Communication Networks

Prof. Laurent Vanbever

Communication Networks
Spring 2022

Laurent Vanbever
nsg.ee.ethz.ch

ETH Zürich (D-ITET)
May 30 2022

Materials inspired from Scott Shenker, Jennifer Rexford, and Ankit Singla

---

Last week on
Communication Networks

DNS

google.ch ⟷ 172.217.16.131
(the end)

Web

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

DNS

google.ch ⟷ 172.217.16.131
(the end)

Web

Records	Name	Value
A	hostname	IP address
NS	domain	DNS server name
MX	domain	Mail server name
CNAME	alias	canonical name
PTR	IP address	corresponding hostname

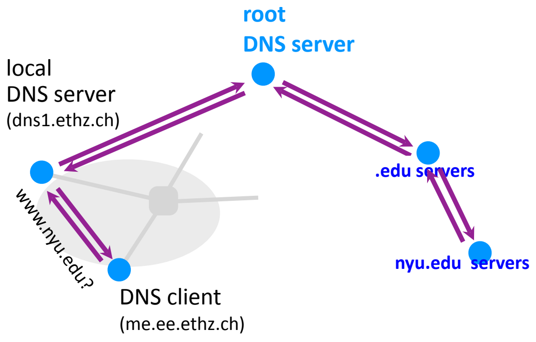
Using DNS relies on two components

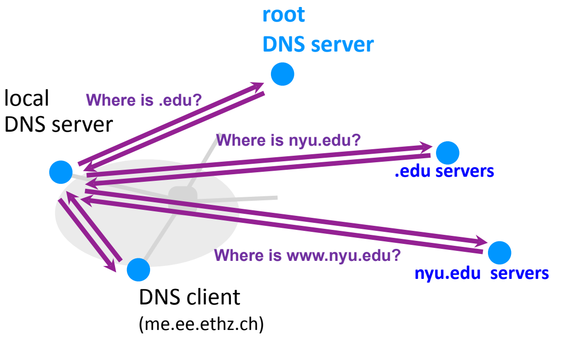
resolver software —*gethostbyname()*→ local DNS server

resolver software:
trigger resolution process
send request to local DNS server

local DNS server:
usually, near the endhosts
configured statically (resolv.conf)
or dynamically (DHCP)

DNS resolution can either be
recursive or iterative





DNS

Web

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

The WWW is made of
three key components

Infrastructure	Content	Implementation
Clients/Browser Servers Proxies	Objects files, pictures, videos, … *organized* in Web sites a collection of objects	URL: name content HTTP: transport content

A Uniform Resource Locator (URL)
refers to an Internet ressource

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)

Web

Course recap

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

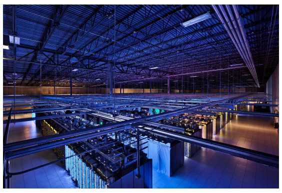
Web

Course recap

The life of Internet packets
(a streamed movie)

Knowledge
Understand how the Internet works and why

from your
network plug…



…to the largest data-centers out there

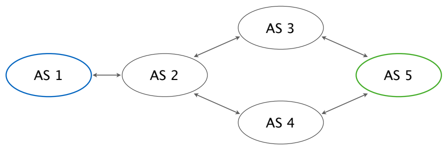
Let's do a quick recap of the lecture by dissecting
"The life of a few packets" together

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

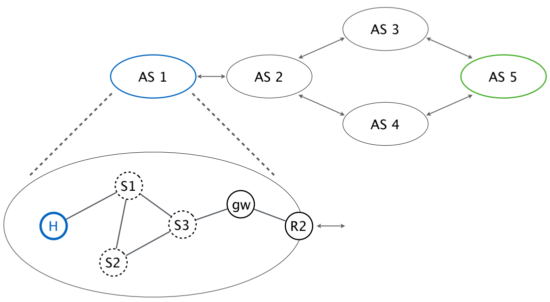
We consider a new host with clean state
I.e., network-wise nothing is configured/known

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

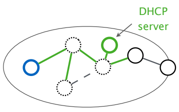
Alternatively, we can manually configure the host
You did that extensively during the routing project

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

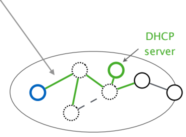


---

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

DHCP discovery:
I want an IP

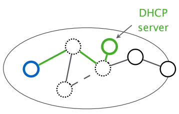
Broadcasted along the layer 2
Spanning Tree computed by the switches



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

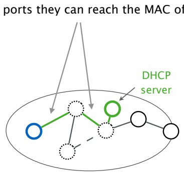


The DHCP server unicasts its answer back to the sender

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`?

