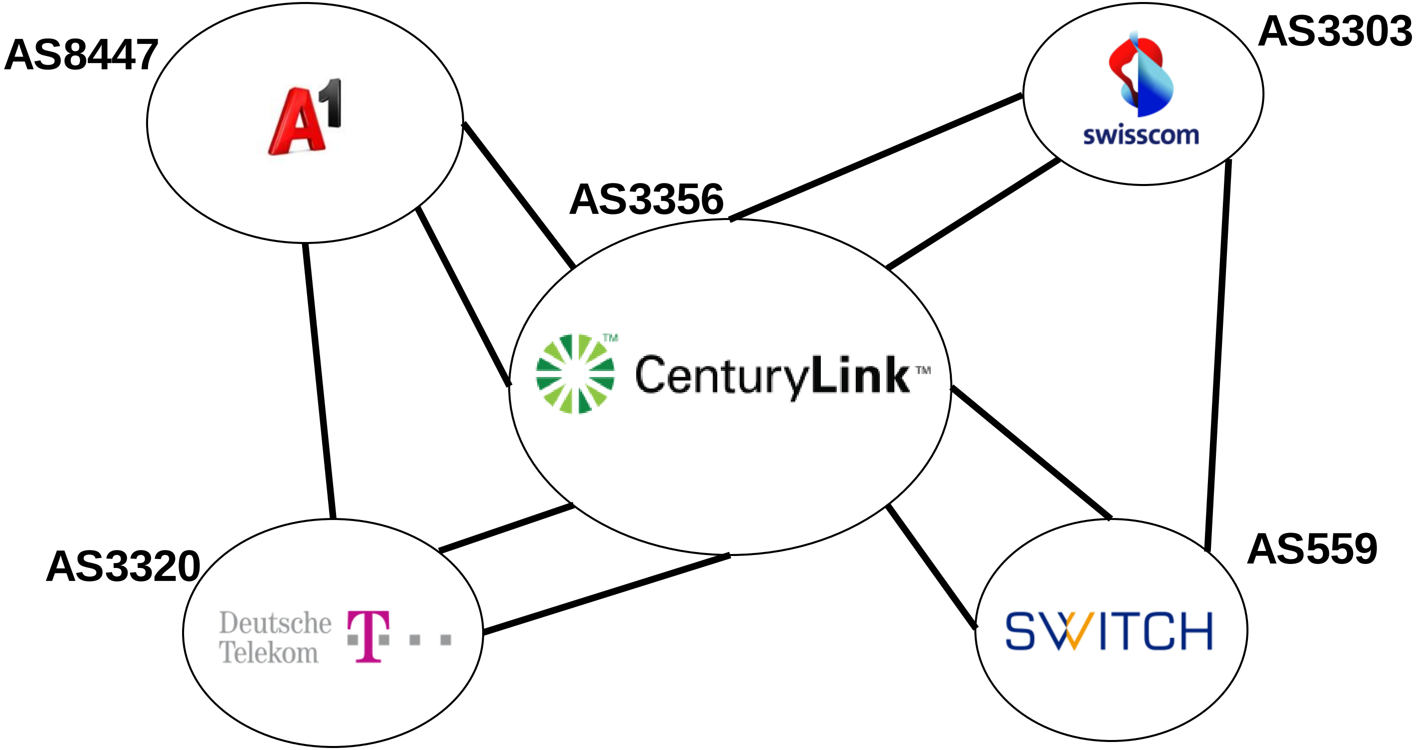
External Source
IP=X, port=Y



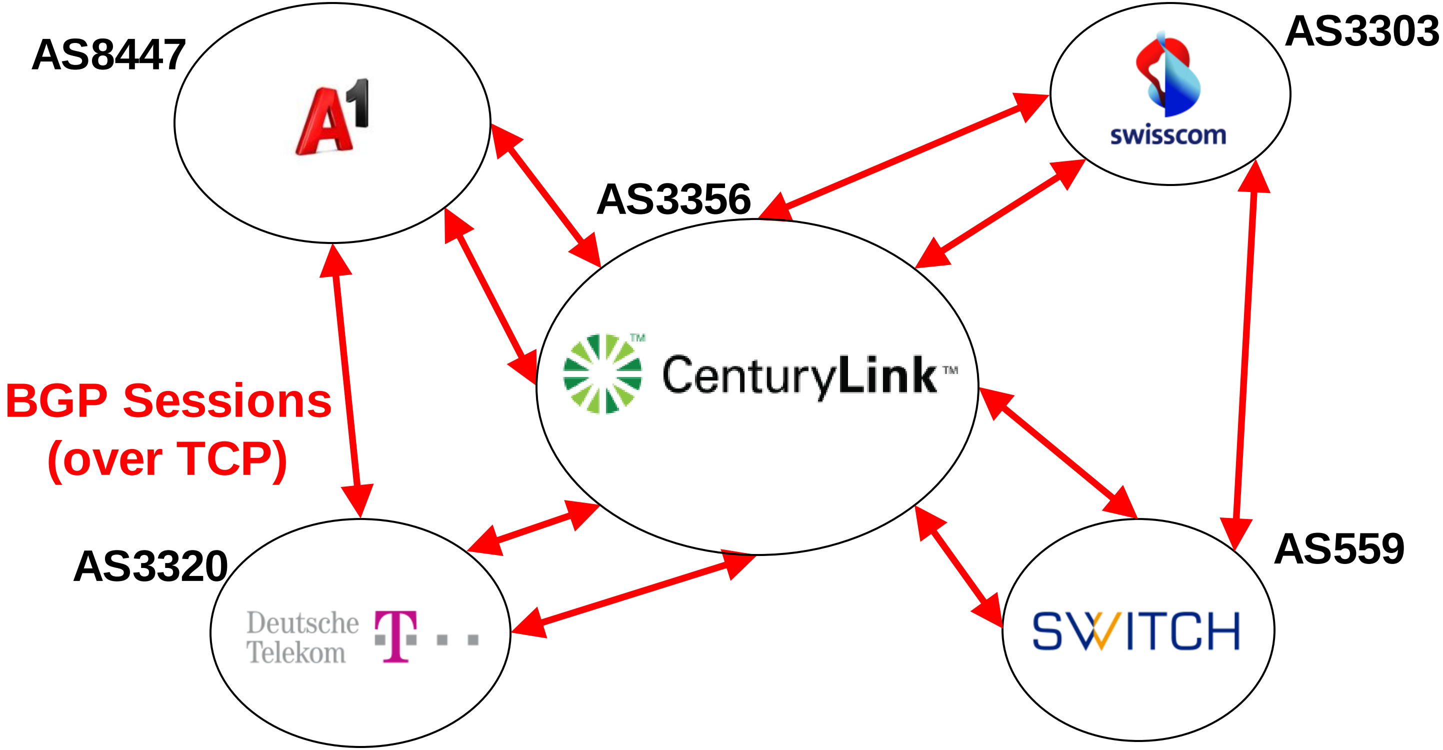
The Internet and BGP

- The Border Gateway Protocol (BGP) “glues” the Internet together
  - The routing protocol between ASes
  - Disseminates information about location and paths for IP prefixes
  - A path-vector protocol
- Business relationships shape topology
