Communication Networks

Spring 2020

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

May 11 2020

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

Last week on

Communication Networks

DNS

Web

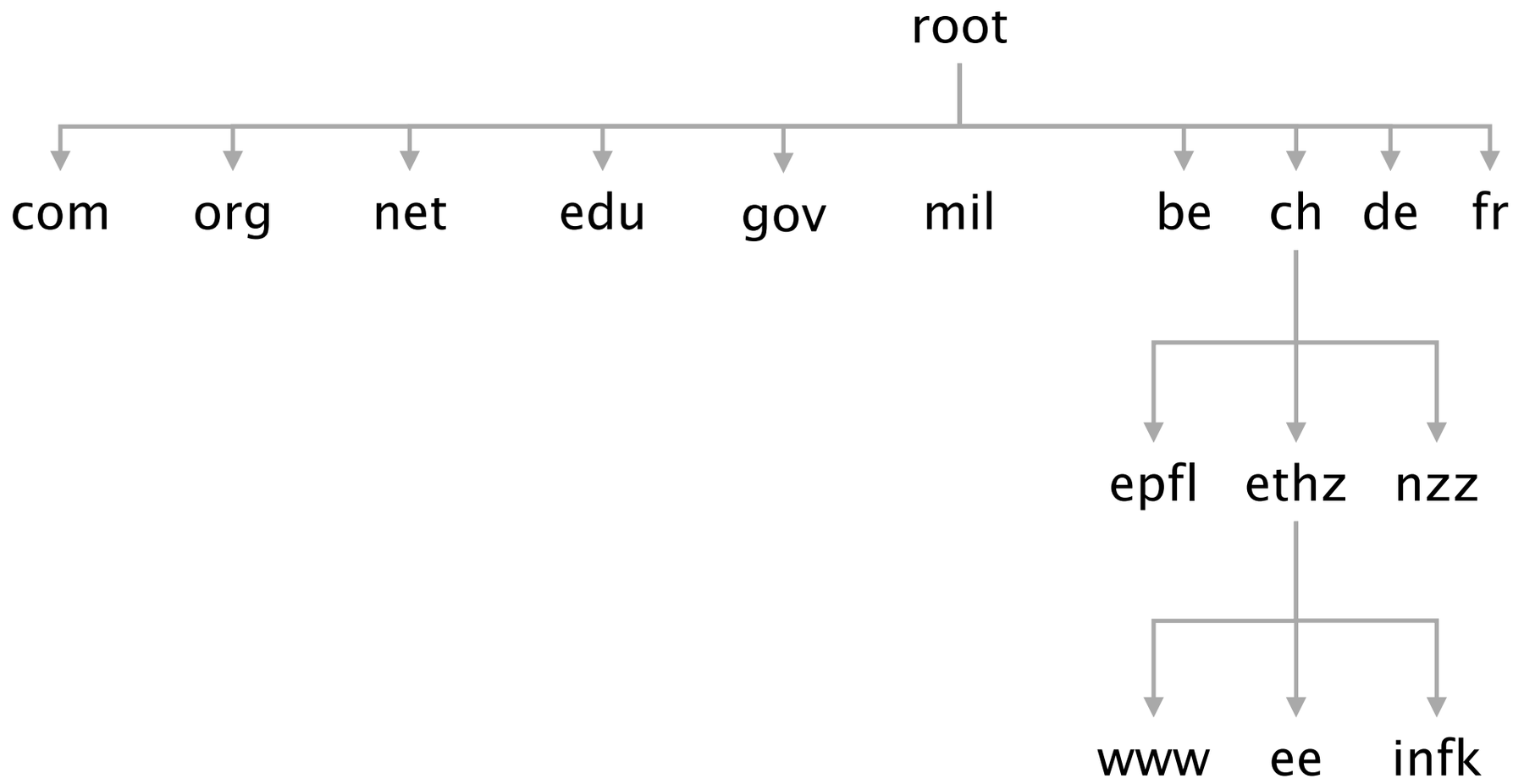
google.ch ⟷ 172.217.16.131

http://www.google.ch

infrastructure

hierarchy of DNS servers

The DNS infrastructure is hierarchically organized



13 root servers (managed professionally)

serve as root (*)

a.	root-servers.net	VeriSign, Inc.
b.	root-servers.net	University of Southern California
c.	root-servers.net	Cogent Communications
d.	root-servers.net	University of Maryland
e.	root-servers.net	NASA
f.	root-servers.net	Internet Systems Consortium
g.	root-servers.net	US Department of Defense
h.	root-servers.net	US Army
i.	root-servers.net	Netnod
j.	root-servers.net	VeriSign, Inc.
k.	root-servers.net	RIPE NCC
l.	root-servers.net	ICANN
m.	root-servers.net	WIDE Project

To scale root servers,
operators rely on BGP anycast

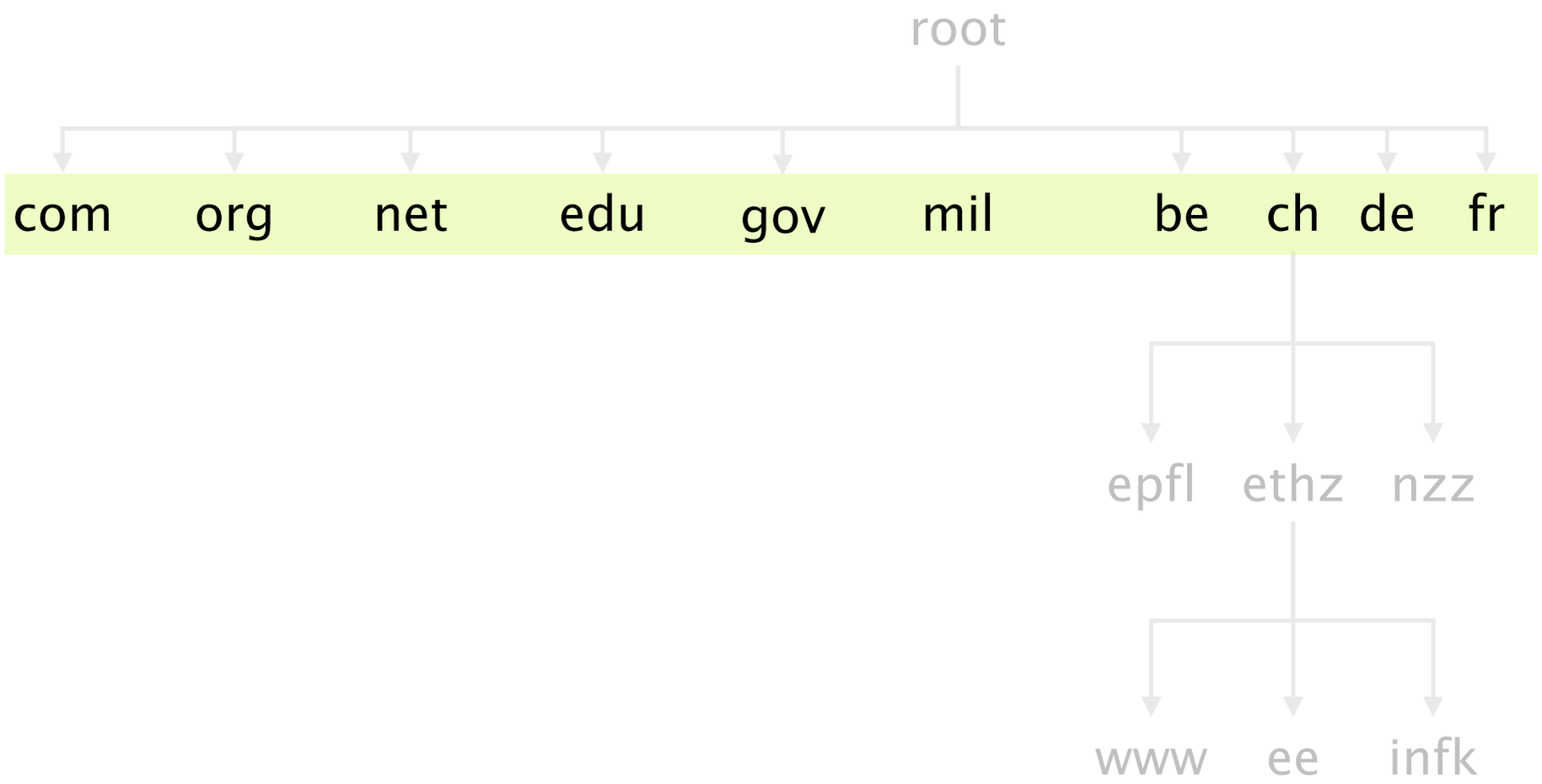
Intuition

Routing finds shortest-paths

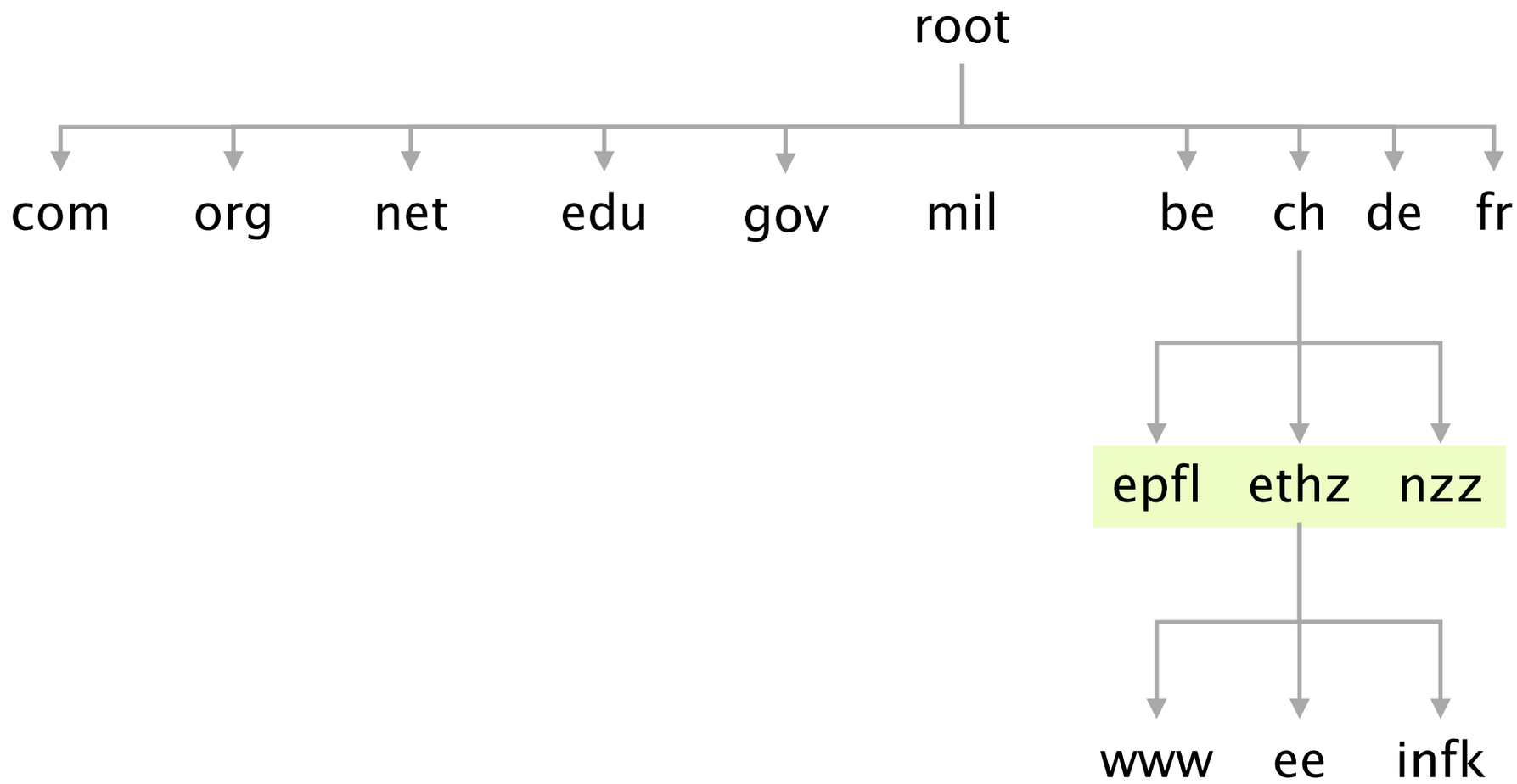
If several locations announce the same prefix,
then routing will deliver the packets to
the “closest” location

This enables seamless replications of resources

TLDs server are also managed professionally by private or non-profit organization