DNS	The Domain Name System translates names to IPs The opposite is also possible
Resource Records	A DNS server stores records for different resources For example domains, mail servers, aliases…
Manual	Alternatively, we can directly provide the IP But normally we do not know the IPs of external domains

---

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

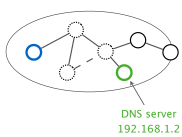
We can also use external DNS servers
e.g. Google's

---

src MAC: host H's MAC
dst MAC: ???

src IP: 192.168.1.20
dst IP: 192.168.1.2

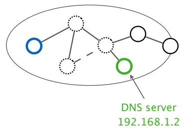
DHCP query:
What is the IP of
`my.video.com?`



src MAC: host H's MAC
dst MAC: ??? → What is the MAC address of the DNS server?

src IP: 192.168.1.20
dst IP: 192.168.1.2 → Known via DHCP

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



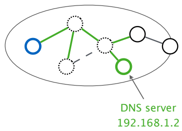
Problem: How to reach destinations in the same layer 2 network?

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

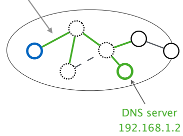
ARP request:
Who has 192.168.1.2
tell 192.168.1.20



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 → Broadcasted along the layer 2 Spanning Tree computed by the switches

ARP request:
Who has 192.168.1.2
tell 192.168.1.20

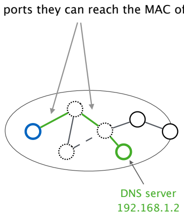


The DNS server unicasts its MAC address

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:

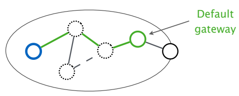
- *recursive*, by offloading it to other servers
- *iterative*, by iteratively querying the "next servers"

In our example, `my.video.com` has the IP: **5.6.7.8**

Problem: How to reach destinations outside of our local network?

Default gateway	We send the packets to our default gateway Known via DHCP (or statically configured)
Routers	The default gateway is normally a layer-3 router For example your "Internet box" at home
How to reach the gateway?	Already solved, we use **ARP** to find the MAC address Then forwarded over the layer 2 network

Our host can finally send a first packet towards `my.video.com`

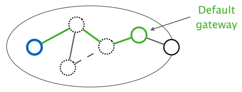


src MAC: host H's MAC
dst MAC: <MAC of gw>

src IP: 192.168.1.20
src port: 5555
dst IP: **5.6.7.8**
dst port: **554**

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> → From ARP request for 192.168.1.1

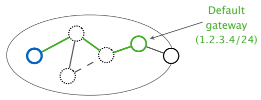
src IP: 192.168.1.20
src port: 5555 → Randomly selected source port
dst IP: **5.6.7.8**
dst port: **443** → HTTP-based streaming (the default nowadays)

TCP SYN → 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



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: 554

TCP SYN

192.168.1.20:5555 ↔ 1.2.3.4:7744

Mapping stored in NAT table

src MAC: <MAC of gw>
dst MAC: ???

src IP: **1.2.3.4**
src port: **7744**
dst IP: 5.6.7.8
dst port: 554

TCP SYN

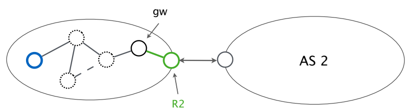
Problem: How to reach external destinations outside of our AS?