  - An AS only forwards traffic where it can earn money from someone

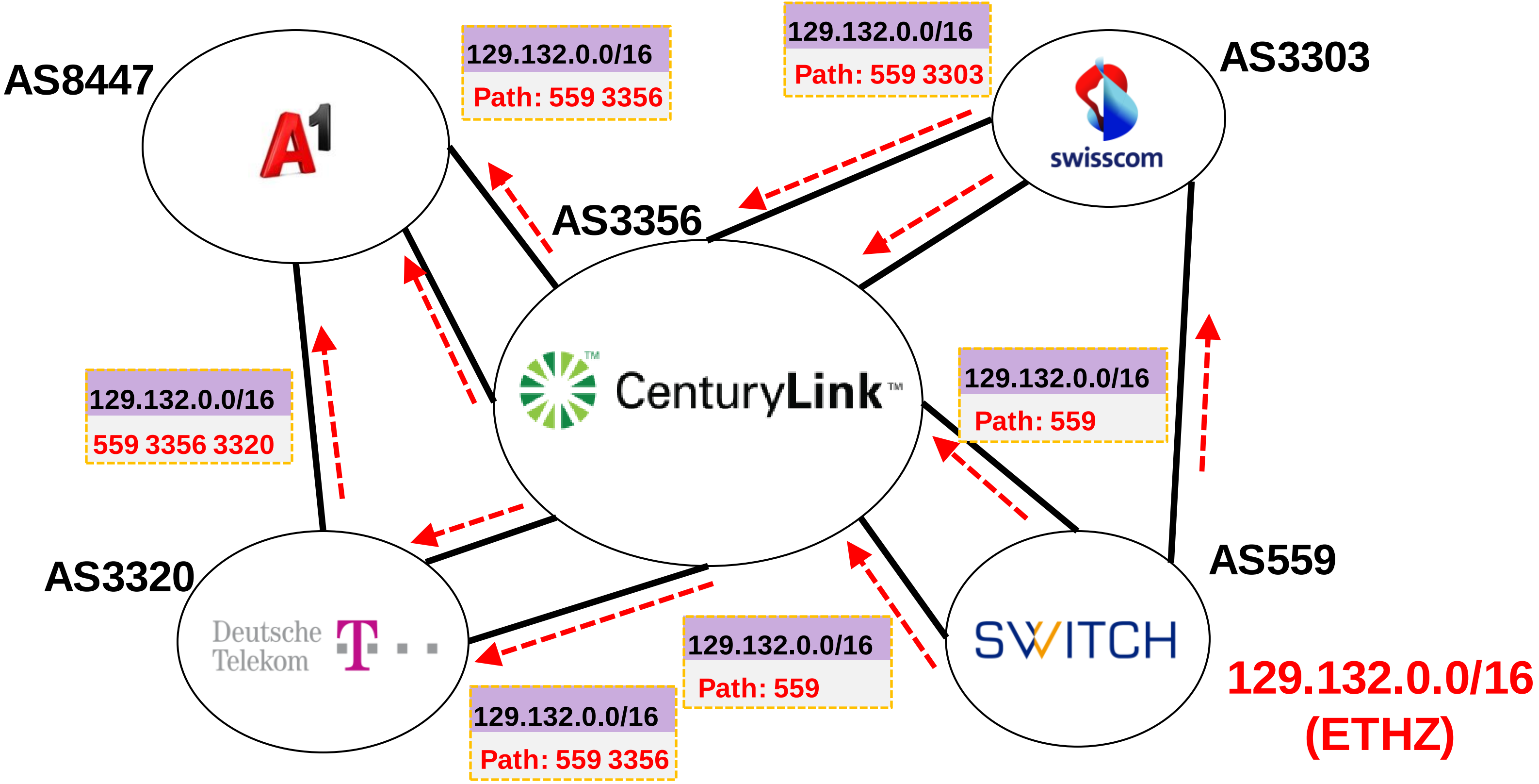
The Internet is a network of networks (ASes)



BGP “glues” these systems together



ASes exchange information about IP prefixes they can reach directly or indirectly



Important Protocols

Transmission Control Protocol (TCP)