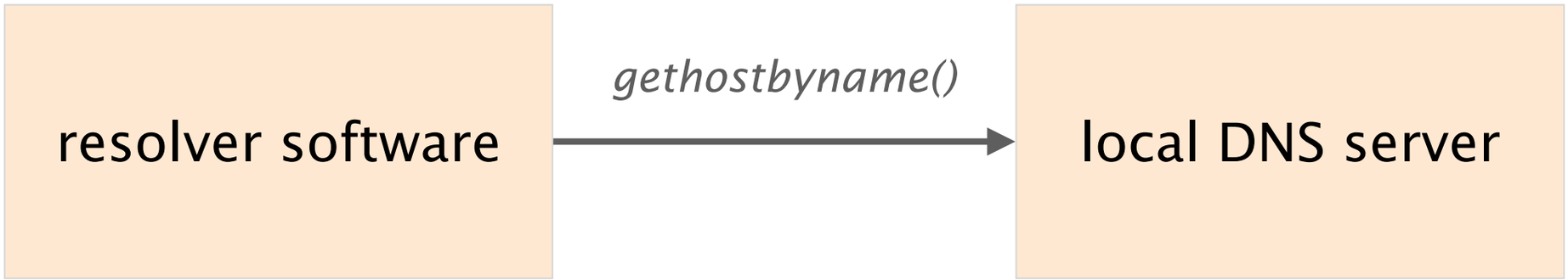
The bottom (and bulk) of the hierarchy is managed by Internet Service Provider or locally



A DNS server stores Resource Records composed of a (name, value, type, TTL)

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



trigger resolution process

send request to local DNS server

usually, near the endhosts

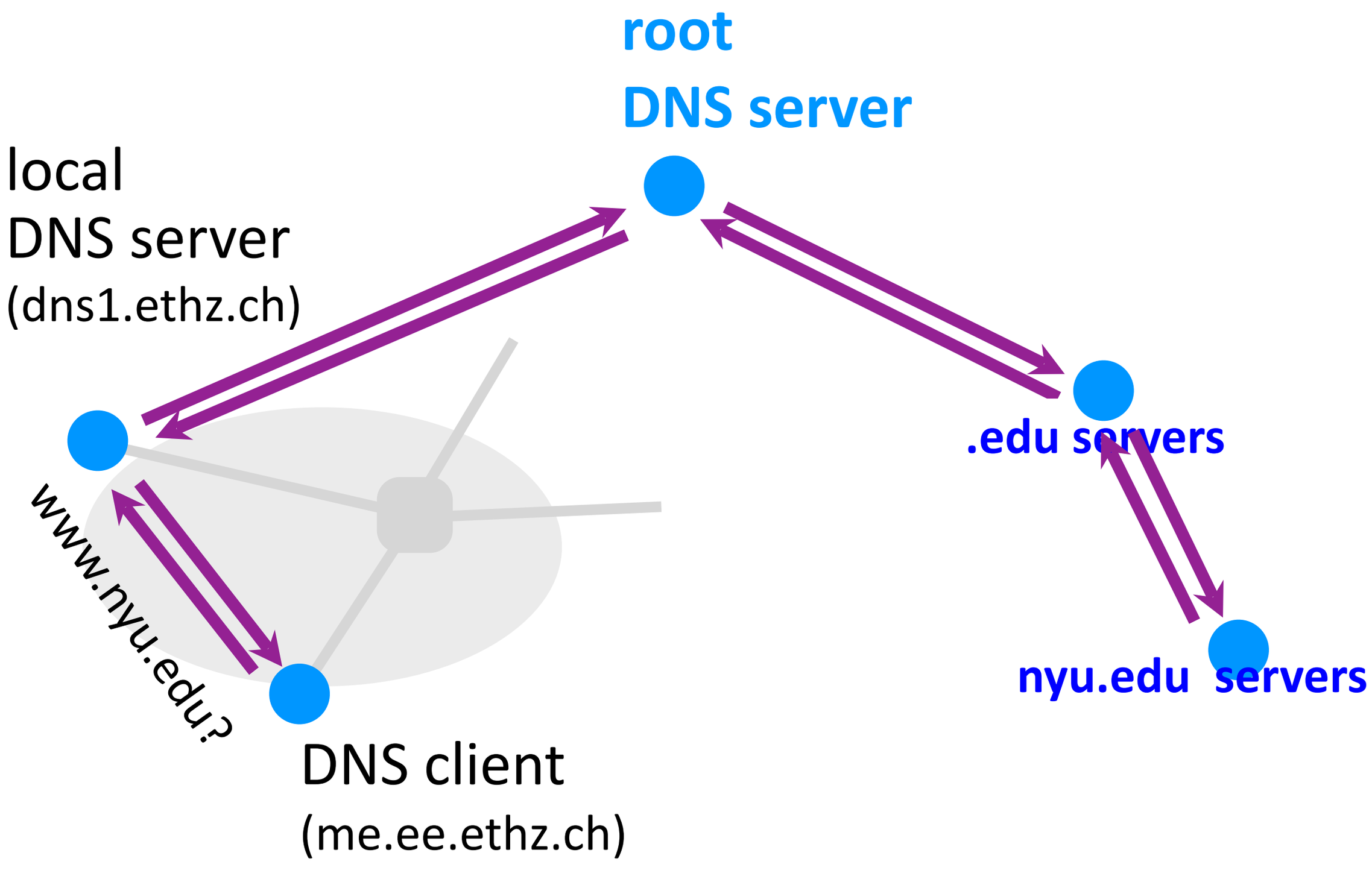
configured statically (resolv.conf)

or dynamically (DHCP)

DNS resolution can either be

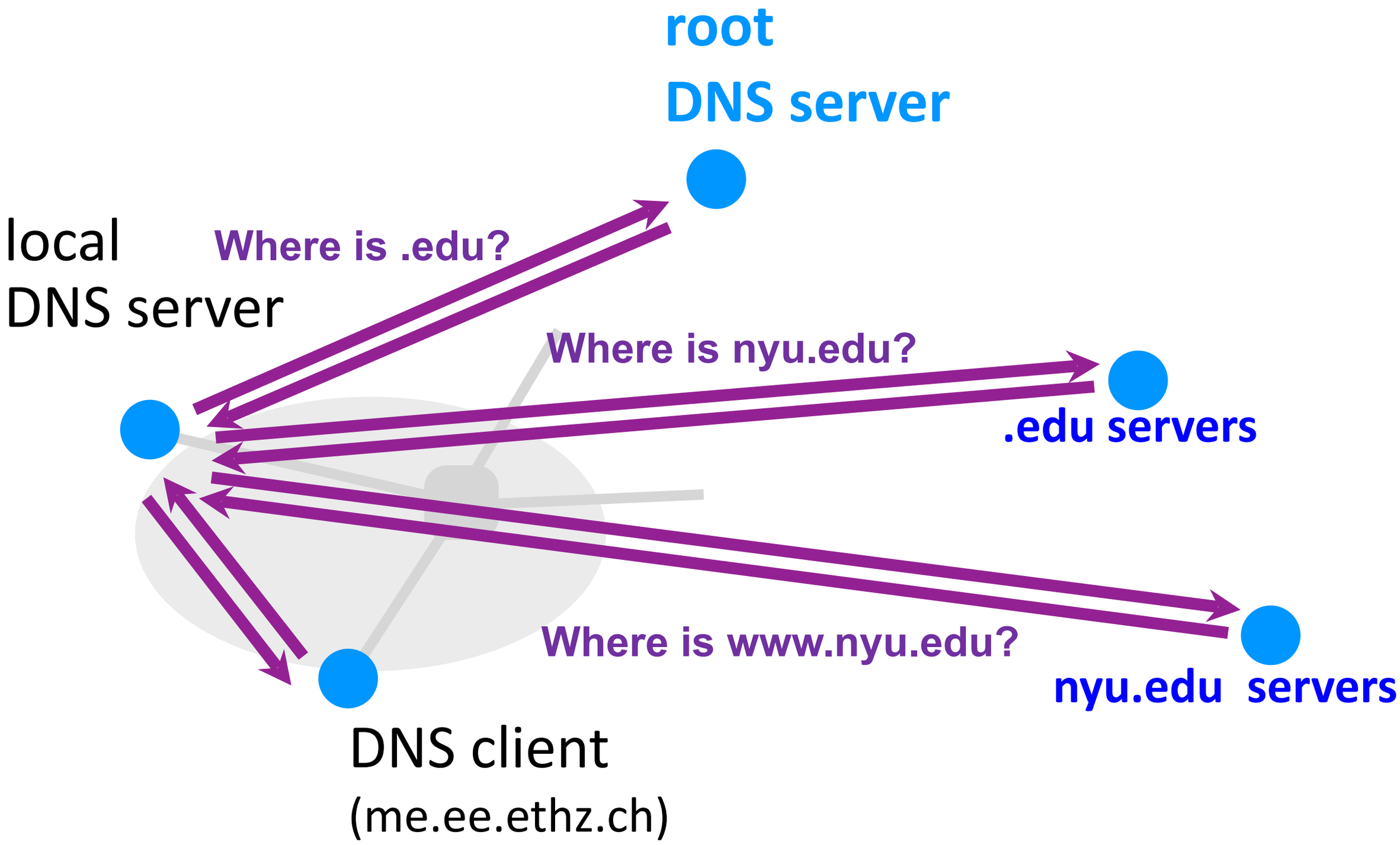
recursive or iterative

When performing a recursive query,
the client offload the task of resolving to the server



When performing a iterative query,

the server only returns the address of the "next server"



DNS

Web

http://www.google.ch

The WWW is made of three key components

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

We'll focus on
its implementation

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

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

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

protocol://hostname[:port]/directory_path/resource

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

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

http://

Protocol

Performance

http://

Protocol

Performance

HTTP clients make request to the server

HTTP request

method <sp> URL <sp> version	<cr><lf>
header field name: value	<cr><lf>
…	
header field name: value	<cr><lf>
<cr><lf>	
body	

method	GET	return resource
	HEAD	return headers only
	POST	send data to server (forms)
URL		relative to server (*e.g.*, /index.html)
version		1.0, 1.1, 2.0

HTTP servers answers to clients' requests

HTTP response

version <sp> status <sp> phrase	<cr><lf>
header field name: value	<cr><lf>
…	
header field name: value	<cr><lf>
<cr><lf>	
body	

	3 digit response code			reason phrase
Status	1XX	informational		
	2XX	success	200	OK
	3XX	redirection	301	Moved Permanently
			303	Moved Temporarily
			304	Not Modified
	4XX	client error	404	Not Found
	5XX	server error	505	Not Supported

HTTP makes the client maintain the state.

This is what the so-called cookies are for!

client stores small state

on behalf of the server *X*

client sends state

in all future requests to *X*

can provide authentication

This week on

Communication Networks

Web

Video Streaming

http://www.google.ch

HTTP-based

(the end, from slide 70/97)

Web

Video Streaming

http://www.google.ch

(the end, from slide 70/97)

