Communication Networks

Spring 2023

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
June 1 2023

Materials inspired from Scott Shenker, Jennifer Rexford, and Ankit Singla
Last week on

Communication Networks
google.ch ➔ 172.217.16.131
(the end)

http://www.google.ch
(the beginning)
google.ch 172.217.16.131
(the end)
<table>
<thead>
<tr>
<th>Records</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>hostname</td>
<td>IP address</td>
</tr>
<tr>
<td>NS</td>
<td>domain</td>
<td>DNS server name</td>
</tr>
<tr>
<td>MX</td>
<td>domain</td>
<td>Mail server name</td>
</tr>
<tr>
<td>CNAME</td>
<td>alias</td>
<td>canonical name</td>
</tr>
<tr>
<td>PTR</td>
<td>IP address</td>
<td>corresponding hostname</td>
</tr>
</tbody>
</table>
DNS resolution can either be recursive or iterative
Where is .edu?

Where is www.nyu.edu?

Where is nyu.edu?

DNS server

root

.edu servers

nyu.edu servers

DNS client
(me.ee.ethz.ch)

local DNS server

DNS client
(me.ee.ethz.ch)
DNS

Web

http://www.google.ch
(the beginning)
The WWW is made of three key components

- **Infrastructure**
  - Clients/Browser
  - Servers
  - Proxies

- **Content**
  - Objects
    - files, pictures, videos, …
    - organized in
  - Web sites
    - a collection of objects

- **Implementation**
  - URL: name content
  - HTTP: transport content
A Uniform Resource Locator (URL) refers to an Internet resource

protocol://hostname[:port]/directory_path/resource
HTTP is a rather simple synchronous request/reply protocol

HTTP is layered over a bidirectional byte stream typically TCP, but QUIC is ramping up

HTTP is text-based (ASCII) human readable, easy to reason about

HTTP is stateless it maintains *no info* about past client requests
Today on Communication Networks
Web

http://www.google.ch
(the end)

Course recap

The life of Internet packets
(a streamed movie)
http://www.google.ch

(the end)
Problem: How to guarantee the highest video quality?
Without seeing this ...
A naive approach: one-size-fits-all

[bitmovin.com]
The three steps behind most contemporary solutions

- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
  - Estimate connection’s available bandwidth
  - Pick a bitrate $\leq$ available bandwidth
Encoding

Replication

Adaptation
1920 x 1080 px

Fast Internet

Screen size: 1920 x 1080 px
With *fast* internet.

Video plays at *high quality*
1920 x 1080 px with *no buffering*

1280x 720 px

Slow Internet

Screen size: 1920 x 1080 px
With *slower* internet.

Video plays at *medium quality*
1280x 720 px with *no buffering*
Normal connection:
The Player downloads the best quality video

Poor connection:
The Player changes to downloading a smaller, faster video file

Normal connection:
The Player returns to the maximum quality video file
Simple solution for encoding: use a “bitrate ladders”

<table>
<thead>
<tr>
<th>Bitrate (kbps)</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>235</td>
<td>320x240</td>
</tr>
<tr>
<td>375</td>
<td>384x288</td>
</tr>
<tr>
<td>560</td>
<td>512x384</td>
</tr>
<tr>
<td>750</td>
<td>512x384</td>
</tr>
<tr>
<td>1050</td>
<td>640x480</td>
</tr>
<tr>
<td>1750</td>
<td>720x480</td>
</tr>
<tr>
<td>2350</td>
<td>1280x720</td>
</tr>
<tr>
<td>3000</td>
<td>1280x720</td>
</tr>
<tr>
<td>4300</td>
<td>1920x1080</td>
</tr>
<tr>
<td>5800</td>
<td>1920x1080</td>
</tr>
</tbody>
</table>