- Deliver data to correct application
- Split data into packets (segmentation), reassemble at destination (reassembly)
- Make sure data is unchanged and arrives in order: acknowledgments (ACKs), checksums, sequence numbers
- Do not overload receiver (flow control)
- Do not overload the network (congestion control)

TCP Header

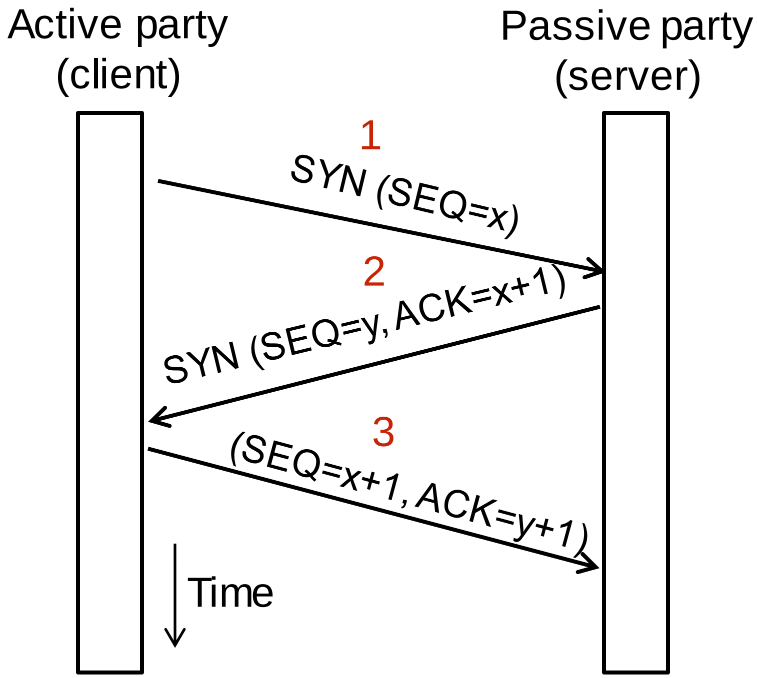
<table>
<tr><td colspan="10">Source port</td><td>Destination port</td></tr>
<tr><td colspan="11">Sequence number</td></tr>
<tr><td colspan="11">Acknowledgement number</td></tr>
<tr><td>TCP header length</td><td></td><td>CWR</td><td>ECE</td><td>URG</td><td>ACK</td><td>PSH</td><td>RST</td><td>SYN</td><td>FIN</td><td>Window size</td></tr>
<tr><td colspan="10">Checksum</td><td>Urgent pointer</td></tr>
<tr><td colspan="11">Options (0 or more 32-bit words)</td></tr>
</table>

Connection Establishment
(§6.5.5, §6.5.7, §6.2.2)

- Both sender and receiver must be ready before we start the transfer of data
  - Need to agree on a set of parameters
  - e.g., the Maximum Segment Size (MSS)
- This is signaling
  - It sets up state at the endpoints
  - Like “dialing” for a telephone call

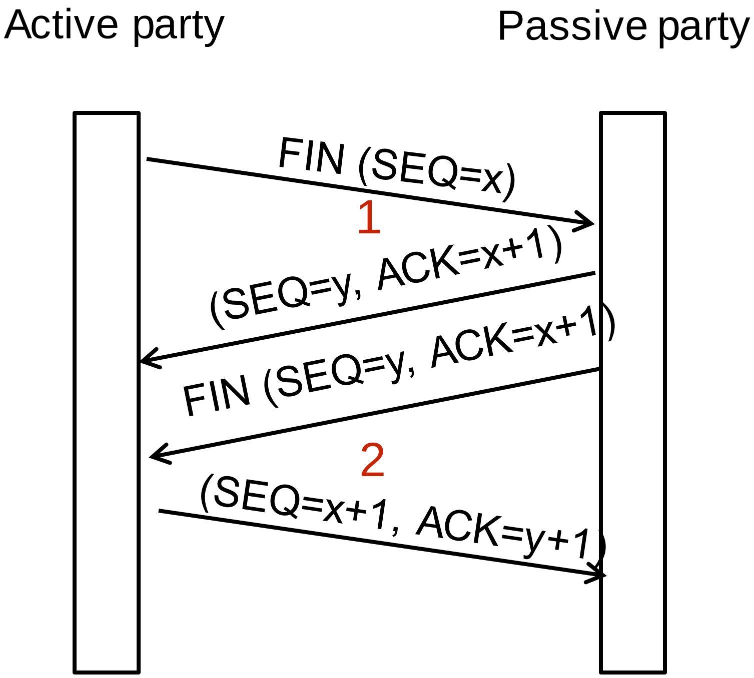
TCP Three-Way Handshake

- Three steps:
  - Client sends SYN(x)
  - Server replies with SYN(y)ACK(x+1)
  - Client replies with ACK(y+1)
  - SYNs are retransmitted if lost
- Sequence and ack numbers carried on further segments



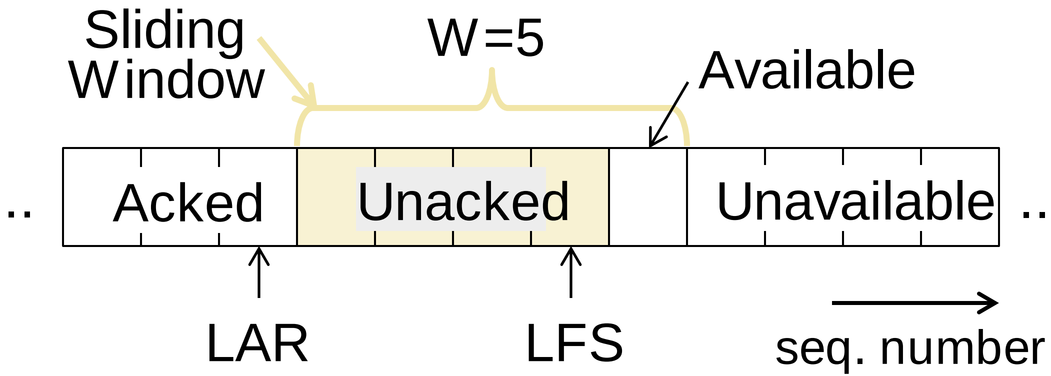
TCP Connection Release
(§6.5.6-6.5.7, §6.2.3)