Web

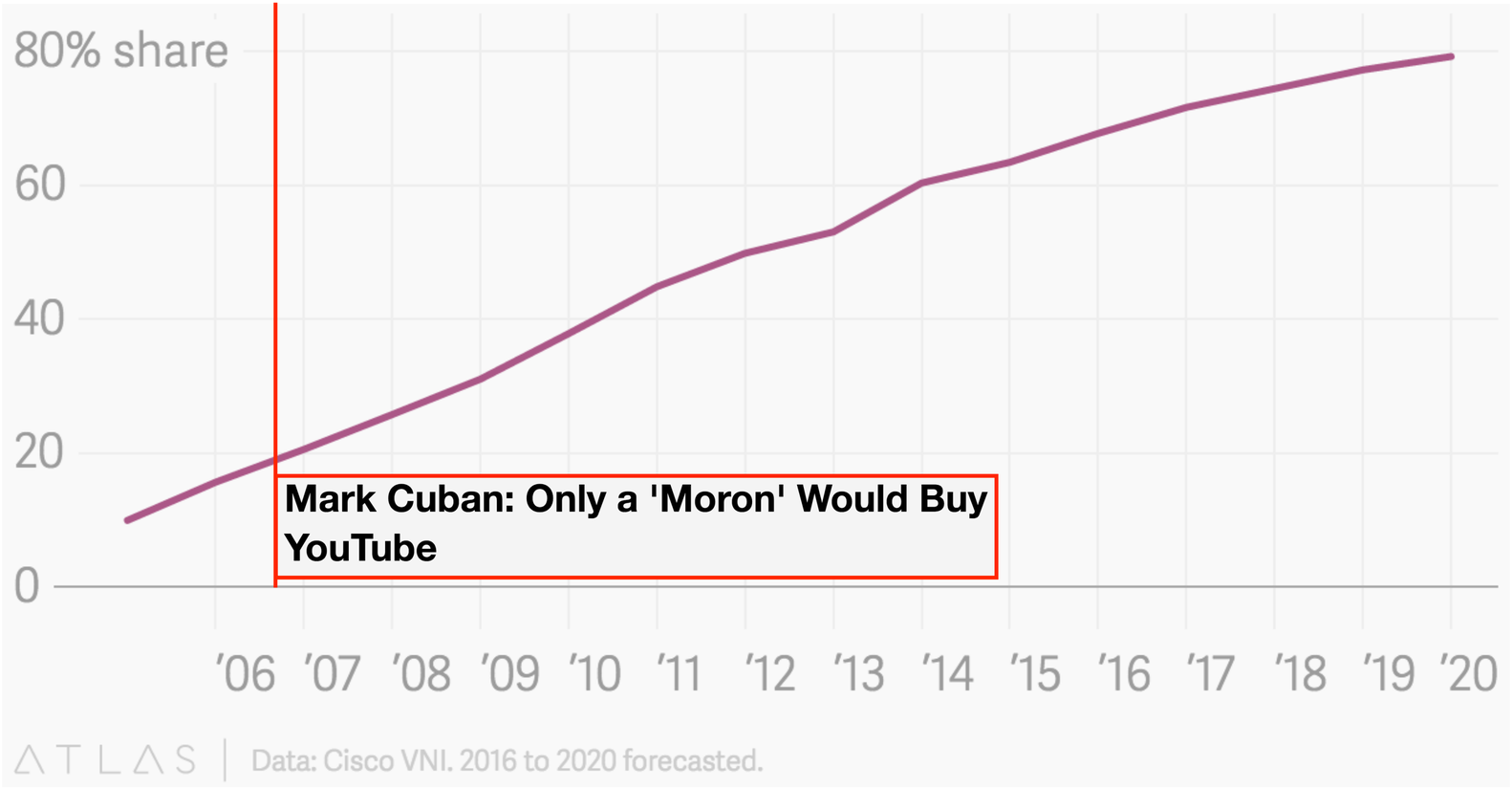
Video Streaming

HTTP-based

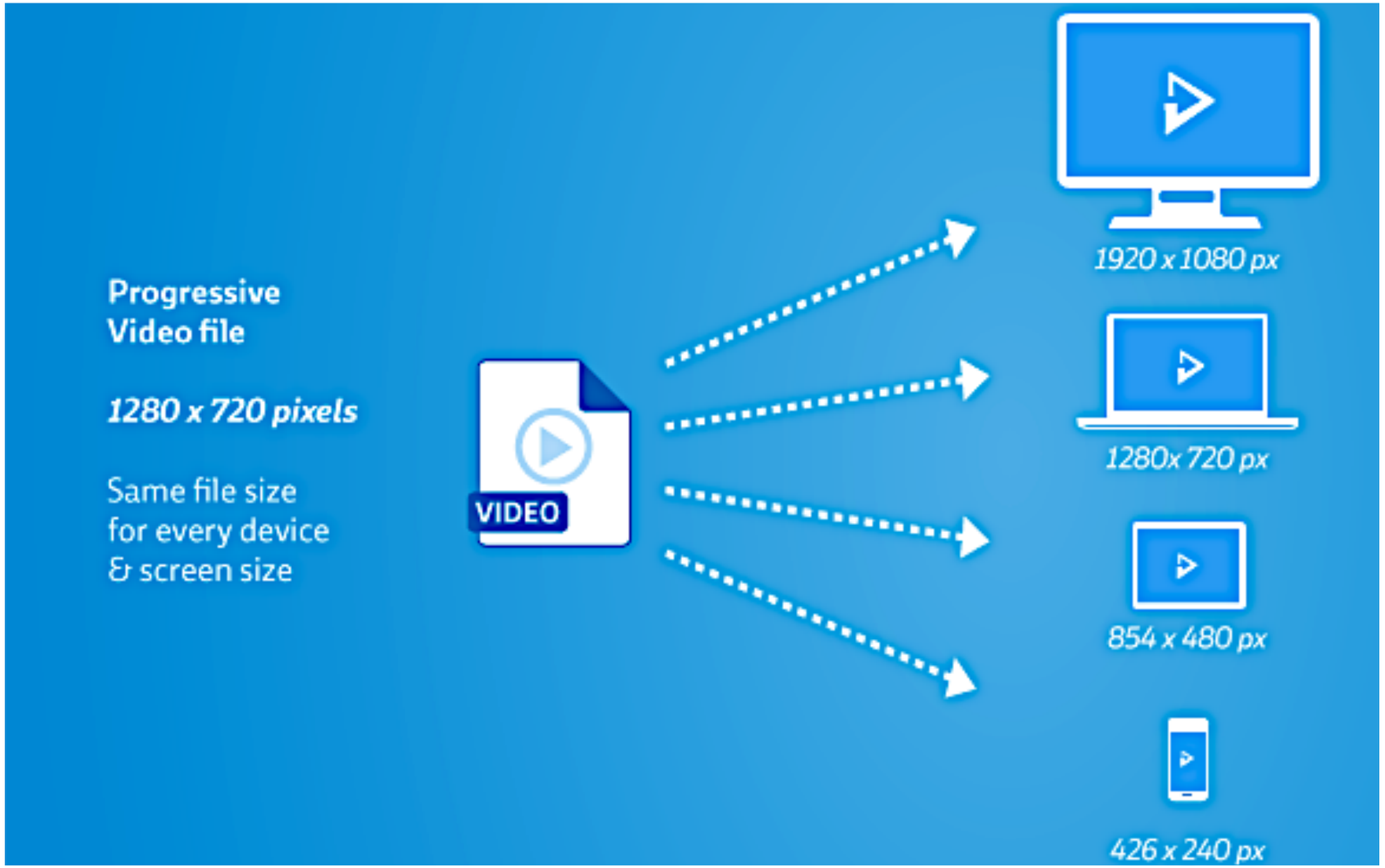
We want the highest video quality

Without seeing this …

Why should you care? Just look at this: video's share of global internet traffic

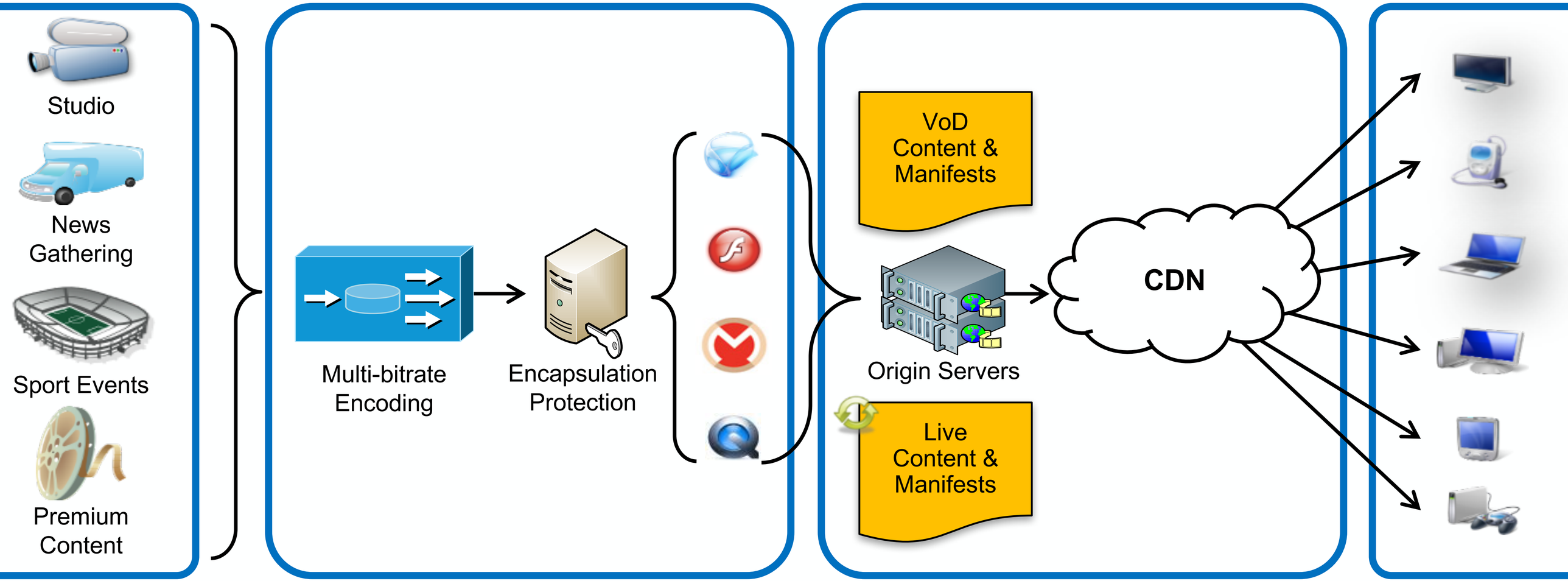


A naive approach: one-size-fits-all



[bitmovin.com]

In practice, things are complex



[Adapted from: Adaptive Streaming of Traditional and Omnidirectional Media, Begen & Timmerer, ACM SIGCOMM Tutorial, 2017]

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

Encoding

Replication

Adaptation

Video size: 1920 x 1080 px

Screen size: 1920 x 1080 px

Video size: 1280x 720 px

Screen size: 1280x 720 px

Video size: 854 x 480 px

Screen size: 854 x 480 px

Video size: 426 x 240 px

Screen size: 426 x 240 px

Fast Internet

1920 x 1080 px

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

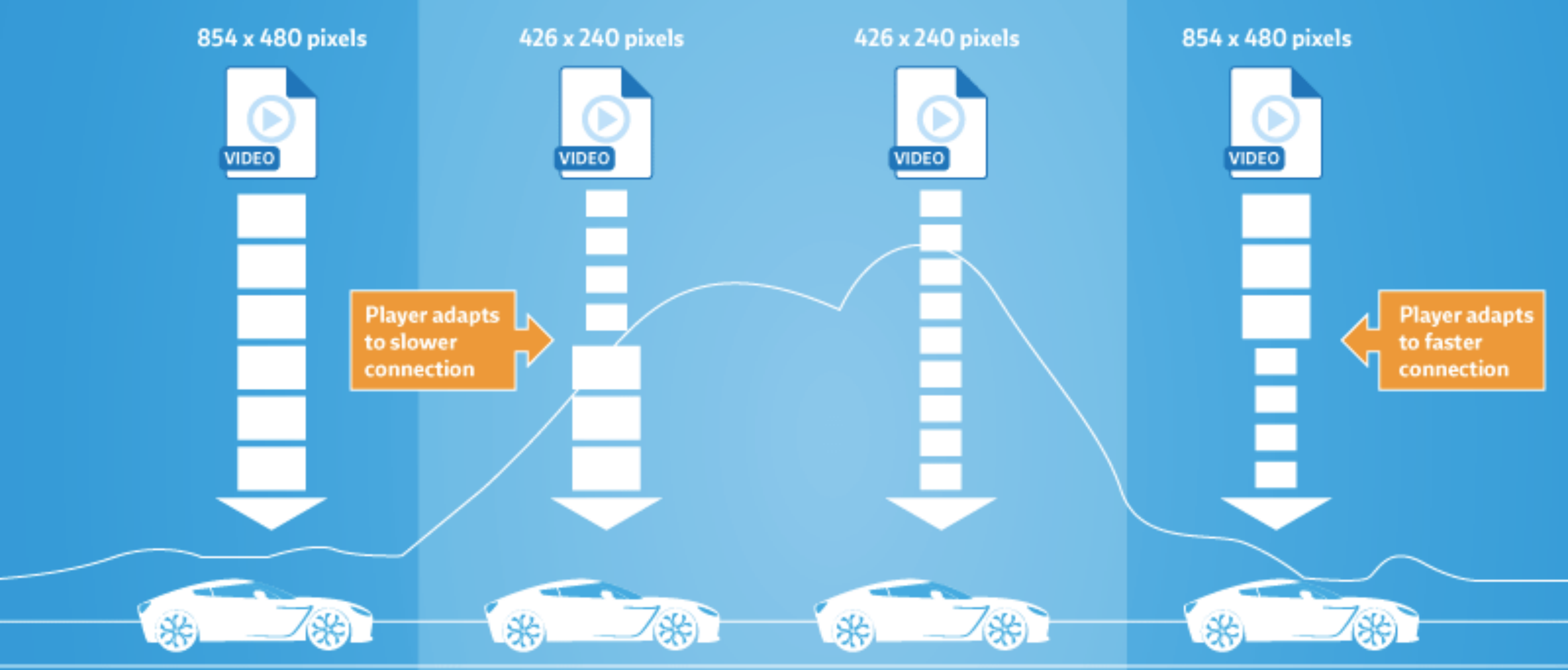
Video plays at **high quality**
1920 x 1080 px with **no buffering**

1280x 720 px

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

Video plays at **medium quality**
1280x 720 px with **no buffering**

Slow Internet


854 x 480 pixels
426 x 240 pixels
426 x 240 pixels
854 x 480 pixels
VIDEO
VIDEO
VIDEO
VIDEO
Player adapts to slower connection
Player adapts to faster connection
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”

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]

Problem: this doesn't take into account the variability in the video content (slow moving vs. fast moving)