BGP	Inter-domain routing using the Border Gateway Protocol A path-vector protocol
Forwarding	Based on the best-matching prefix (longest match) One next hop for each prefix
iBGP & eBGP	Two versions of BGP to distribute routes eBGP distributes routes between ASes

Our packet is forwarded over multiple hops based on best-matching BGP routes

forwarding table:

prefix	next-hop
5.6.7.0/24	R2



src MAC: <MAC of gw>
dst MAC: <MAC of R2>

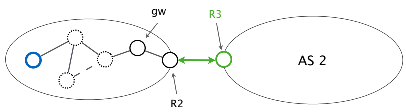
src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 554

TCP SYN

Our packet is forwarded over multiple hops based on best-matching BGP routes

forwarding table:

prefix	next-hop
5.6.7.0/24	R3



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: 554

TCP SYN

src MAC: <MAC of R2>
dst MAC: <MAC of R3>

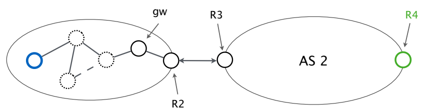
src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 554

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: 554

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: 554

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: 554

TCP SYN

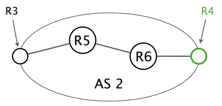
Problem: How to reach next hops which are not directly connected?

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:

prefix	next-hop
5.6.7.0/24	R4



src MAC: <MAC of R3>
dst MAC: <MAC of R5>

src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 554

TCP SYN

src MAC: <MAC of R5>
dst MAC: <MAC of R6>

src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 554

TCP SYN

src MAC: <MAC of R6>
dst MAC: <MAC of R4>

src IP: 1.2.3.4
src port: 7744
dst IP: 5.6.7.8
dst port: 554

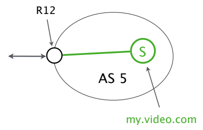
TCP SYN

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
src port: 7744
dst IP: 5.6.7.8
dst port: 554

TCP SYN



Problem: How does the server know to which application the packet belongs?

Dst port	The virtual ports identify the target application Completely different than physical ports on a device
Well-known	Ports in the range 0-1023 For example our video streaming port 554
Ephemeral	Most ports in the range 1024-65535 For example our source port(s): 7744 (5555 before NAT)

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>

src IP: **5.6.7.8**
src port: **554**
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

pkt created by S

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

src IP: 5.6.7.8
src port: 554
dst IP: 1.2.3.4
dst port: 7744

TCP SYN+ACK

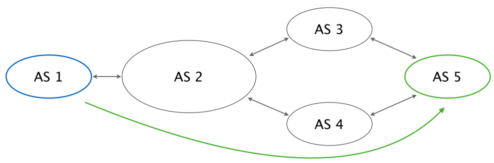
pkt received by H (after NAT)

src MAC: <MAC of gw>
dst MAC: <MAC of H>

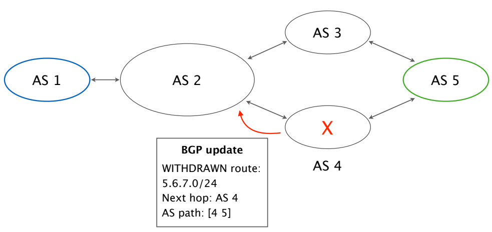
src IP: 5.6.7.8
src port: 554
dst IP: **192.168.1.20**
dst port: **5555**

TCP SYN+ACK

Our host is now able to watch a video on `my.video.com` using the AS path [1 2 4 5]



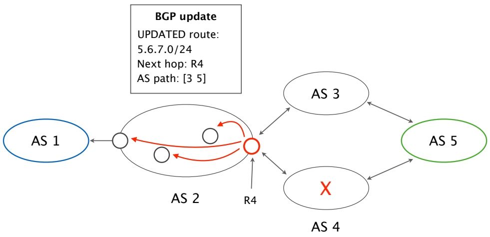
But suddenly AS 4 withdraws the route due to internal link failures



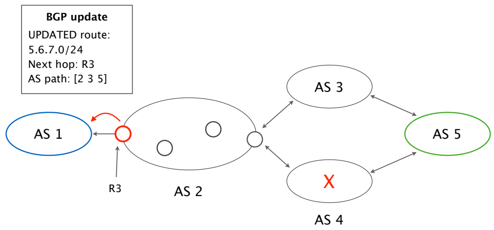
Problem: How to find new BGP routes after failures or BGP attribute changes?