- Two steps:
  - Active party sends FIN(x), passive party sends ACK
  - Passive party sends FIN(y), active party sends ACK
  - FINs are retransmitted if lost
- Each FIN/ACK closes one direction of data transfer



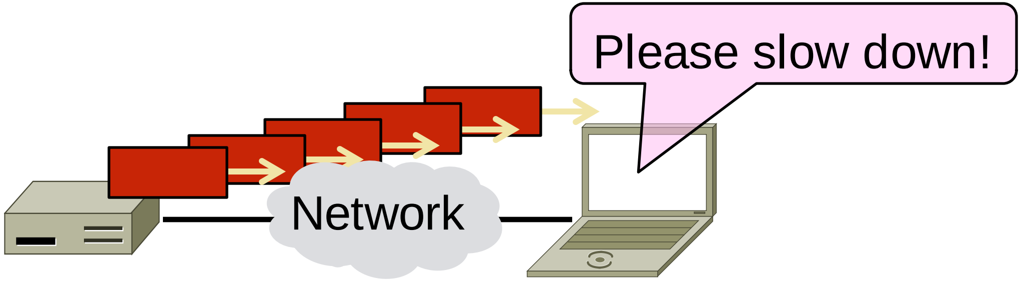
Sliding Window
(§3.4, §6.5.8)

- Sender buffers up to W segments until they are acknowledged
  - LFS=LAST FRAME SENT, LAR=LAST ACK RECEIVED
  - Sends while LFS – LAR ≤ W



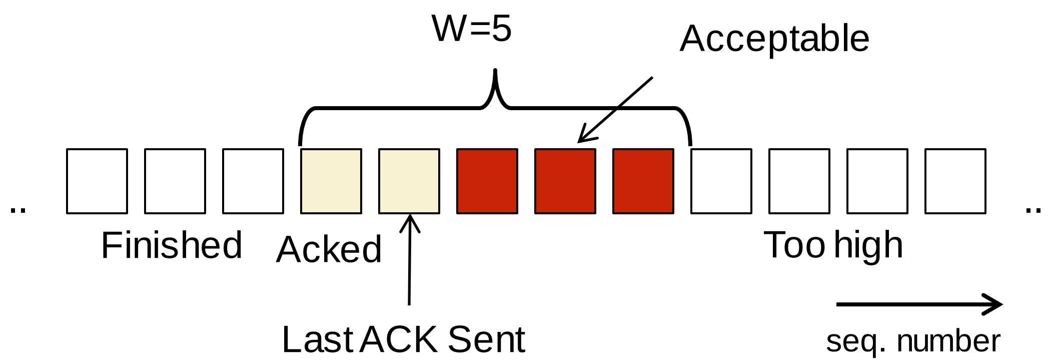
Flow Control (§6.5.8)

- Adding flow control to the sliding window algorithm
  - To slow the over-enthusiastic sender



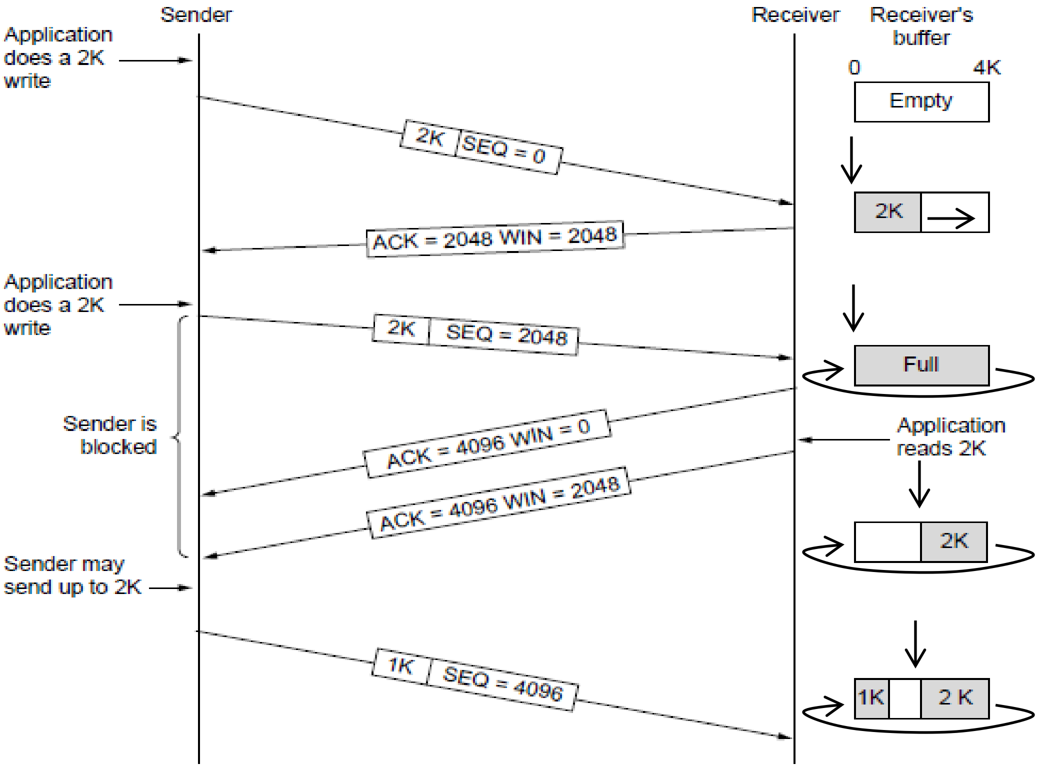
Flow Control

- Avoid loss at receiver by telling sender the available buffer space
  - Finished: delivered to application; Acked: not yet delivered
  - Tell sender available space in buffer: W - Acked