[netflix.com]
Your player download “chunks” of video at different bitrates
Depending on your network connectivity, your player fetches chunks of different qualities.
Your player gets metadata about chunks via "Manifest"

```xml
<?xml version="1.0" encoding="UTF-8"?>
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns="urn:mpeg:DASH:schema:MPD:2011"
     xsi:schemaLocation="urn:mpeg:DASH:schema:MPD:2011"
     profiles="urn:mpeg:dash:profile:isoff-main:2011"
     type="static"
     mediaPresentationDuration="PT0H9M56.46S"
     minBufferTime="PT15.0S">
  <BaseUrl>http://witestlab.poly.edu/~ffund/video/2s_480p_only/</BaseUrl>
  <Period start="PT0S">
    <AdaptationSet bitstreamSwitching="true">
      <Representation id="0" codecs="avc1" mimeType="video/mp4"
                      width="480" height="360" startWithSAP="1" bandwidth="101492">
        <SegmentBase>
          <Initialization sourceURL="bunny_2s_100kbit/bunny_100kbit.mp4"/>
        </SegmentBase>
        <SegmentList duration="2">
          <SegmentURL media="bunny_2s_100kbit/bunny_2s1.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s2.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s3.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s4.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s5.m4s"/>
          <SegmentURL media="bunny_2s_100kbit/bunny_2s6.m4s"/>
        </SegmentList>
      </Representation>
    </AdaptationSet>
  </Period>
</MPD>
```
Open Connect: Starting from a Greenfield (a mostly Layer 0 talk)

Dave Temkin
06/01/2015
“Storage” appliance

Designed for bulk storage of regional content catalogs

(several servers required, number varies by catalog)
Storage Appliances

Storage appliances are 2U servers that are focused on reliable dense storage and cost effective throughput. This appliance is used to hold the Netflix catalog in many IX locations around the world and embedded at our larger ISP partner locations.

Storage appliance focus areas

- Large storage capacity
- 2U for rack efficiency (no deeper than 29 inches)
- Enough low cost NAND to reach 10GB/s of throughput (<0.3 DWPD)
- Network flexibility to connect at 6x10GE LAG or 1x100GE
- 2 and 4 post racking
- AC or DC power
- Single processor

Storage appliance high-level specifications

<table>
<thead>
<tr>
<th>Option</th>
<th>Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chassis</td>
<td>Sanmina</td>
</tr>
<tr>
<td>Motherboard</td>
<td>Supermicro</td>
</tr>
<tr>
<td>Processor</td>
<td>Intel</td>
</tr>
<tr>
<td>Memory</td>
<td>Micron</td>
</tr>
<tr>
<td>Hard Drive</td>
<td>HGST</td>
</tr>
<tr>
<td>Solid State Drive</td>
<td>Micron, Toshiba</td>
</tr>
<tr>
<td>Network Controller</td>
<td>Chelsio</td>
</tr>
<tr>
<td>Power draw operational (peak)</td>
<td>~500W</td>
</tr>
<tr>
<td>Power Supply Unit</td>
<td>Redundant Hot Swap AC/DC</td>
</tr>
<tr>
<td>Operational throughput</td>
<td>~36Gbps</td>
</tr>
<tr>
<td>Raw storage capacity</td>
<td>~288 TB</td>
</tr>
</tbody>
</table>
You're watching
The Big Bang Theory
Season 12: Ep. 1

The Conjugal Configuration
Sheldon and Amy’s honeymoon hits a scheduling snafu. Leonard upsets Penny with an unflattering comparison and Raj sparks a Twitter war with a celebrity.
You're watching

The Big Bang Theory
Season 12: Ep. 1

The Conjugal Configuration

Sheldon and Amy's honeymoon hits a scheduling snag. Leonard upsets Penny with an unflattering comparison and Raj sparks a Twitter war with a celebrity.