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**



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

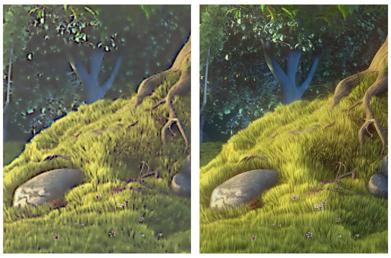
Problem: How to handle lost or reordered packets?

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

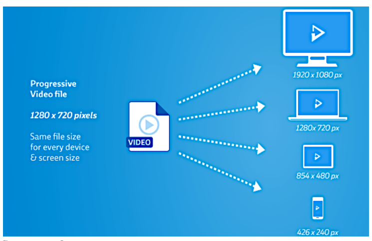
Transport Project
Your GBN sender and receiver provide some of these features
But for example, we do *not* provide fairness

Problem: How to guarantee the highest video quality?



Without seeing this ...

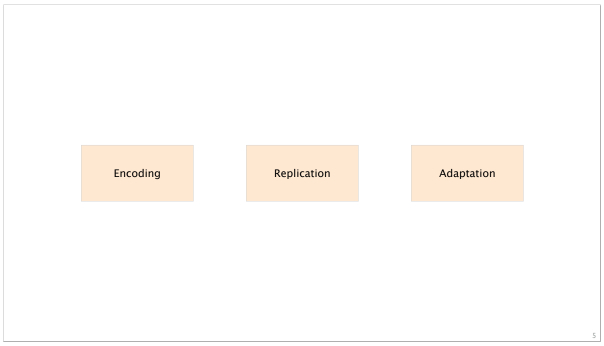
A naive approach: one-size-fits-all

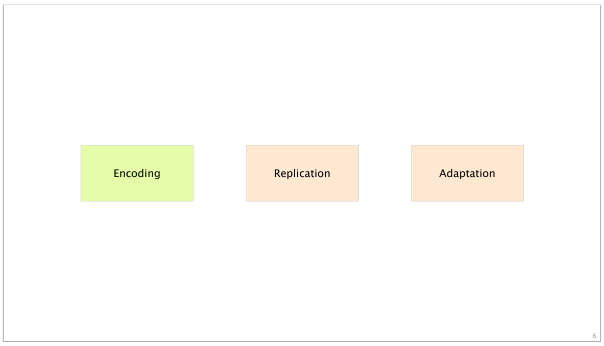


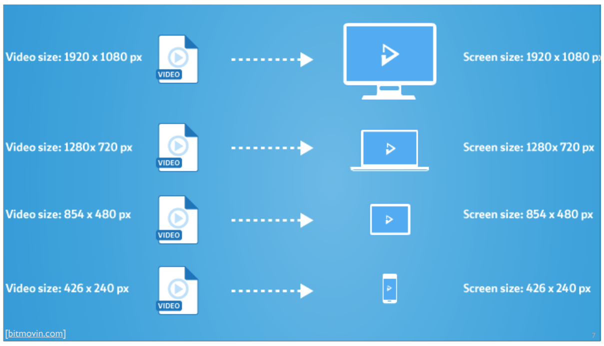
[bitmovin.com]

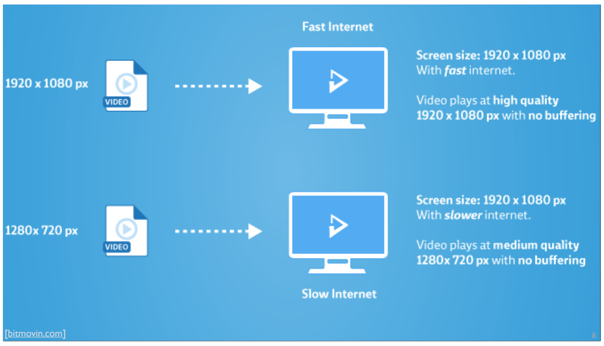
The three steps behind most contemporary solutions