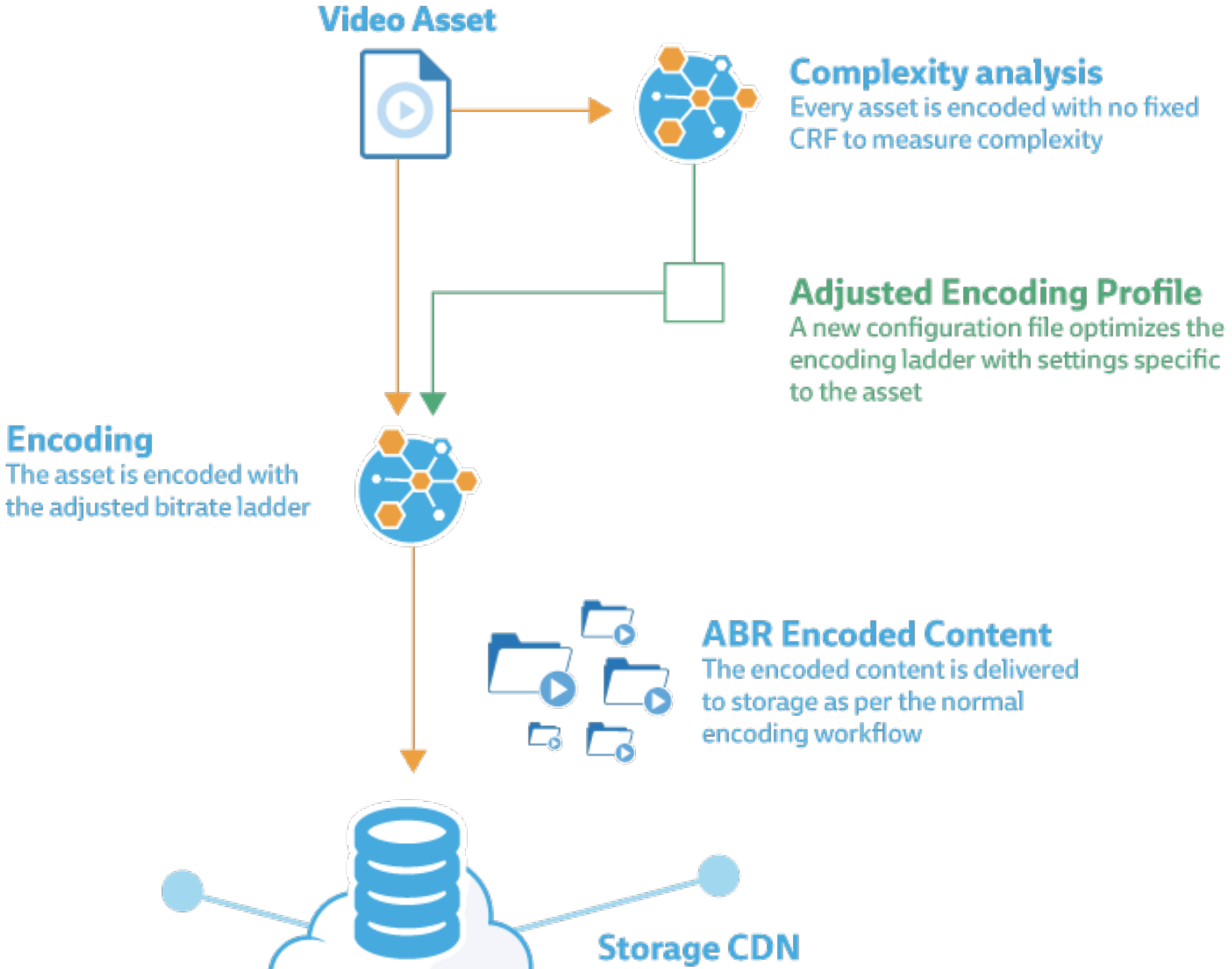
Bitrate (kbps)	Resolution
235	320x240
375	384x288
560	512x384
750	512x384
1050	640x480
1750	720x480
2350	1280x720
3000	1280x720
4300	1920x1080
5800	1920x1080

[netflix.com]

[netflix.com]

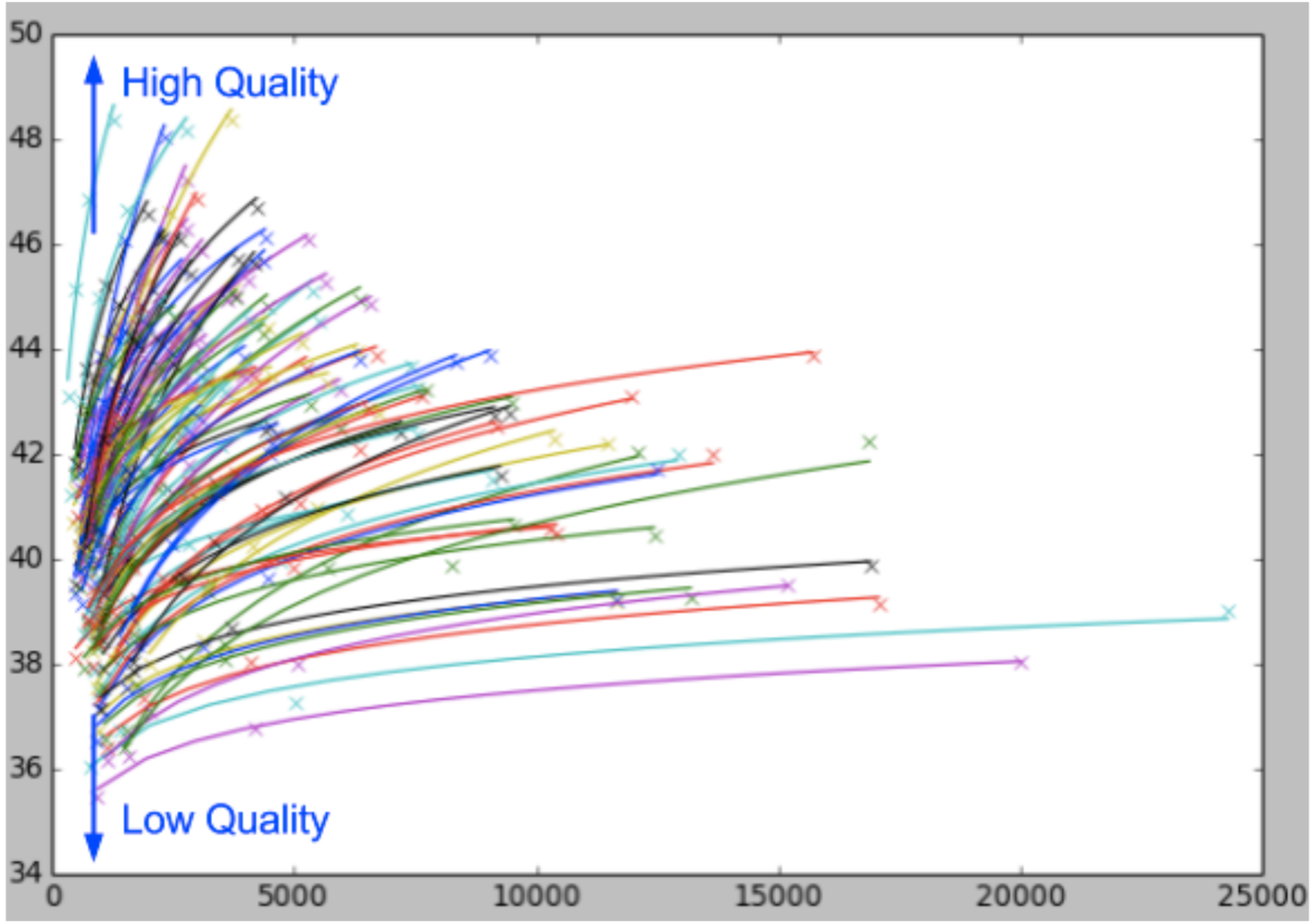
[bitmovin.com]

Encoding


Video Asset
Complexity analysis
Every asset is encoded with no fixed CRF to measure complexity
Adjusted Encoding Profile
A new configuration file optimizes the encoding ladder with settings specific to the asset
Encoding
The asset is encoded with the adjusted bitrate ladder
ABR Encoded Content
The encoded content is delivered to storage as per the normal encoding workflow
Storage CDN

Encoding

Video quality
(PSNR in dB)

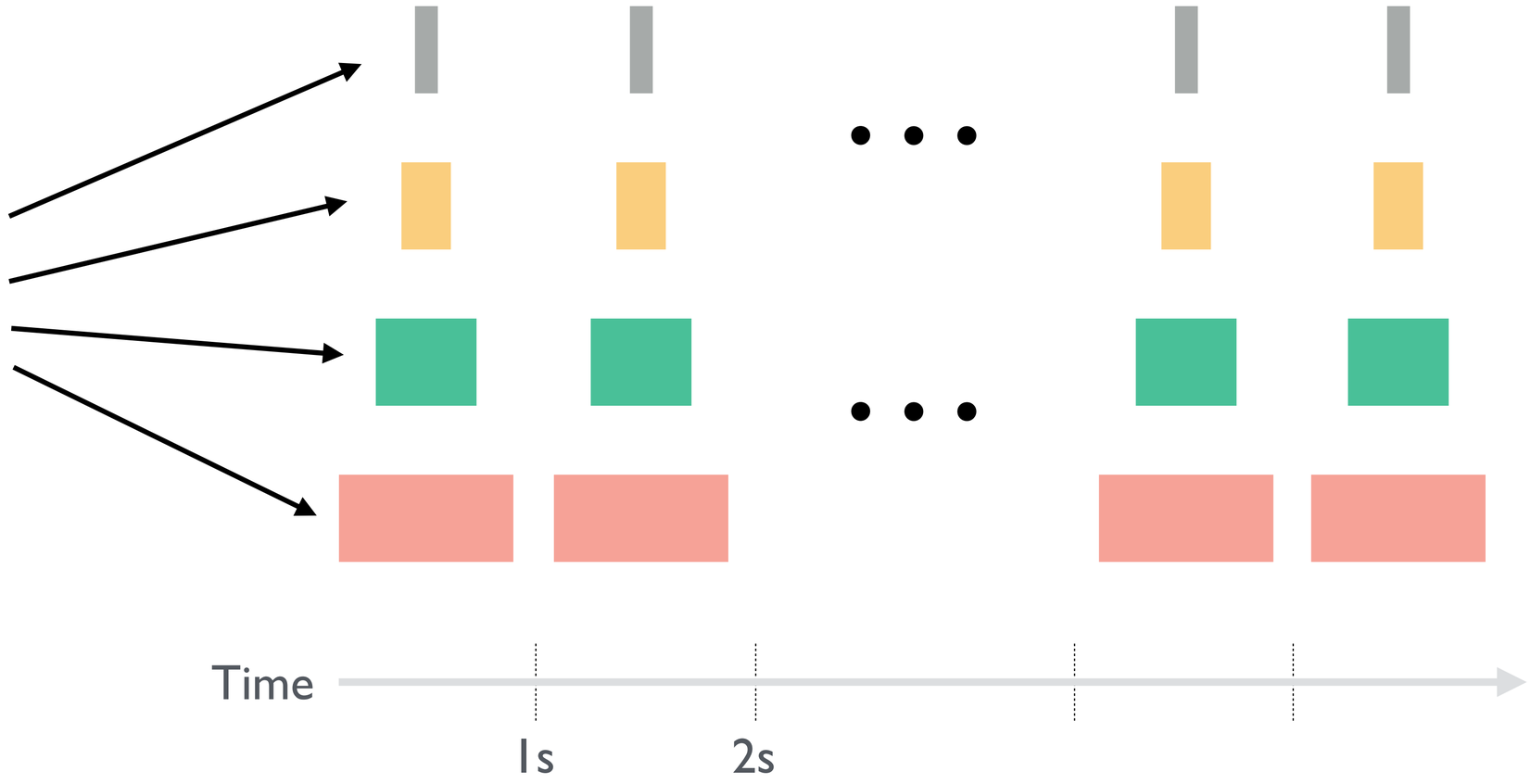


Bitrate (Kbps)

[netflix.com]

Your player download “chunks” of video at different bitrates

[netflix.com]



Depending on your network connectivity, your player fetches chunks of different qualities

[netflix.com]