Paused

https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net
Complete Playback Workflow @Netflix

1. Report health status, routability, and available assets
2. “Play” request
3. Determines required assets
4. Picks OCA, generates URL
5. Device gets OCA URL from Playback Apps, video streams from OCA to Device

Client Devices

OCAs

Playback Apps
Steering Service
Cache Control Service

Netflix in AWS

[more-ip-event.net]
How many OCA appliances in Swisscom?
I found at least 35 of them

Assuming all of them are fully loaded → **10 080 TB of storage!!** (288 TB x 35)

>2 million 1080p movies, assuming 100 min encoded at 5 Mbps
Besides OCAs within ISPs, Netflix also hosts caches at various IXPs and datacenters

ipv4-c001-zrh001-ix.1.oca.nflxvideo.net  45.57.18.130  ipv4-c013-zrh001-ix.1.oca.nflxvideo.net  45.57.19.135
ipv4-c002-zrh001-ix.1.oca.nflxvideo.net  45.57.18.131  ipv4-c014-zrh001-ix.1.oca.nflxvideo.net  45.57.19.136
ipv4-c003-zrh001-ix.1.oca.nflxvideo.net  45.57.18.132  ipv4-c015-zrh001-ix.1.oca.nflxvideo.net  45.57.18.137
ipv4-c004-zrh001-ix.1.oca.nflxvideo.net  45.57.19.130  ipv4-c016-zrh001-ix.1.oca.nflxvideo.net  45.57.18.138
ipv4-c005-zrh001-ix.1.oca.nflxvideo.net  45.57.19.131  ipv4-c017-zrh001-ix.1.oca.nflxvideo.net  45.57.19.137
ipv4-c006-zrh001-ix.1.oca.nflxvideo.net  45.57.19.132  ipv4-c018-zrh001-ix.1.oca.nflxvideo.net  45.57.19.138
ipv4-c007-zrh001-ix.1.oca.nflxvideo.net  45.57.18.133  ipv4-c019-zrh001-ix.1.oca.nflxvideo.net  45.57.18.139
ipv4-c008-zrh001-ix.1.oca.nflxvideo.net  45.57.18.134  ipv4-c020-zrh001-ix.1.oca.nflxvideo.net  45.57.18.140
ipv4-c009-zrh001-ix.1.oca.nflxvideo.net  45.57.18.135  ipv4-c021-zrh001-ix.1.oca.nflxvideo.net  45.57.18.141
ipv4-c010-zrh001-ix.1.oca.nflxvideo.net  45.57.18.136  ipv4-c022-zrh001-ix.1.oca.nflxvideo.net  45.57.19.139
ipv4-c011-zrh001-ix.1.oca.nflxvideo.net  45.57.19.133  ipv4-c023-zrh001-ix.1.oca.nflxvideo.net  45.57.19.140
ipv4-c012-zrh001-ix.1.oca.nflxvideo.net  45.57.19.134  ipv4-c024-zrh001-ix.1.oca.nflxvideo.net  45.57.19.141

At least 24 instances in Zurich Equinix, see https://openconnect.netflix.com/en/peering/#locations
If you are interested in finding out more: check out https://openconnect.netflix.com

Deployment guide: https://openconnect.netflix.com/deploymentguide.pdf
Capacity (Mbps)

Time

Network

Downloading

1s chunks at different bit-rates

Capacity < current rate ⇒ decrease rate

Playing out
Common solution approach

- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
  - Estimate connection’s available bandwidth
  - Pick a bitrate ≤ available bandwidth
Buffer-based capacity adaptation

Decide based on the buffer alone?
Buffer-based capacity adaptation

Nearly full buffer $\Rightarrow$ large rate
Buffer-based capacity adaptation

**Nearly empty buffer ⇒ small rate**
Buffer-based capacity adaptation

[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]
Web

Course recap

The life of Internet packets
(a streamed movie)
Communication Networks

So what?!
Knowledge

Understand **how** the Internet works and **why**

from your network plug... 

...to the largest data-centers out there
Let's do a quick recap of the lecture by dissecting "The life of a few packets" together

Our goal: watch a video on my.video.com
A destination outside of our local network

We consider a new host with clean state
Network-wise nothing is configured/known

Which packets do we need to achieve that?
Our **host** belongs to AS 1, my.video.com belongs to **AS 5**
Our **host** belongs to AS 1, my.video.com belongs to **AS 5**
**Problem:** Who and where am I?