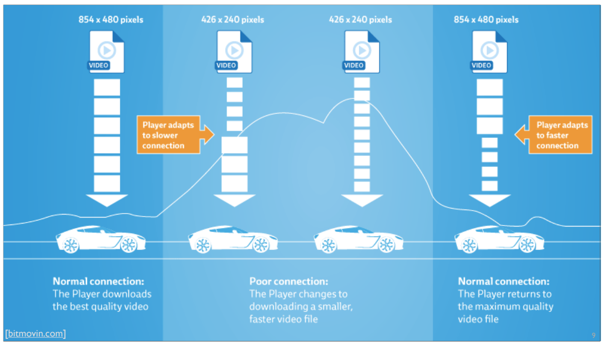
- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
  - Estimate connection's available bandwidth
  - Pick a bitrate ≤ available bandwidth









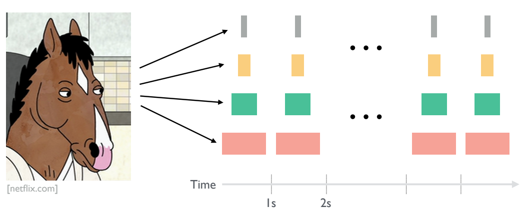


Simple solution for encoding: use a "bitrate ladders"

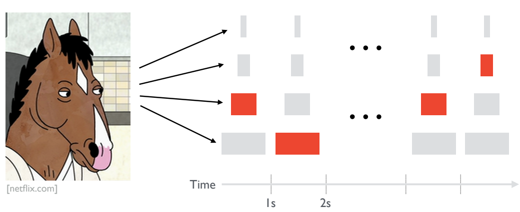
Bitrate (kbps)	Resolution
235	320x240
375	384x288
560	512x384
750	512x384
1050	640x480
1750	720x480
2350	1280x720
3000	1280x720
4300	1920x1080
5800	1920x1080