Your player gets metadata about chunks via "Manifest"

```
<?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]

Encoding

Replication

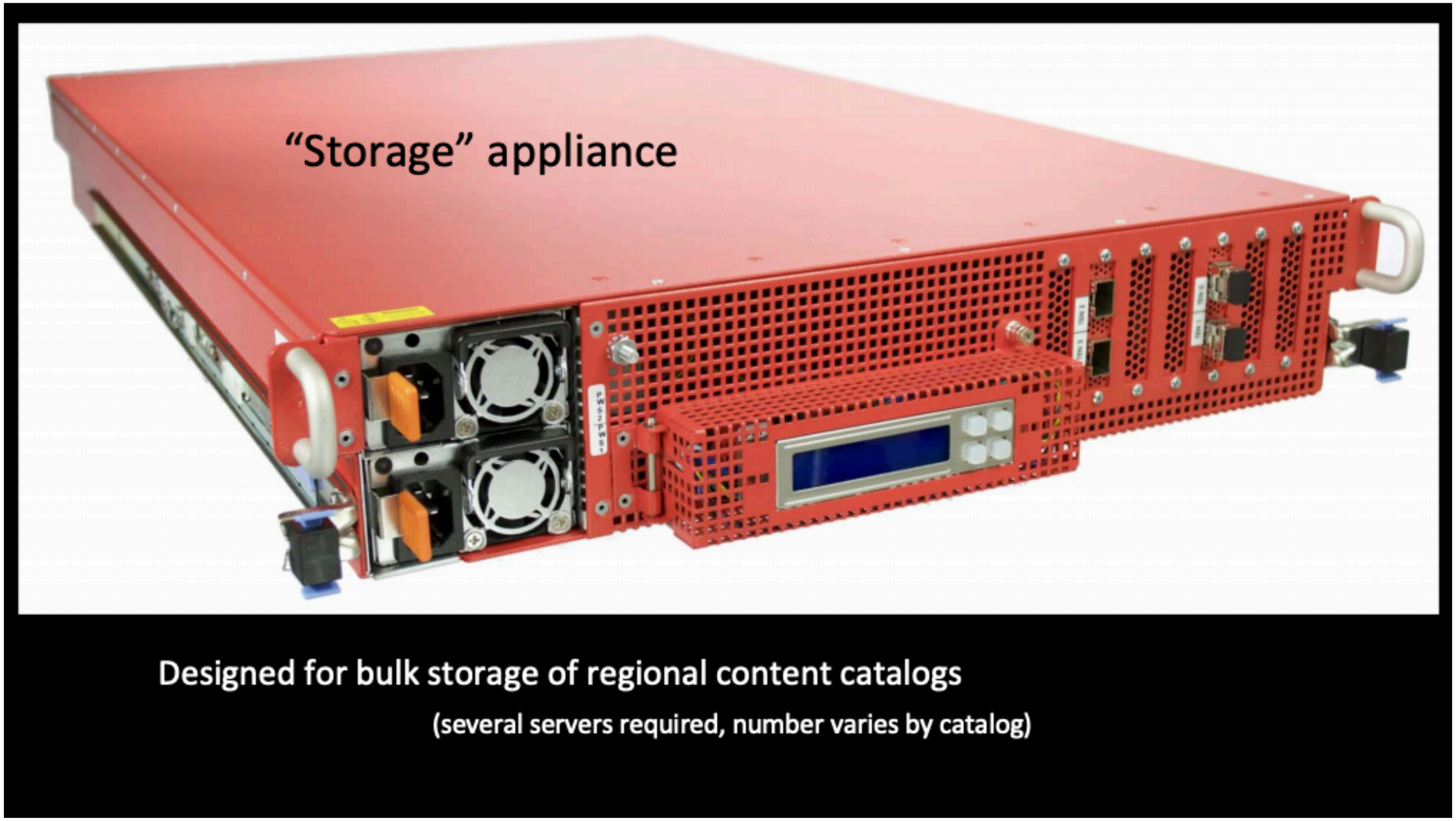
Adaptation

NETFLIX

Open Connect: Starting from a Greenfield (a mostly Layer 0 talk)

Dave Temkin

06/01/2015



[more-ip-event.net]

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.

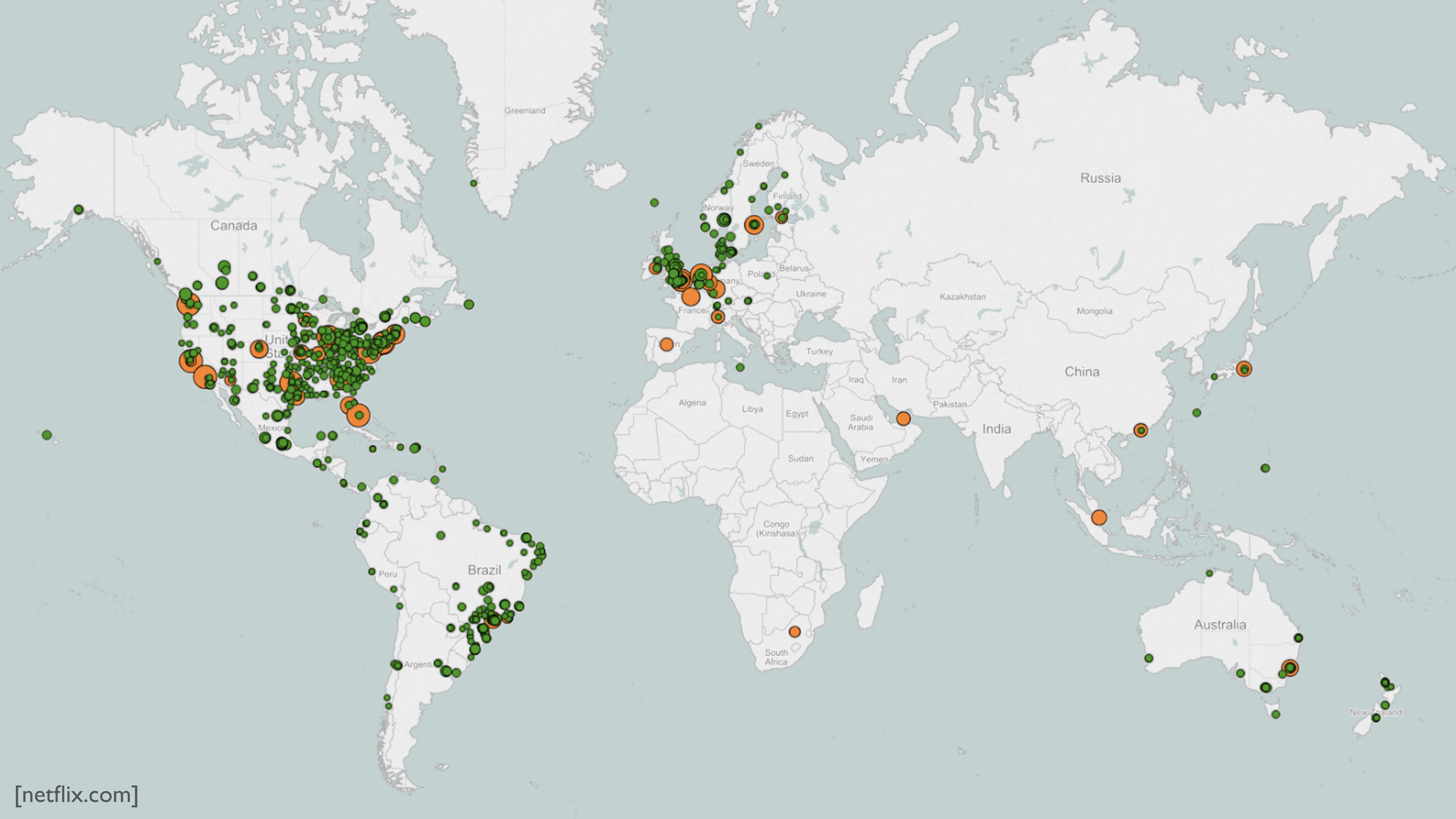
Storage appliance focus areas

- 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

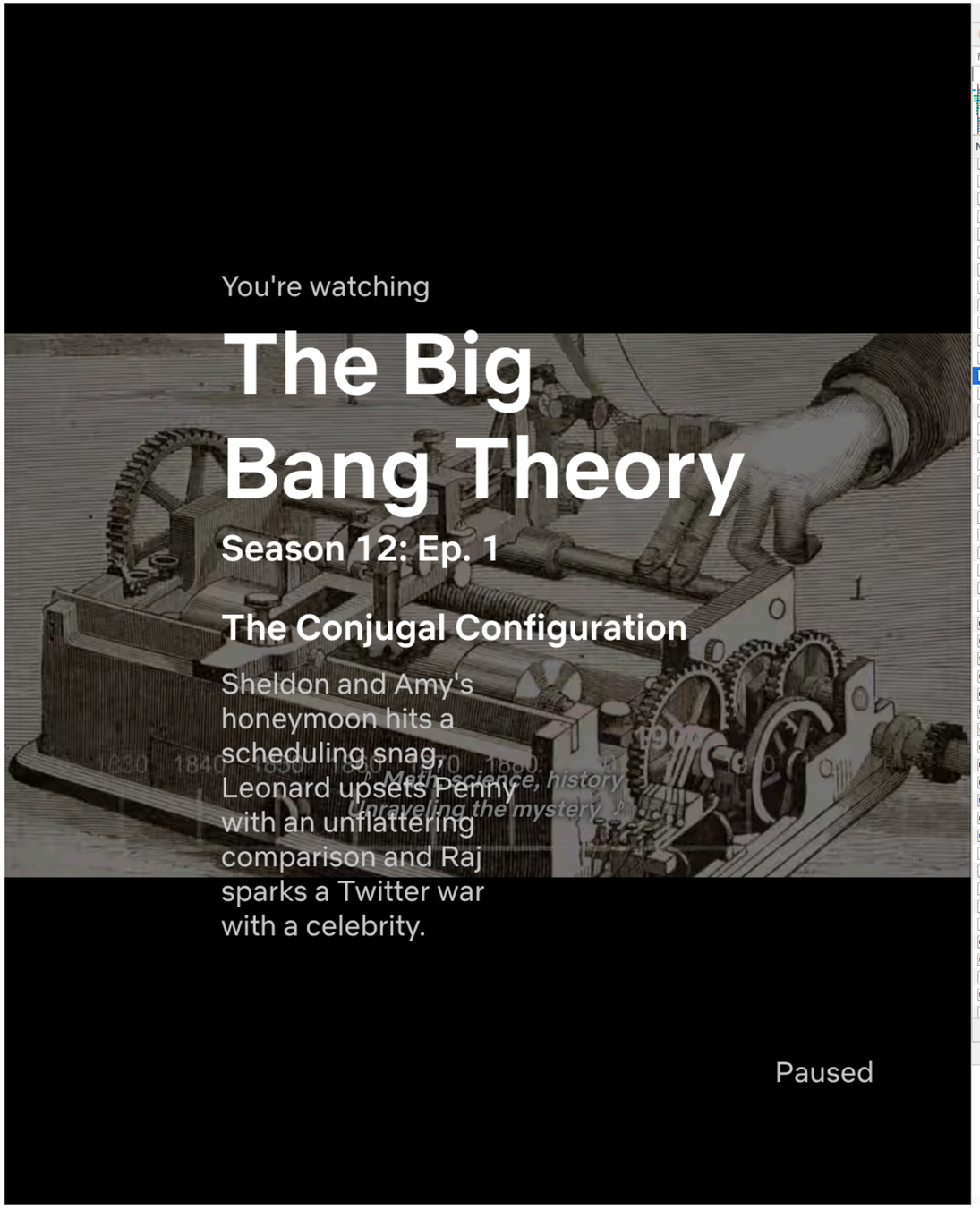
[openconnect.netflix.com]

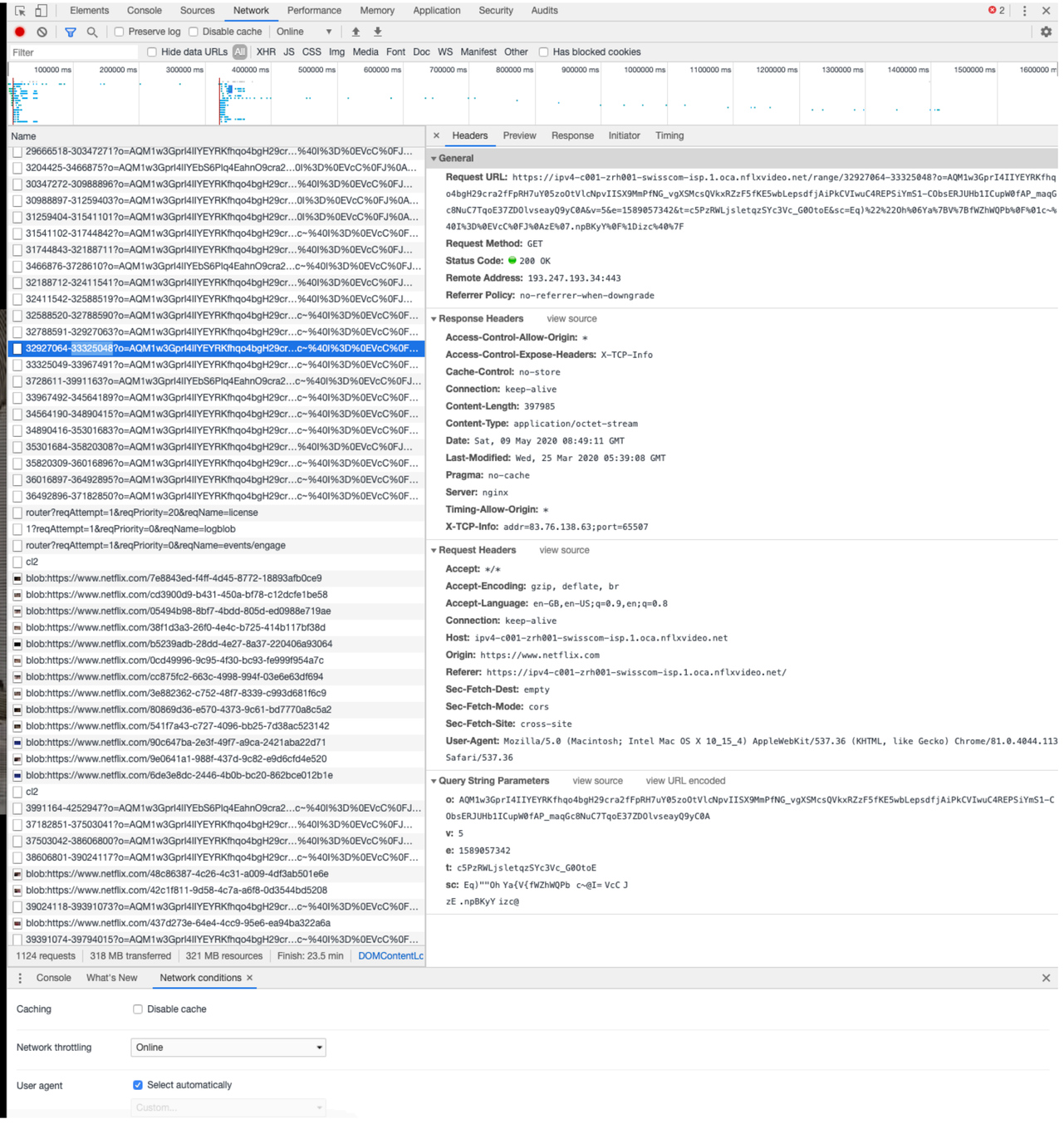

Greenland
Canada
Sweden
Finland
Norway
Russia
Belarus
Ukraine
France
Kazakhstan
Mongolia
Turkey
China
Iraq
Iran
Pakistan
Algeria
Libya
Egypt
Saudi Arabia
India
Sudan
Yemen
Congo (Kinshasa)
South Africa
Mexico
Peru
Brazil
Australia
New Zealand

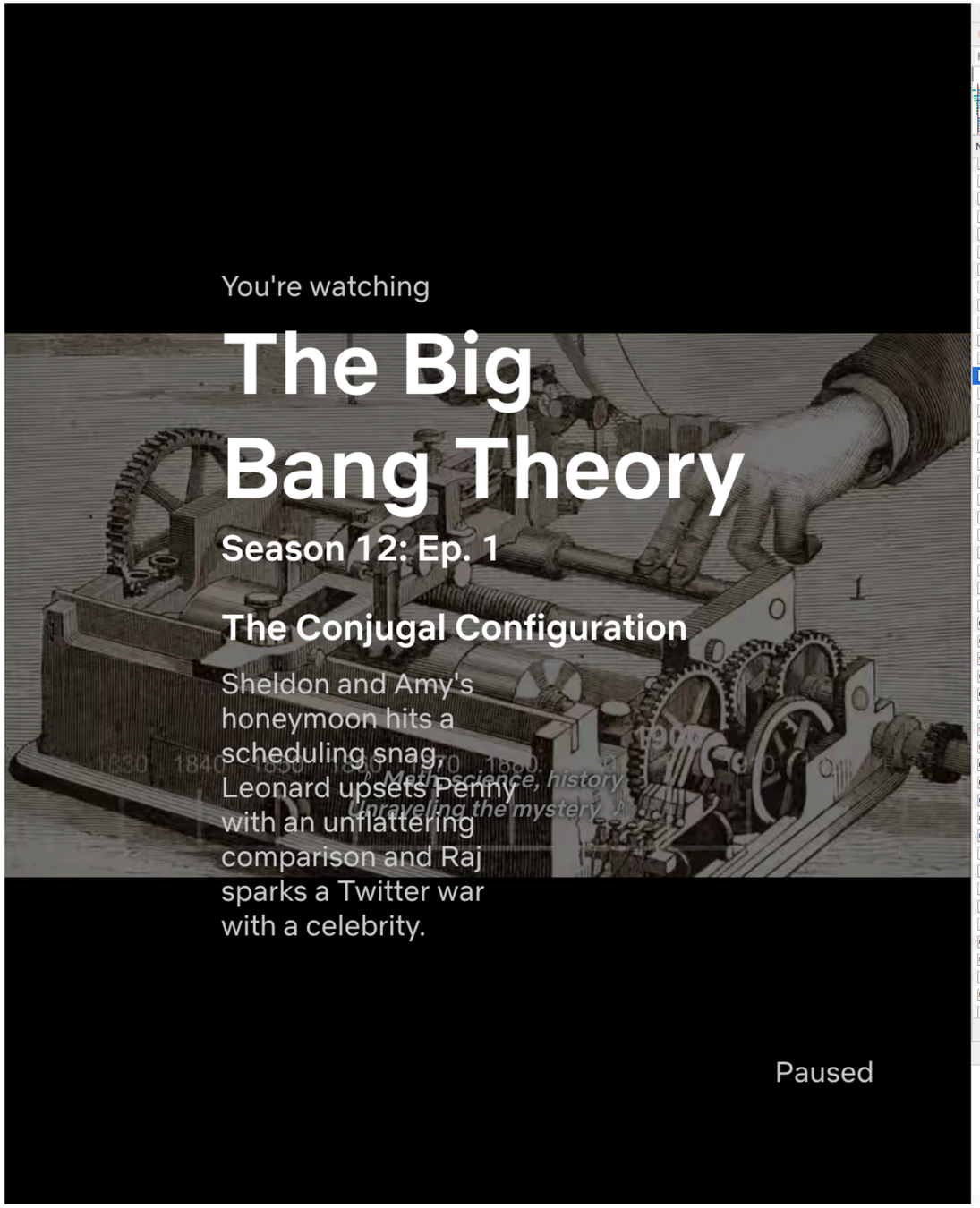
[netflix.com]

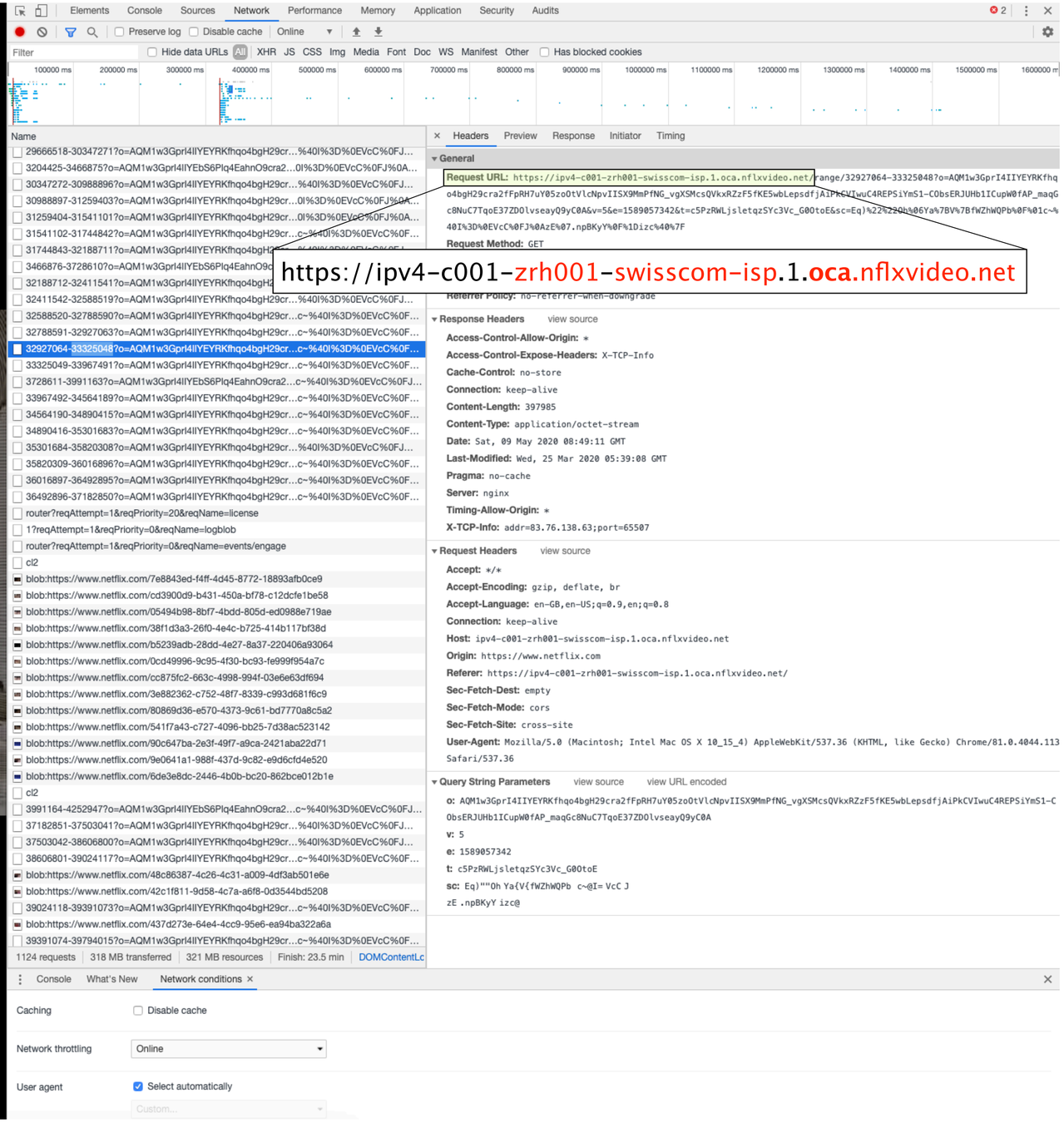

WEST COAST US
DAREDEVIL
DAREDEVIL
DAREDEVIL
AUSTRALIA
DAREDEVIL

[netflix.com]

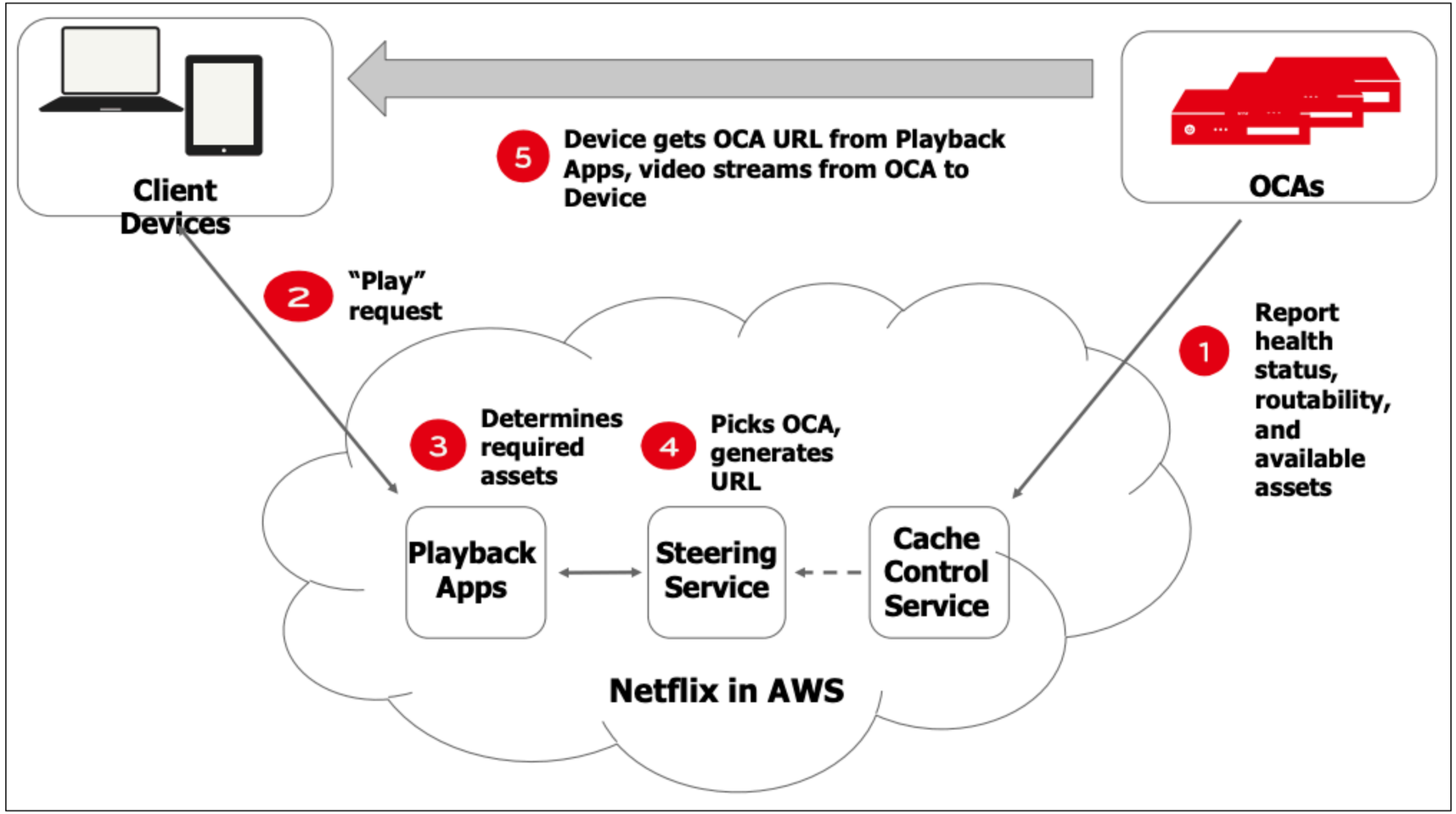

You're watching
The Big Bang Theory
Season 12: Ep. 1
The Conjugal Configuration
Sheldon and Amy's honeymoon hits a scheduling snag, Leonard upsets Penny with an unflattering comparison and Raj sparks a Twitter war with a celebrity.
Paused


Elements Console Sources Network Performance Memory Application Security Audits
Preserve log Disable cache Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies
Name
Headers Preview Response Initiator Timing
General
Request URL: https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net/range/32927064-33325048?o=AQM1w3GprI4IIYEYRKfhqo4bgH29cra2fFpRH7uY05zoOtVlcNpvIISX9MmPfNG_vgXSMcsQVkxRZzF5fKE5wbLepsdfjAiPkCVIwuC4REPSiYmS1-CObsERJUHb1ICupW0fAP_maqGc8NuC7TqoE37ZDOlvseayQ9yC0A&v=5&e=1589057342&t=c5PzRWLjsletqzSYc3Vc_G0OtoE&sc=Eq)%22%22Oh%06Ya%7BV%7BfWZhWQPb%0F%01c~%40I%3D%0EVcC%0FJ%0AzE%07.npBKyY%0F%1Dizc%40%7F
Request Method: GET
Status Code: 200 OK
Remote Address: 193.247.193.34:443
Referrer Policy: no-referrer-when-downgrade
Response Headers view source
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: X-TCP-Info
Cache-Control: no-store
Connection: keep-alive
Content-Length: 397985
Content-Type: application/octet-stream
Date: Sat, 09 May 2020 08:49:11 GMT
Last-Modified: Wed, 25 Mar 2020 05:39:08 GMT
Pragma: no-cache
Server: nginx
Timing-Allow-Origin: *