DHCP

The Dynamic Host Configuration Protocol provides:

- an IP address
- the corresponding IP prefix
- the IP of the default gateway
- DNS server to use
- (many other options)

Manual

Alternatively, we can manually configure the host.

You did that extensively during the routing project.
DHCP works within a broadcast domain
(i.e. a local L2 network)

src MAC: host H’s MAC
dst MAC: ff:ff:ff:ff:ff:ff

---

DHCP discovery:
I want an IP
src MAC: host H’s MAC
dst MAC: ff:ff:ff:ff:ff:ff

DHCP discovery:
I want an IP

Broadcasted along the layer 2 Spanning Tree computed by the switches

DHCP server
The DHCP server unicasts its answer back to the sender

<table>
<thead>
<tr>
<th>src MAC: MAC of DHCP</th>
<th>dst MAC: host H’s MAC</th>
</tr>
</thead>
</table>

**DHCP offer:**
- Use 192.168.1.20/24
- Default gw: 192.168.1.1
- DNS server: 192.168.1.2
The DHCP server unicasts its answer back to the sender

src MAC: MAC of DHCP
dst MAC: host H’s MAC

DHCP offer:
Use 192.168.1.20/24
Default gw: 192.168.1.1
DNS server: 192.168.1.2

The switches have learned over which physical ports they can reach the MAC of H

These slides show a simplified version of DHCP, see exercise 3 for more details
**Problem:** Who is my.video.com?

<table>
<thead>
<tr>
<th>DNS</th>
<th>The Domain Name System translates names to IPs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The opposite is also possible</td>
</tr>
<tr>
<td>Resource Records</td>
<td>A DNS server stores records for different resources</td>
</tr>
<tr>
<td></td>
<td>For example domains, mail servers, aliases…</td>
</tr>
<tr>
<td>Manual</td>
<td>Alternatively, we can directly provide the IP</td>
</tr>
<tr>
<td></td>
<td>But normally we do not know the IPs of external domains</td>
</tr>
</tbody>
</table>
Here, we'll consider that the DNS server is located in the local L2 network
Here, we'll consider that the DNS server is located in the **local L2 network**

We can also use external DNS servers

e.g. Google's
src MAC: host H’s MAC
dst MAC: ???

src IP: 192.168.1.20
dst IP: 192.168.1.2

**DHCP query:**
What is the IP of my.video.com?
src MAC: host H’s MAC
dst MAC: ???

src IP: 192.168.1.20
dst IP: 192.168.1.2

DHCP query:
What is the IP of my.video.com?

What is the MAC address of the DNS server?
Known via DHCP

DNS server 192.168.1.2
**Problem:** How to reach destinations in the same layer 2 network?

| ARP          | The Address Resolution Protocol discovers MACs of IPs
|              | Only works inside one layer 2 network

| ARP tables   | Hosts cache ARP replies in their local ARP table
|              | Entries will eventually expire

| Manual       | Alternatively, we can populate the ARP table statically
Our host performs an ARP request for the IP of the DNS server

src MAC: host H’s MAC
dst MAC: ff:ff:ff:ff:ff:ff

ARP request:
Who has 192.168.1.2
tell 192.168.1.20

DNS server 192.168.1.2
Our host performs an ARP request for the IP of the DNS server

src MAC: host H’s MAC
dst MAC: ff:ff:ff:ff:ff:ff

ARP request:
Who has 192.168.1.2
tell 192.168.1.20

Broadcasted along the layer 2
Spanning Tree computed by the switches

DNS server
192.168.1.2
The DNS server unicasts its MAC address

src MAC: MAC of DNS
dst MAC: host H’s MAC

ARP reply:
192.168.1.20 is at
<MAC of DNS server>

The switches have learned over which physical ports they can reach the MAC of H.
We can finally perform our DNS query
(not shown in detail)

The DNS server might contact other name servers depending on what is in its cache

We have seen two resolution strategies:

- **recursive**, by offloading it to other servers
- **iterative**, by iteratively querying the "next servers"

In our example, my.video.com has the IP: **5.6.7.8**
**Problem:** How to reach destinations outside of our local network?

