クラウドコンピューティング

基礎論

第10回

創造情報•小林克志

ikob@acm.org

Course Outline

- Administrivia
- Cloud computing
- Service reliability
- Scale-up / Scale-out
- Distributed data stores
- Global services
- Datacenter networkings (1)
- Datacenter networkings (2)
- Network performance
- User experiences
- Network latencies
- Advanced topics

Outline

- Administravia
- Homework review
- User eXperience on Web service (cont'd)
  - IW10
  - HTTP
- NS2 simulation

Booting NS2 simulator with Docker

- Install Docker platform from docker.com

- Prepare X11 server application, and then allow Xserver accept remote access.

  MacOS: Start XQuartz, click “XQuartz -> Preference”, choose “Security” tab, check “Allow remote access” box, and then restart XQuartz. Also, allow remote access with the command as:

```
$ xhost +
```

  Windows: Start MobaXterm, click “Settings” menu, choose “X11” tab, and then set “X11 remote access” to “full”

- Find and set the IP address of Docker plathome by using:

  `ifconfig` for MacOS
  `ipconfig` for Windows

- Create a directory to share between the container and server. Then copy sample script as:

```
$ mkdir ~/ex-1
$ cp <Download file> ~/ex-1/
```

- Start docker container as:

```
$ docker run -it -v ~/ex-1:/opt \
         -e DISPLAY=<IP Address>:0 ekiourk/ns2 bash
root@e2e1dbbc0950:/ns2/ns-2.35#
```

- Verify shared directory contents as:

```
root@e2e1dbbc0950:/ns2/ns-2.35# ls /opt
```

For hands-on exercise : Install two softwares

1.Wireshark : A packet capture and analyzer

<- Sorry Canceled.

- Just install package from http://www.wireshark.org

2.NS2 : Network simulator.

Three options are there:

A. Docker
- Install docker software and container:
  - https://github.com/ekiourk/docker-ns2
  - X-window server is required. It depends on OS.

B. Native applicationIf you are using Linux, use this option.
- Install ns-allinone-2.35 from source because NS-2 package.
  Note that some distribution may not work.
- X-window, perl, gnuplot are also required.

C. Virtual Machine (VM)
- Install Hypervisor Software.
  - Oracle VirtualBoX is free.
    vmware or others are also welcome.
- Linux VM image with NS2 software will be available from the course Web.

演習に向けて：２つのソフトウェアをインストールする

1.パケットアナライザ wireshark

<- Sorry Canceled.

・http://www.wireshark.org を参考にインストールすること。

2.ネットワークシミュレータ NS2

A. Docker

・Docker をインストールし、コンテナを使用する。

- https://github.com/ekiourk/docker-ns2
- X-window の設定は OS に依存する。

B. Native

・Linux を利用している場合は、この方法を使う。

- ns-allinone-2.35 を install すること。
  - X-window, perl, gnuplot なども必要となる。

C. VM で動作

- ハイパーバイザの導入
  - Oracle VirtualBoX であれば無償
    vmware 他でもかまわない
- NS2 付きの Linux 仮想マシンイメージを講義ページで配布する

Final report

- Choose one network latency topic surveyed in:
  Briscoe, Bob, et al. "Reducing Internet Latency: a survey of techniques and their merits." IEEE Communications Surveys & Tutorials.

- Write review of **all of the cited papers** in the chosen topic, but excluding [202]
  The review must include both strong and weak points of the papers.

- Submit via course Web.

  - Submission due date is July 31.

  - The review must be 2-4 pages long and written either in English or Japanese.

最終レポート

- 以下のサーベイ論文からネットワーク遅延に関するトピックを一つ選択せよ:

  Briscoe, Bob, et al. "Reducing Internet Latency: a survey of techniques and their merits." IEEE Communications Surveys & Tutorials.