X-TCP-Info: addr=83.76.138.63;port=65507
Request Headers view source
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: keep-alive
Host: ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net
Origin: https://www.netflix.com
Referer: https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: cross-site
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36
Query String Parameters view source view URL encoded
o: AQM1w3GprI4IIYEYRKfhqo4bgH29cra2fFpRH7uY05zoOtVlcNpvIISX9MmPfNG_vgXSMcsQVkxRZzF5fKE5wbLepsdfjAiPkCVIwuC4REPSiYmS1-CObsERJUHb1ICupW0fAP_maqGc8NuC7TqoE37ZDOlvseayQ9yC0A
v: 5
e: 1589057342
t: c5PzRWLjsletqzSYc3Vc_G0OtoE
sc: Eq)""Oh Ya{V{fWZhWQPb c~@I= VcC J
zE .npBKyY izc@
1124 requests 318 MB transferred 321 MB resources Finish: 23.5 min DOMContentLc
Console What's New Network conditions
Caching Disable cache
Network throttling Online
User agent Select automatically


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


Elements Console Sources Network Performance Memory Application Security Audits
Preserve log Disable cache Online
Filter Hide data URLs All XHR JS CSS Img Media Font Doc WS Manifest Other Has blocked cookies
Name
Headers Preview Response Initiator Timing
General
Request URL: https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net/range/32927064-33325048?o=AQM1w3GprI4IIYEYRKfhqo4bgH29cra2fFpRH7uY05zoOtVlcNpvIISX9MmPfNG_vgXSMcsQVkxRZzF5fKE5wbLepsdfjAiPkCVIwuC4REPSiYmS1-CObsERJUHb1ICupW0fAP_maqGc8NuC7TqoE37ZDOlvseayQ9yC0A&v=5&e=1589057342&t=c5PzRWLjsletqzSyc3Vc_G0OtoE&sc=Eq)%22%22Oh%06Ya%7BV%7BfWZhWQPb%0F%01c~%40I%3D%0EVcC%0FJ%0AzE%07.npBKyY%0F%1Dizc%40%7F
Request Method: GET
https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net
Response Headers view source
Access-Control-Allow-Origin: *
Access-Control-Expose-Headers: X-TCP-Info
Cache-Control: no-store
Connection: keep-alive
Content-Length: 397985
Content-Type: application/octet-stream
Date: Sat, 09 May 2020 08:49:11 GMT
Last-Modified: Wed, 25 Mar 2020 05:39:08 GMT
Pragma: no-cache
Server: nginx
Timing-Allow-Origin: *
X-TCP-Info: addr=83.76.138.63;port=65507
Request Headers view source
Accept: */*
Accept-Encoding: gzip, deflate, br
Accept-Language: en-GB,en-US;q=0.9,en;q=0.8
Connection: keep-alive
Host: ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net
Origin: https://www.netflix.com
Referer: https://ipv4-c001-zrh001-swisscom-isp.1.oca.nflxvideo.net/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: cross-site