<table>
<thead>
<tr>
<th>Default gateway</th>
<th>We send the packets to our default gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Known via DHCP (or statically configured)</td>
</tr>
<tr>
<td>Routers</td>
<td>The default gateway is normally a layer-3 router</td>
</tr>
<tr>
<td></td>
<td>For example your &quot;Internet box&quot; at home</td>
</tr>
<tr>
<td>How to reach the gateway?</td>
<td>Already solved, we use <strong>ARP</strong> to find the MAC address</td>
</tr>
<tr>
<td></td>
<td>Then forwarded over the layer 2 network</td>
</tr>
</tbody>
</table>
Our host can finally send a first packet towards my.video.com

src MAC: host H’s MAC
dst MAC: <MAC of gw>
src IP: 192.168.1.20
src port: 5555
dst IP: 5.6.7.8
dst port: 443
TCP SYN
Our host can finally send a first packet towards my.video.com

- src MAC: host H’s MAC
- dst MAC: <MAC of gw>
- src IP: 192.168.1.20
- src port: 5555
- dst IP: 5.6.7.8
- dst port: 443
- TCP SYN

From ARP request for 192.168.1.1
Randomly selected source port
HTTP-based streaming (the default nowadays)
TCP-based data transmission
**Problem:** How to reach external destinations using a private IP as source address?

**NAT**
Network Address Translation solves this problem
A single public IP is shared between hosts

**Benefits**
NAT has multiple benefits:
- "solution" to the IPv4 address depletion
- better privacy and anonymization
- hosts not reachable from the outside
Here, we'll consider that the default gateway performs NAT.

**Diagram:**
- Default gateway (1.2.3.4/24)
- Network topology showing host connections.

**TCP SYN Packets:**
1. **Source:** Host H's MAC
   **Destination:** <MAC of gw>
   **Source IP:** 192.168.1.20
   **Source Port:** 5555
   **Destination IP:** 5.6.7.8
   **Destination Port:** 443
   **Protocol:** TCP SYN

2. **Source:** 1.2.3.4
   **Destination:** <MAC of gw>
   **Source IP:** 1.2.3.4
   **Source Port:** 7744
   **Destination IP:** 5.6.7.8
   **Destination Port:** 443
   **Protocol:** TCP SYN

**Mapping:**
- 192.168.1.20:5555 → 1.2.3.4:7744
- Mapping stored in NAT table.

---

**Table (Left):**
- **Source MAC:** Host H’s MAC
- **Destination MAC:** <MAC of gw>
- **Source IP:** 192.168.1.20
- **Source Port:** 5555
- **Destination IP:** 5.6.7.8
- **Destination Port:** 443
- **Protocol:** TCP SYN

---

**Table (Right):**
- **Source MAC:** <MAC of gw>
- **Destination MAC:** ???
- **Source IP:** 1.2.3.4
- **Source Port:** 7744
- **Destination IP:** 5.6.7.8
- **Destination Port:** 443
- **Protocol:** TCP SYN
**Problem:** How to reach external destinations outside of our AS?

<table>
<thead>
<tr>
<th>BGP</th>
<th>Inter-domain routing using the Border Gateway Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A path-vector protocol</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Forwarding</th>
<th>Based on the best-matching prefix (longest match)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One next hop for each prefix</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>iBGP &amp; eBGP</th>
<th>Two versions of BGP to distribute routes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>eBGP distributes routes between ASes</td>
</tr>
</tbody>
</table>
Our packet is forwarded over multiple hops based on best-matching BGP routes

Forwarding table:
- Prefix: 5.6.7.0/24
- Next-hop: R2

Source MAC: <MAC of gw>
Destination MAC: <MAC of R2>
Source IP: 1.2.3.4
Source port: 7744
Destination IP: 5.6.7.8
Destination port: 443
TCP SYN
Our packet is forwarded over multiple hops based on best-matching BGP routes.