[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"/>
```

[witestlab.poly.edu]

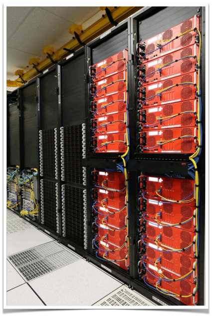
13

Encoding

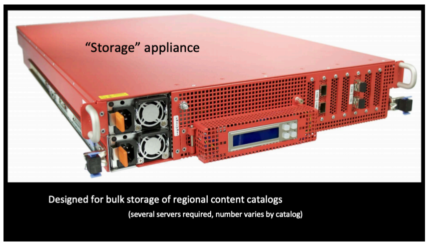
Replication

Adaptation

14



15



[more-ip-event.net]

16

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

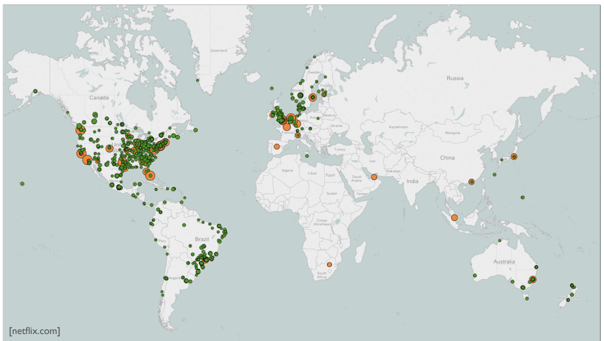
- Large storage capacity
- 2U for rack efficiency (no deeper than 29 inches)
- Enough low cost NAND to reach 10GB/s of throughput (<0.3 DWPD)
- Network flexibility to connect at 6x10GE LAG or 1x100GE
- 2 and 4 post racking
- AC or DC power
- Single processor

Storage appliance high-level specifications

Option	Vendors
Chassis	Sanmina
Motherboard	Supermicro
Processor	Intel
Memory	Micron
Hard Drive	HGST
Solid State Drive	Micron, Toshiba
Network Controller	Chelsio
Power draw operational (peak)	~500W
Power Supply Unit	Redundant Hot Swap AC/DC
Operational throughput	~36Gbps
Raw storage capacity	~288 TB

[openconnect.netflix.com]

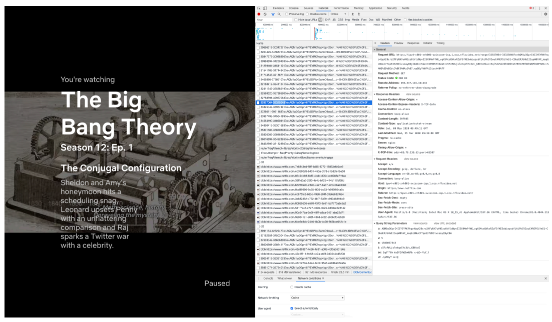
17



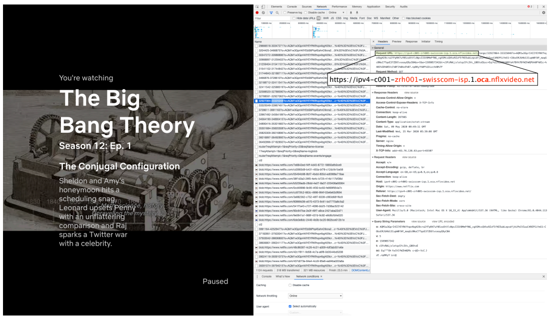
[netflix.com]

[netflix.com]

19

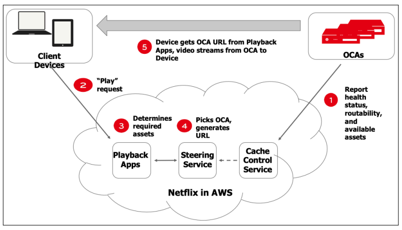


20



21

Complete Playback Workflow @Netflix



[more-ip-event.net]

22

How many OCA appliances in Swisscom?

I found at least 35 of them

ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.34	ipv4-c001-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.2
ipv4-c002-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.35	ipv4-c002-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.3
ipv4-c003-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.36	ipv4-c003-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.4
ipv4-c004-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.37	ipv4-c004-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.5
ipv4-c005-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.38	ipv4-c005-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.6
ipv4-c006-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.39	ipv4-c006-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.7
ipv4-c007-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.40	ipv4-c007-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.8
ipv4-c008-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.41	ipv4-c009-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.9
ipv4-c001-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.98	ipv4-c001-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.72
ipv4-c002-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.99	ipv4-c002-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.73
ipv4-c003-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.100	ipv4-c003-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.74
ipv4-c004-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.101	ipv4-c005-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.67
ipv4-c005-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.102	ipv4-c006-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.68
ipv4-c006-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.103	ipv4-c007-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.69
ipv4-c007-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.104	ipv4-c008-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.70