- 選択したトピックで引用されている**全ての論文**をレビューすること。
  レビューには論文の良い点、悪い点両方を含むこと。

- 講義 Web ページから提出すること。

  - 締め切りは 7 月 31 日（金）

  - レビューは 2-4ページにまとめ、英語あるいは日本語で記述すること。

Today’s Assignment

- University people are frustrated with the academic affair system, such as UTAS, due to poor usability. A smartphone application named Oraio is released, which improves the usability of the official service. Oraio requires an user to register own ID/ password. Then, Oraio retrieves sign-upped classes, and builds own schedule and attendance records automatically.
  Some universities officially discouraged their students using such application because of the security concern.

- Whether should the university administrator of information technology allow or disallow such application ?

  - If your student ID is even: Discuss the above from the viewpoint of CSO.

  - Otherwise: Discuss the above from the viewpoint of CIO.

- Submit your answers in Japanese or in English via the course web.

  Note: CIO: Chief Information Officer / CSO : Chief Security Officer

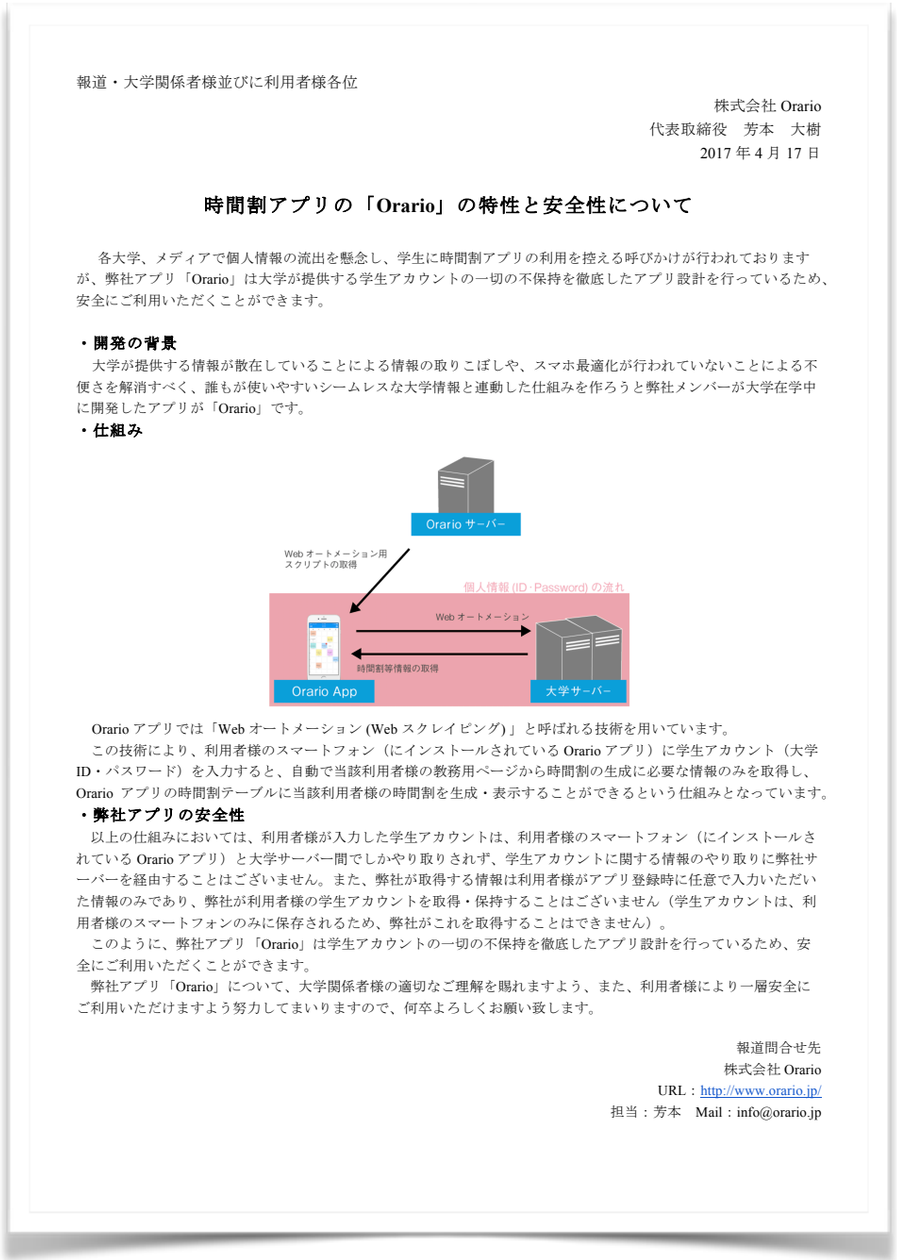
本日の課題

- 大学の学務システムについて使い勝手などで不満は多い。学務システムを向上させるスマートホンアプリケーションとして Oraio が登場した。Oraio はマッシュアップアプリケーションの一種で学生の ID/Password を利用して学務情報を取得、スケジュール、出席履歴などが生成される。
  いくつかの大学はこのようなアプリケーションの利用に対してセキュリテイの懸念などから注意喚起をおこなった。
- 大学の情報技術責任者としてこのようなアプリケーションを許容すべきか禁止すべきか？
  - 学籍番号が偶数： CSO の視点で議論せよ
  - それ以外：CIO の視点で議論せよ
- 講義 Web ページから回答すること。

  CIO: Chief Information Officer / CSO : Chief Security Officer

Oraio application

- According to the public statement, the application store ID/password on owner's SmartPhone only, but not on any network services including Oraio.