forwarding table:

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6.7.0/24</td>
<td>R3</td>
</tr>
</tbody>
</table>

TCP SYN

src MAC: <MAC of gw>  
dst MAC: <MAC of R2>  
src IP: 1.2.3.4  
src port: 7744  
dst IP: 5.6.7.8  
dst port: 443  
TCP SYN

src MAC: <MAC of R2>  
dst MAC: <MAC of R3>  
src IP: 1.2.3.4  
src port: 7744  
dst IP: 5.6.7.8  
dst port: 443  
TCP SYN
Finally, we reach another AS

forwarding table:
prefix       next-hop
5.6.7.0/24    R4

```
src MAC: <MAC of gw>
dst MAC: <MAC of R2>
src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 443
TCP SYN
```

```
src MAC: <MAC of R2>
dst MAC: <MAC of R3>
src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 443
TCP SYN
```

```
src MAC: <MAC of R3>
dst MAC: ???
src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 443
TCP SYN
```
Problem: How to reach next hops which are not directly connected?

IGP
Forwarding information from Interior Gateway Protocols
Used for intra-domain routing

Two types
We saw two different types of protocols:
- link-state protocols (e.g., OSPF)
- distance-vector protocols (e.g., RIP)
Using the shortest IGP path, our packet reaches R4

forwarding table:

<table>
<thead>
<tr>
<th>prefix</th>
<th>next-hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.6.7.0/24</td>
<td>R4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>src MAC:</th>
<th>dst MAC:</th>
<th>src IP:</th>
<th>src port:</th>
<th>dst IP:</th>
<th>dst port:</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;MAC of R3&gt;</td>
<td>&lt;MAC of R5&gt;</td>
<td>1.2.3.4</td>
<td>7744</td>
<td>5.6.7.8</td>
<td>443</td>
</tr>
<tr>
<td>src MAC:</td>
<td>dst MAC:</td>
<td>src IP:</td>
<td>src port:</td>
<td>dst IP:</td>
<td>dst port:</td>
</tr>
<tr>
<td>&lt;MAC of R5&gt;</td>
<td>&lt;MAC of R6&gt;</td>
<td>1.2.3.4</td>
<td>7744</td>
<td>5.6.7.8</td>
<td>443</td>
</tr>
<tr>
<td>src MAC:</td>
<td>dst MAC:</td>
<td>src IP:</td>
<td>src port:</td>
<td>dst IP:</td>
<td>dst port:</td>
</tr>
<tr>
<td>&lt;MAC of R6&gt;</td>
<td>&lt;MAC of R4&gt;</td>
<td>1.2.3.4</td>
<td>7744</td>
<td>5.6.7.8</td>
<td>443</td>
</tr>
<tr>
<td></td>
<td>TCP SYN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP SYN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>TCP SYN</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Skipping a few similar steps, our packet finally reaches the **my.video.com** server.

```
src MAC: <MAC of R12>
dst MAC: <MAC of S>

src IP: 1.2.3.4
dst IP: 5.6.7.8
src port: 7744
dst port: 443

TCP SYN
```
**Problem:** How does the server know to which application the packet belongs?

<table>
<thead>
<tr>
<th>Port Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dst port</td>
<td>The virtual ports identify the target application</td>
</tr>
<tr>
<td></td>
<td>Completely different than physical ports on a device</td>
</tr>
<tr>
<td>Well-known</td>
<td>Ports in the range 0-1023</td>
</tr>
<tr>
<td></td>
<td>For example our video streaming port 443</td>
</tr>
<tr>
<td>Ephemeral</td>
<td>Most ports in the range 1024-65535</td>
</tr>
<tr>
<td></td>
<td>For example our source port(s): 7744 (5555 before NAT)</td>
</tr>
</tbody>
</table>
The server answers back with a SYN+ACK packet, which can take a different return path towards H

pkt created by S

```
src MAC: <MAC of S>
dst MAC: <MAC of R12>
dst IP: 5.6.7.8
src port: 443
dst IP: 1.2.3.4
dst port: 7744
TCP SYN+ACK
```
The server answers back with a SYN+ACK packet, which can take a different return path towards H.