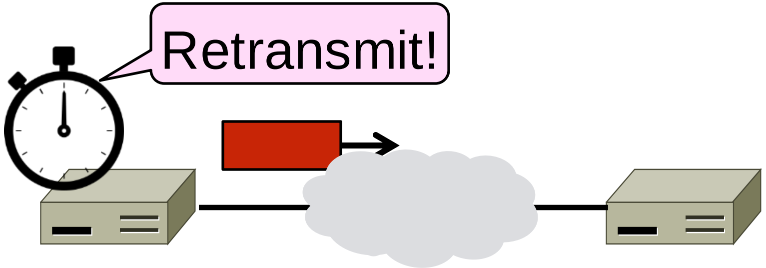
Flow Control

- TCP-style example
  - SEQ/ACK sliding window
  - Flow control with WIN
  - SEQ + length < ACK + WIN
  - 4KB buffer at receiver
  - Circular buffer of bytes



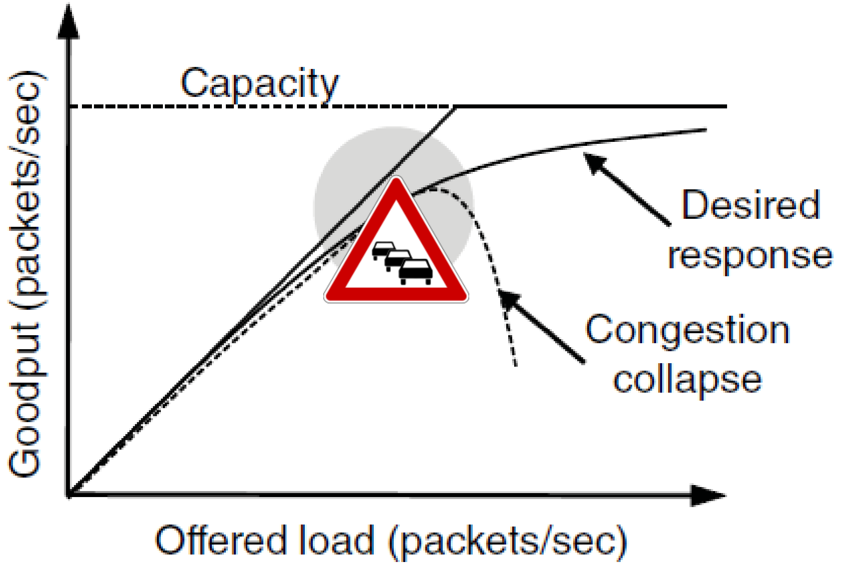
Retransmissions

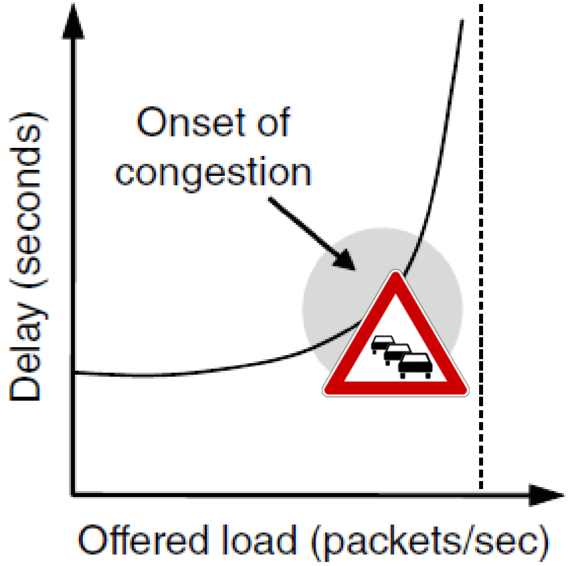
- With sliding window, the strategy for detecting loss is the timeout
  - Set timer when a segment is sent
  - Cancel timer when ack is received
  - If timer fires, retransmit data as lost



Effects of Congestion

- What happens to performance as we increase the load?





Bandwidth Allocation

- Important task for network is to allocate its capacity to senders
  - Good allocation is efficient and fair

- Efficient means that most capacity is used but there is no congestion
- Fair means that every sender gets a reasonable share of the network
  - Different possible definitions of fairness

Max-Min Fairness

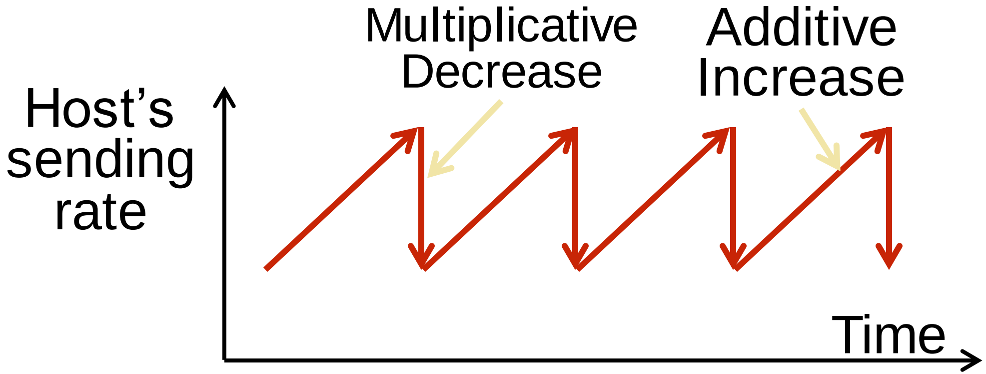
- Intuitively, flows bottlenecked on a link get an equal share of that link
- Max-min fair allocation:
  - Increasing the rate of one flow will decrease the rate of a smaller flow
  - This “maximizes the minimum” flow
  - Easy to calculate centrally but needs to be distributed