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/81.0.4044.113 Safari/537.36
Query String Parameters view source view URL encoded
o: AQM1w3GprI4IIYEYRKfhqo4bgH29cra2fFpRH7uY05zoOtVlcNpvIISX9MmPfNG_vgXSMcsQVkxRZzF5fKE5wbLepsdfjAiPkCVIwuC4REPSiYmS1-CObsERJUHb1ICupW0fAP_maqGc8NuC7TqoE37ZDOlvseayQ9yC0A
v: 5
e: 1589057342
t: c5PzRWLjsletqzSyc3Vc_G0OtoE
sc: Eq)""Oh Ya{V{fWZhWQPb c~@I= VcC J
zE .npBKyY izc@
1124 requests 318 MB transferred 321 MB resources Finish: 23.5 min DOMContentLc
Console What's New Network conditions
Caching Disable cache
Network throttling Online
User agent Select automatically

Complete Playback Workflow @Netflix



[more-ip-event.net]

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-c002-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.35
ipv4-c003-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.36
ipv4-c004-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.37
ipv4-c005-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.38
ipv4-c006-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.39
ipv4-c007-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.40
ipv4-c008-zrh001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.41
ipv4-c001-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.98
ipv4-c002-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.99
ipv4-c003-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.100
ipv4-c004-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.101
ipv4-c005-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.102
ipv4-c006-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.103
ipv4-c007-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.104
ipv4-c008-zrh002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.105
ipv4-c001-zrh003-swisscom-isp.1.oca.nflxvideo.net	193.247.193.242
ipv4-c002-zrh003-swisscom-isp.1.oca.nflxvideo.net	193.247.193.243

ipv4-c001-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.2
ipv4-c002-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.3
ipv4-c003-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.4
ipv4-c004-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.5
ipv4-c005-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.6
ipv4-c006-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.7
ipv4-c007-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.8
ipv4-c009-gva001-swisscom-isp.1.oca.nflxvideo.net	193.247.193.9