<table>
<thead>
<tr>
<th>pkt created by S</th>
<th>pkt received by H (after NAT)</th>
</tr>
</thead>
<tbody>
<tr>
<td>src MAC: &lt;MAC of S&gt;</td>
<td>src MAC: &lt;MAC of gw&gt;</td>
</tr>
<tr>
<td>dst MAC: &lt;MAC of R12&gt;</td>
<td>dst MAC: &lt;MAC of H&gt;</td>
</tr>
<tr>
<td>src IP: 5.6.7.8</td>
<td>src IP: 5.6.7.8</td>
</tr>
<tr>
<td>src port: 443</td>
<td>src port: 443</td>
</tr>
<tr>
<td>dst IP: 1.2.3.4</td>
<td>dst IP: 192.168.1.20</td>
</tr>
<tr>
<td>dst port: 7744</td>
<td>dst port: 5555</td>
</tr>
<tr>
<td>TCP SYN+ACK</td>
<td>TCP SYN+ACK</td>
</tr>
</tbody>
</table>
Our host is now able to watch a video on my.video.com using the AS path [1 2 4 5]
But suddenly AS 4 withdraws the route due to internal link failures.

BGP update
WITHDRAWN route: 5.6.7.0/24
Next hop: AS 4
AS path: [4 5]
**Problem**: How to find new BGP routes after failures or BGP attribute changes?

**BGP decision algorithm**: The BGP decision algorithm finds a new best route based on all currently available routes towards a prefix.

**Convergence**: The new route is distributed over iBGP and eBGP. Unexpected forwarding behavior during the convergence.
Router R4 selects a new best route via AS 3 and distributes it via iBGP

BGP update
UPDATED route: 5.6.7.0/24
Next hop: R4
AS path: [3 5]
Finally, the new route is advertised via **eBGP** to AS 1 which now reaches 5.6.7.0/24 via [1 2 3 5]
What happens to our packets during the convergence?

Some packets are dropped immediately
E.g., on the failed links or in a buffer

Other packets might be part of a forwarding loop
They are eventually dropped once the TTL value reaches 0
**Problem:** How to handle lost or reordered packets?

**Reliable Transport:**
- TCP is the most-used Reliable Transport protocol
- UDP is an example for an unreliable protocol

**Features:**
- Reliable transport protocols provide:
  - correctness, data is delivered in order & unmodified
  - timeliness, minimized time until data is transferred
  - efficiency, optimal use of bandwidth
  - fairness, between concurrent flows

**Your GBN sender and receiver provide some of these features**

But for example, we do *not* provide fairness
Now you (better) understand this!

http://www.opte.org
Your final grade

- Exam: 70%
  - written, open book
- Projects: 30%
  - routing: 20%
  - transport: 10%
Your final grade

Exam

70%
written, open book

Projects

30%
20% routing
10% transport
The exam will be open book.
The questions will be similar to the exercises
Make sure you can do **all** the exercises, especially the ones in previous exams

https://comm-net.ethz.ch/#tab-exam
Don't forget the assignments, they matter

No programming question  no Python at the exam

*but*  we could ask you to describe a procedure in English

What would you change in your solution to achieve $X$?

No configuration question  no FRRouting at the exam

*but*  we could ask you to describe a configuration in English

How would you enforce policy $X$?
We'll organize another remote Q&A session closer to the exam (details to follow)
Communication Networks

What's next?
Master-level lecture, every Fall semester

Advanced Topics in Communication Networks

Topics
(examples)

Advanced Routing + labs
Network Programmability
Network Verification
Network Measurements
Network Security
Transport protocols
Video streaming
Network sustainability

https://adv-net.ethz.ch/
Consider doing one of your theses with our group!
bachelor, semester or master

https://nsg.ee.eethz.ch/theses/
That's all Folks!