Congestion Control

- Keep second congestion window (cwnd)
- Window size dynamically adjusts to network utilization (congestion)
- Goal: achieve efficient and fair allocation

Additive Increase Multiplicative Decrease (AIMD)
(§6.3.2)

- Increase congestion window by a constant additive amount every round-trip time (RTT)
- Decrease congestion window by a multiplicative factor when congestion occurs (packet loss)
- Produces “sawtooth”pattern



AIMD Properties

- Converges to an allocation that is efficient and fair when hosts run it
  - Holds for general topologies
- Other increase/decrease control laws do not! (Try MIAD, MIMD, AIAD)
- Requires only binary feedback about congestion from the network

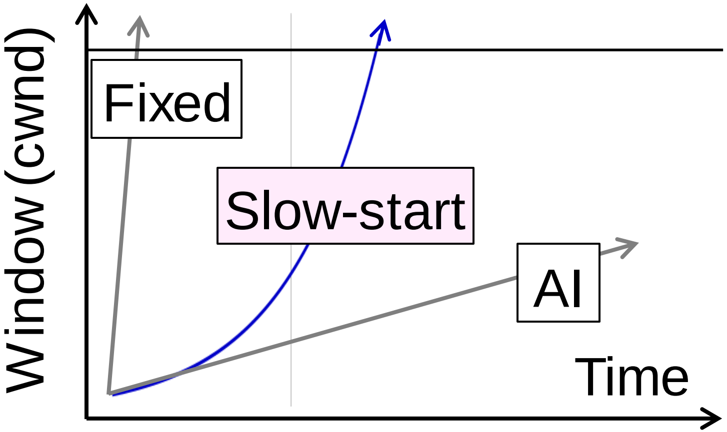
Feedback Signals

- Several possible signals, with different pros/cons
  - Classic TCP uses packet loss as a signal
  - Today many different congestion-control algorithms

Signal	Example Protocol	Pros / Cons
Packet loss	TCP NewReno TCP Cubic (Linux)	+Hard to get wrong -Hear about congestion late
Packet delay	Compound TCP (Windows) BBR (Google)	+Hear about congestion early -Need to infer congestion
Router indication	TCPs with Explicit Congestion Notification	+Hear about congestion early -Require router support

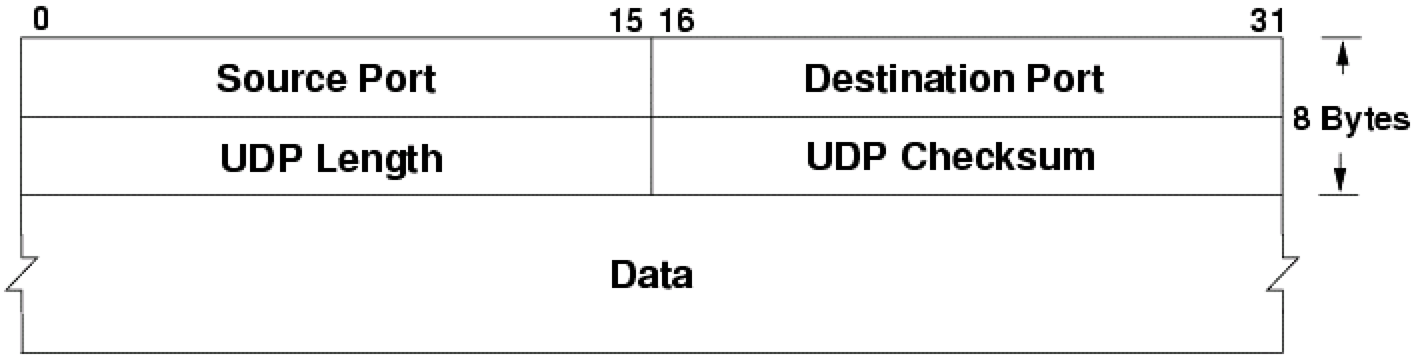
Slow-Start Solution

- Start by doubling cwnd every RTT
  - Exponential growth (1, 2, 4, 8, 16, …)
  - Start slow, quickly reach large values
- Switch to AIMD after packet loss

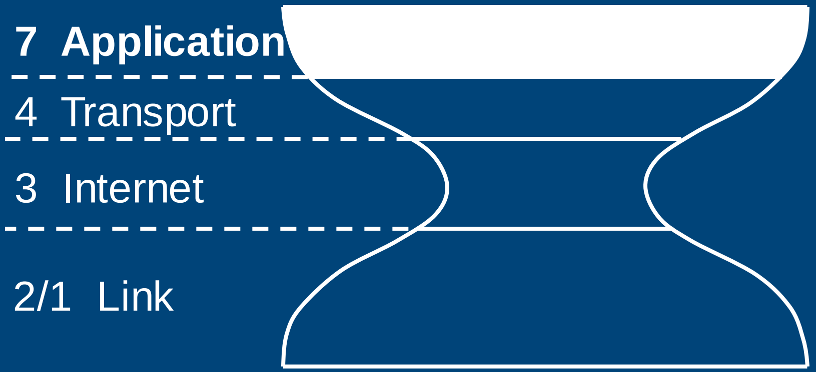


User Datagram Protocol (UDP)

- Only delivers packets to correct application
- Checksum for error detection
- No segmentation, no retransmissions, no flow control, no congestion control
- Advantage: no connection establishment, very little overhead