ipv4-c008-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.105	ipv4-c009-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.71
ipv4-c001-zrh003-swisscom-isp.1.oca.nflxvideo.net	193.247.193.242	ipv4-c010-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.66
ipv4-c002-zrh003-swisscom-isp.1.oca.nflxvideo.net	193.247.193.243		

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

23

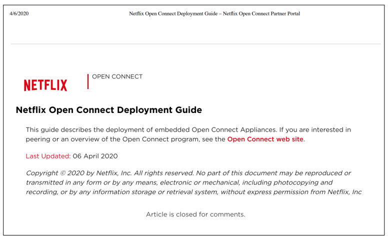
Besides OCAs within ISPs, Netflix also hosts caches at various IXPs and datacenters

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

24

If you are interested in finding out more: check out https://openconnect.netflix.com


4/6/2020 Netflix Open Connect Deployment Guide – Netflix Open Connect Partner Portal

NETFLIX | OPEN CONNECT

Netflix Open Connect Deployment Guide

This guide describes the deployment of embedded Open Connect Appliances. If you are interested in peering or an overview of the Open Connect program, see the **Open Connect web site**.

Last Updated: 06 April 2020

Copyright © 2020 by Netflix, Inc. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, without express permission from Netflix, Inc

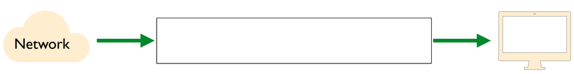
Article is closed for comments.

Deployment guide: https://openconnect.netflix.com/deploymentguide.pdf

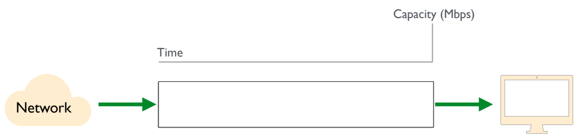
25

Encoding | Replication | Adaptation

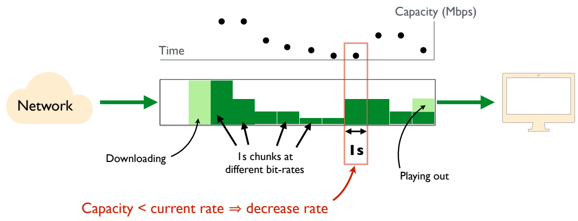
26



27



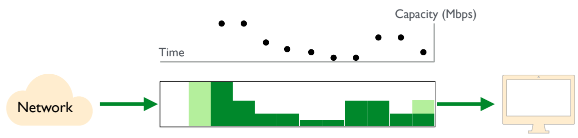
28



Common solution approach

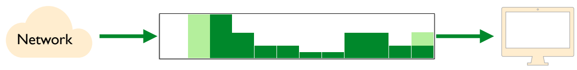
- Encode video in multiple bitrates
- Replicate using a content delivery network
- Video player picks bitrate adaptively
  - Estimate connection's available bandwidth
  - Pick a bitrate ≤ available bandwidth

Buffer-based capacity adaptation



Decide based on the buffer alone?

Buffer-based capacity adaptation

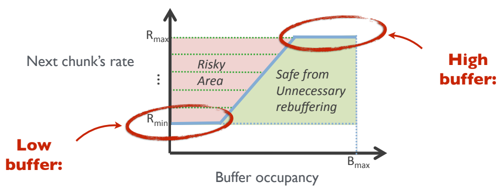


Nearly full buffer ⇒ large rate

Buffer-based capacity adaptation

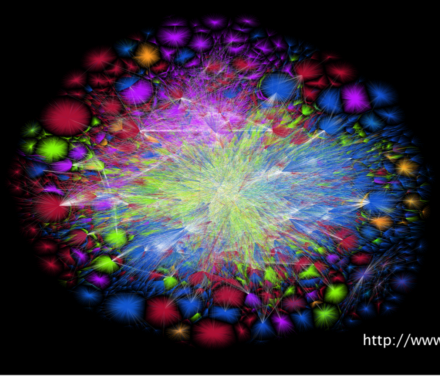
Nearly empty buffer ⇒ small rate

Buffer-based capacity adaptation



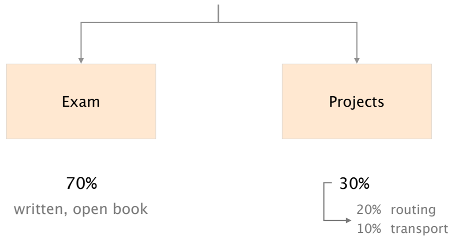
[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]

Now you (better) understand this!



http://www.opte.org

Your final grade



Your final grade

Exam

70%
written, open book

Projects

30%
20% routing
10% transport

The exam will be open book, most of the questions will be open-ended, with some multiple choices

verify your understanding of the material

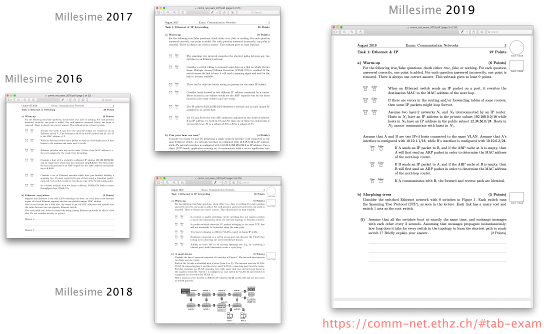
Make sure you can do *all* the exercises, especially the ones in previous exams

Millesime 2017

Millesime 2019

Millesime 2016

Millesime 2018



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)

Master-level lecture, every Fall semester

Advanced Topics in Communication Networks

Topics (examples)

- Tunneling
- Hierarchical routing
- Traffic Engineering
- Virtual Private Networks
- Quality of Service/Scheduling
- IP Multicast
- Fast Convergence
- Network virtualization
- Network programmability
- Network measurements

+ labs & a project

if you liked the routing project, you will like this lecture as well

https://adv-net.ethz.ch/

Consider doing one of your theses with our group!
bachelor, semester or master

https://nsg.ee.ethz.ch/theses/

Communication Networks

Spring 2022

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

May 30 2022