ipv4-c001-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.72
ipv4-c002-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.73
ipv4-c003-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.74
ipv4-c005-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.67
ipv4-c006-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.68
ipv4-c007-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.69
ipv4-c008-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.70
ipv4-c009-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.71
ipv4-c010-gva002-swisscom-isp.1.oca.nflxvideo.net	193.247.193.66

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

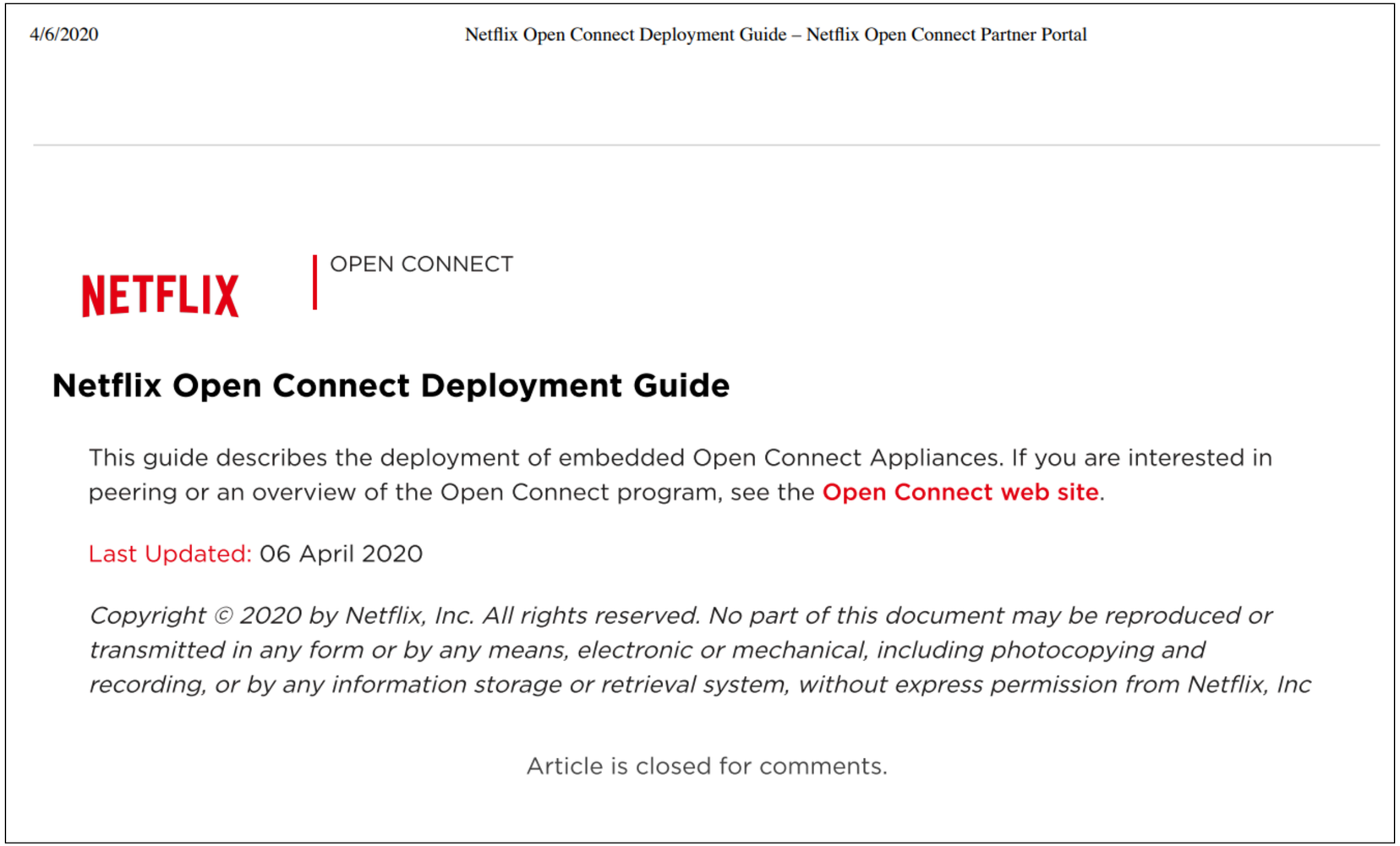
>2 million 1080p movies, assuming 100 min encoded at 5 Mbps

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



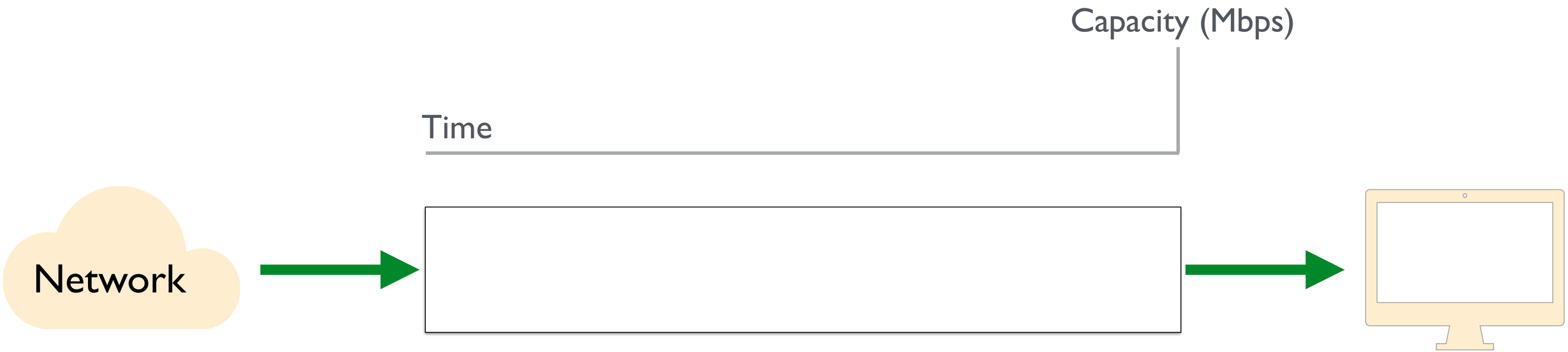
Deployment guide: https://openconnect.netflix.com/deploymentguide.pdf

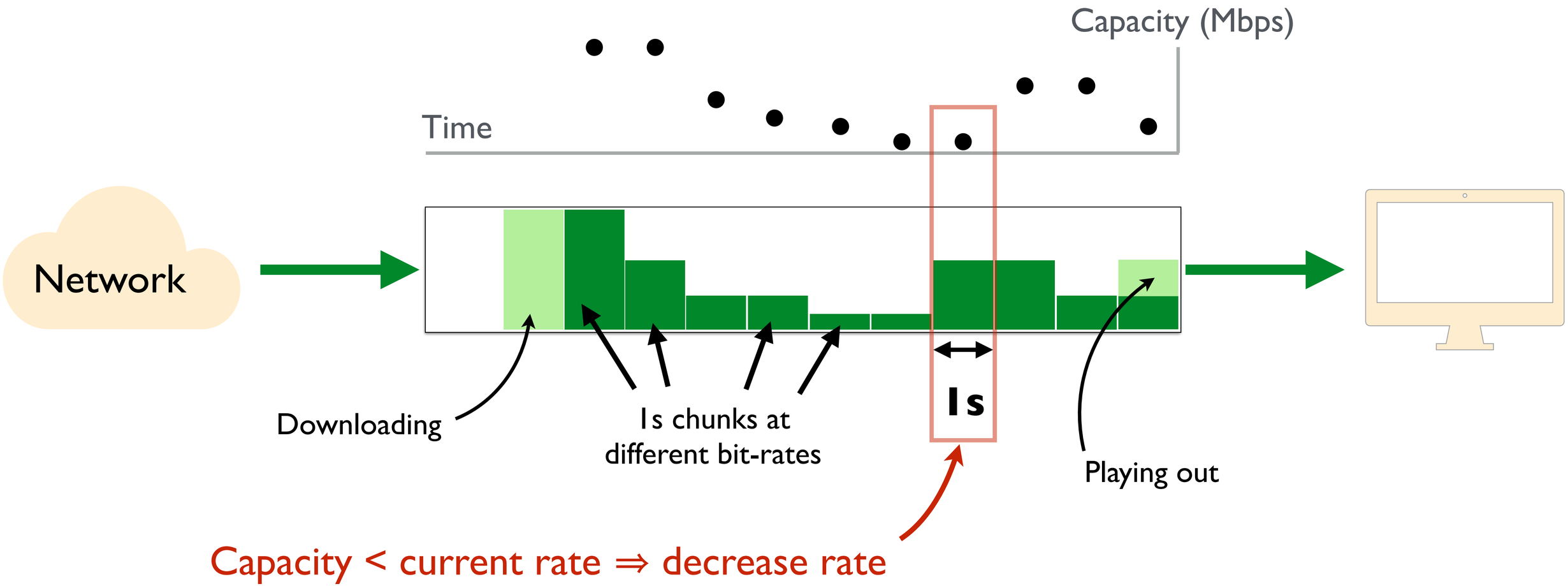
Encoding

Replication

Adaptation


Network


Capacity (Mbps)
Time
Network


Capacity (Mbps)
Time
Network
Downloading
1s chunks at
different bit-rates
1s
Playing out
Capacity < current rate ⇒ decrease rate

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

Estimating available capacity

ACM SIGCOMM

A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell*, Mark Watson*
Stanford University, Netflix*

"A random sample of 300,000 Netflix sessions shows that roughly 10% of sessions experience a median throughput less than half of the 95th percentile throughput."

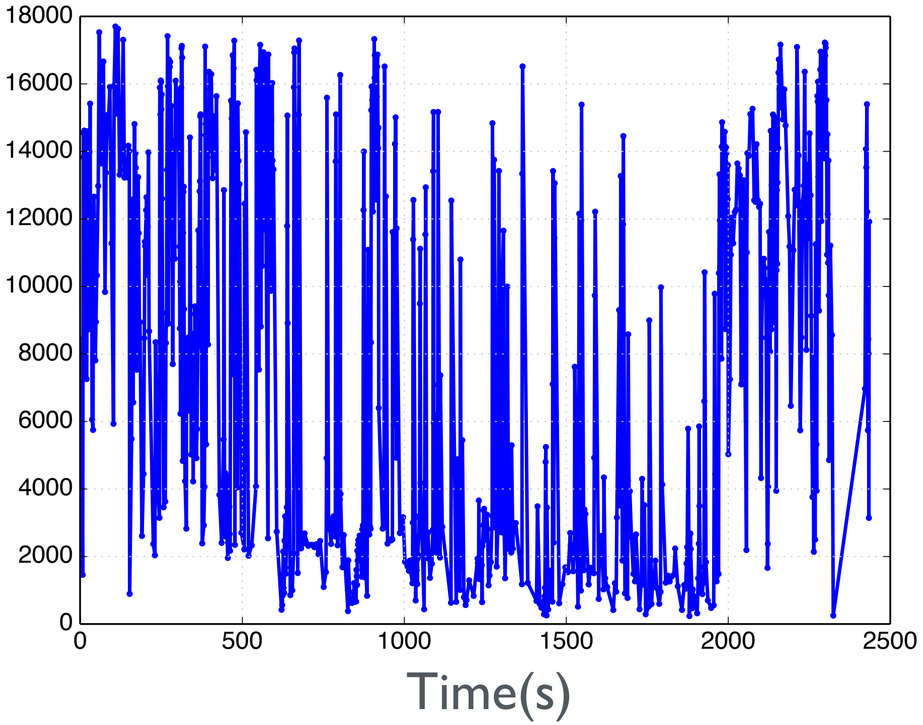
Avg. throu
chunk dow

"20–30% of rebuffers are unnecessary"

Time(s)

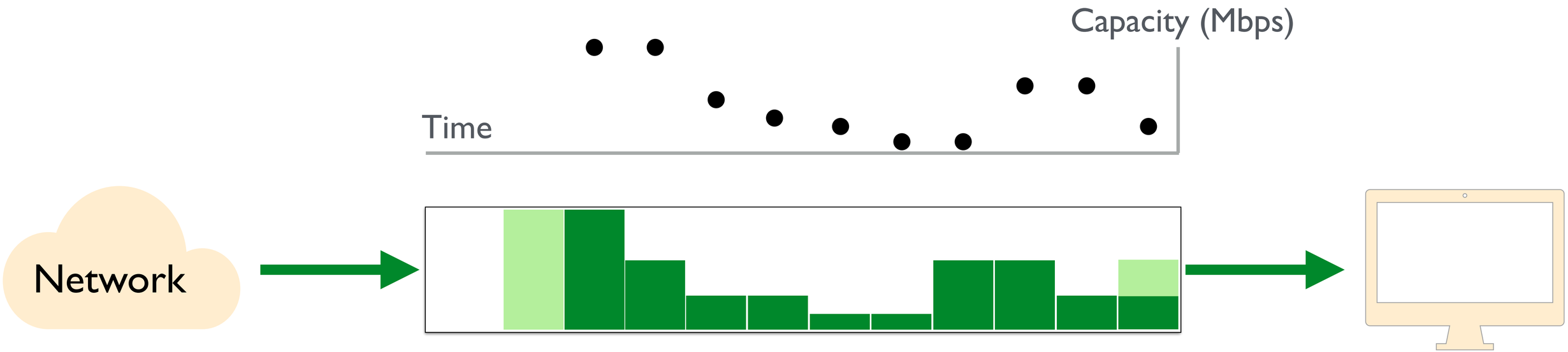
Estimating available capacity

Avg. throughput over chunk download (kbps)



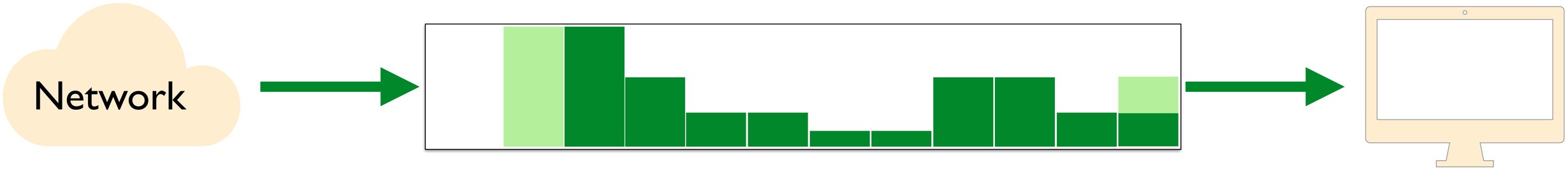
[A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service, Huang et al., ACM SIGCOMM 2014]

Capacity estimation



Decide based on the buffer alone?

Buffer-based adaptation

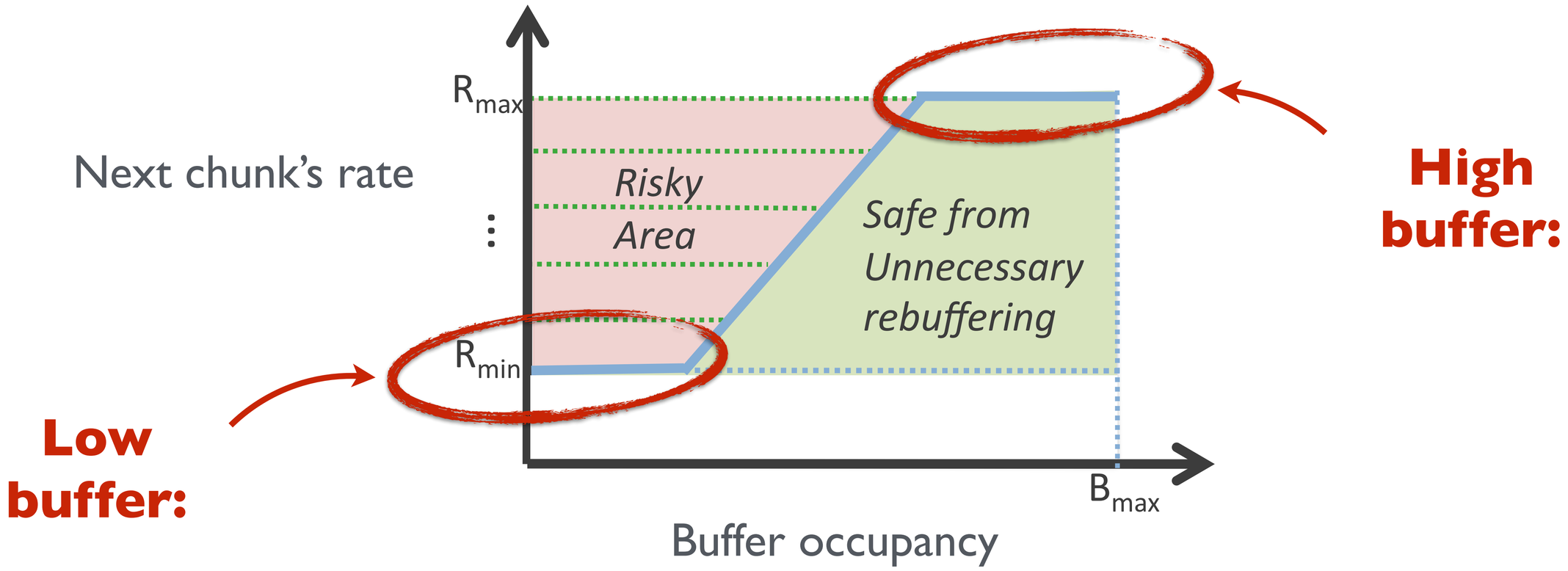


Nearly full buffer ⇒ large rate

Buffer-based adaptation

Nearly empty buffer ⇒ small rate

Buffer-based adaptation



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

Problem: startup phase?

Pick a rate based on immediate past throughput

Communication Networks

Spring 2020

Laurent Vanbever

nsg.ee.ethz.ch

ETH Zürich (D-ITET)

May 11 2020