Important Protocols

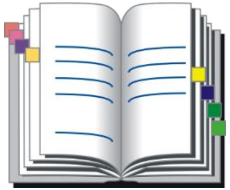


Names and Addresses

- Names: higher-level (user-understandable) resource identifiers
- Addresses: lower-level resource locators
  - Multiple levels, e.g., full name → email → IP address → Ethernet address
- Resolution (or lookup): mapping a name to an address

Name, e.g.,
“Markus Legner,”
or “ethz.ch”

Lookup



Directory

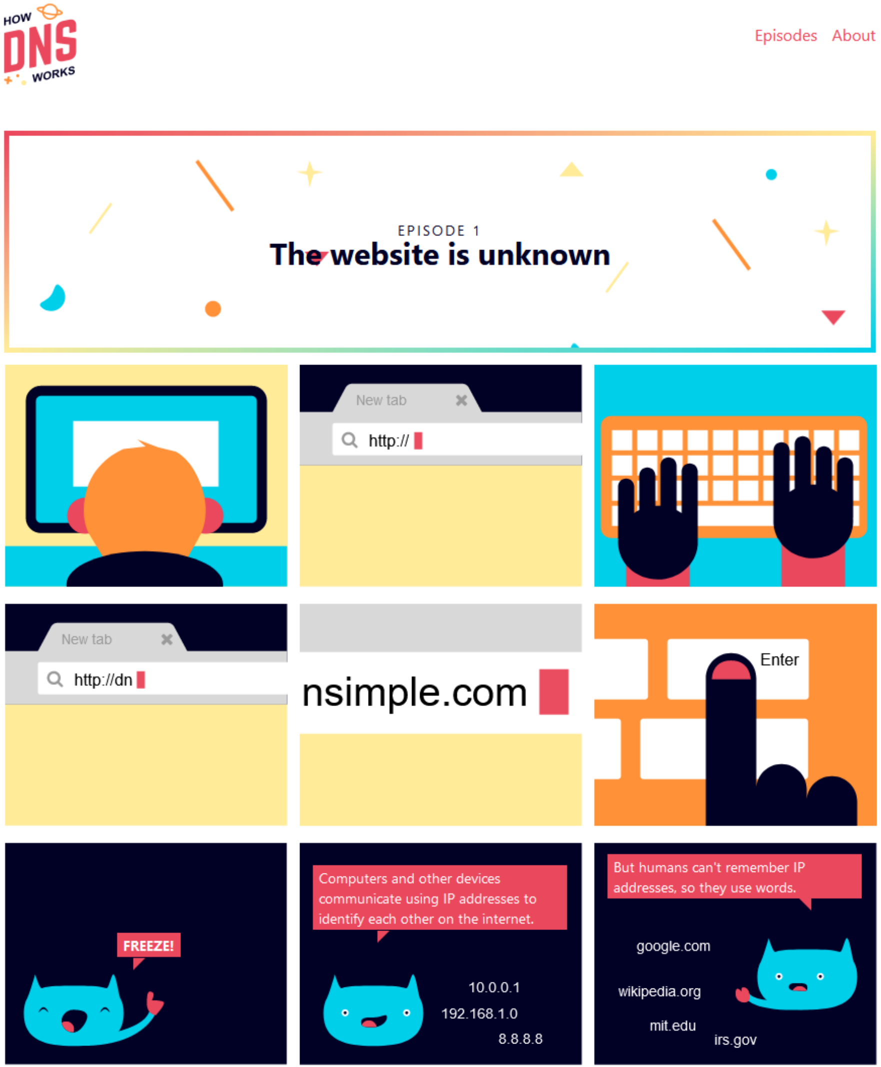
Address, e.g.,
“ETH Zurich”
or IPv4 “129.132.19.216”

Domain Name System (DNS)

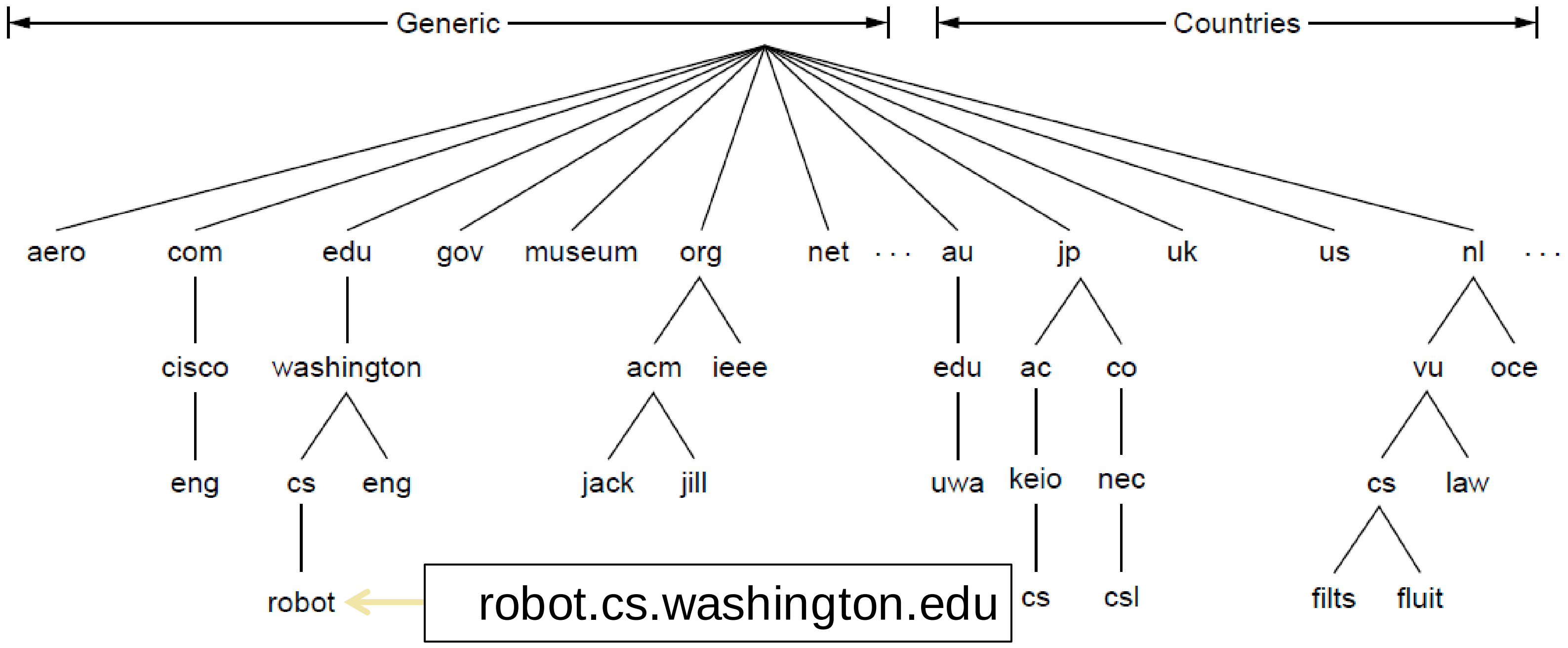
(§7.1.1-7.1.3)

- Translates between human-readable host names and IP addresses
- Hierarchical name space
- Distributed operation

https://howdns.works

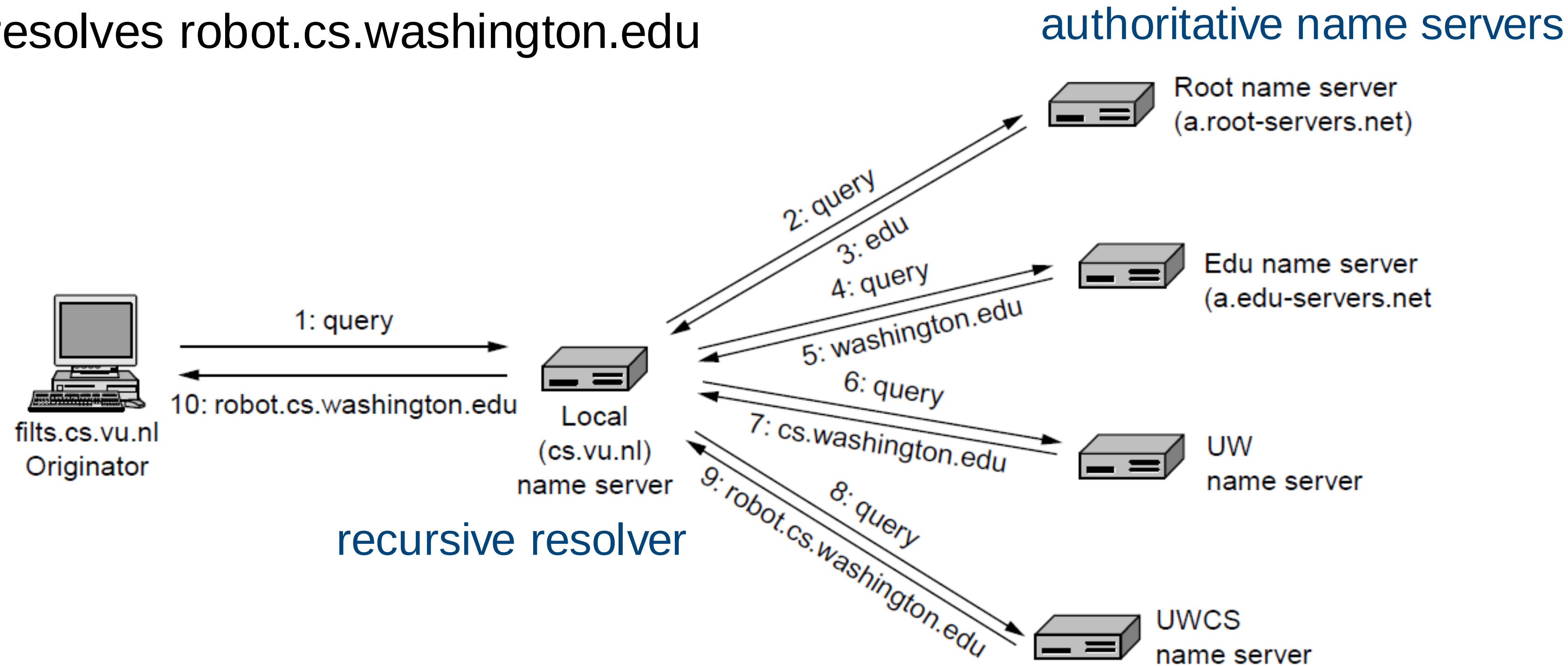


DNS Namespace



DNS Resolution

Host resolves robot.cs.washington.edu



Recursive Resolvers Make Extensive Use of Caching

- Caching is crucial to make DNS scalable
- At every step of the name resolution, recursive resolvers check cache
  - If result is cached (and not expired), use it directly
  - If result is not cached, contact authoritative name server and add result to cache

Summary

Summary

- Computer networks are based on layering
  - Provide modularity and reusability
- Security was of little concern in the early days of the Internet; only reliability built in
  - Security mechanisms added later → we will see them in this course
- Important protocols of today's Internet: IP, BGP, TCP, DNS, HTTP
  - Task: think about how each of the protocols can be attacked
  - We will discuss additional security-related protocols in this course: IPSec, TLS, BGPsec, DNSSEC, …