- What is different from 3'd party web browsers, such as Chrome, Firefox.
- Many OS provides ID/Passoword management frameworks which backup customer's ID/Password on their storages via the Internet.



報道・大学関係者様並びに利用者様各位

株式会社 Orario
代表取締役　芳本　大樹
2017 年 4 月 17 日

時間割アプリの「Orario」の特性と安全性について

各大学、メディアで個人情報の流出を懸念し、学生に時間割アプリの利用を控える呼びかけが行われておりますが、弊社アプリ「Orario」は大学が提供する学生アカウントの一切の不保持を徹底したアプリ設計を行っているため、安全にご利用いただくことができます。

・開発の背景

大学が提供する情報が散在していることによる情報の取りこぼしや、スマホ最適化が行われていないことによる不便さを解消すべく、誰もが使いやすいシームレスな大学情報と連動した仕組みを作ろうと弊社メンバーが大学在学中に開発したアプリが「Orario」です。

・仕組み

Orario アプリでは「Web オートメーション (Web スクレイピング) 」と呼ばれる技術を用いています。

この技術により、利用者様のスマートフォン（にインストールされている Orario アプリ）に学生アカウント（大学 ID・パスワード）を入力すると、自動で当該利用者様の教務用ページから時間割の生成に必要な情報のみを取得し、Orario アプリの時間割テーブルに当該利用者様の時間割を生成・表示することができるという仕組みとなっています。

・弊社アプリの安全性

以上の仕組みにおいては、利用者様が入力した学生アカウントは、利用者様のスマートフォン（にインストールされている Orario アプリ）と大学サーバー間でしかやり取りされず、学生アカウントに関する情報のやり取りに弊社サーバーを経由することはございません。また、弊社が取得する情報は利用者様がアプリ登録時に任意で入力いただいた情報のみであり、弊社が利用者様の学生アカウントを取得・保持することはございません（学生アカウントは、利用者様のスマートフォンのみに保存されるため、弊社がこれを取得することはできません）。

このように、弊社アプリ「Orario」は学生アカウントの一切の不保持を徹底したアプリ設計を行っているため、安全にご利用いただくことができます。

弊社アプリ「Orario」について、大学関係者様の適切なご理解を賜りますよう、また、利用者様により一層安全にご利用いただけますよう努力してまいりますので、何卒よろしくお願い致します。

報道問合せ先
株式会社 Orario
URL：http://www.orario.jp/
担当：芳本　Mail：info@orario.jp

Mission of CSO / CIO

- Chief Security Officer (CSO)
    - Mitigation and/or reduction of compliance, operational, strategic, financial and reputational security risk strategies relating to the protection of people, intellectual assets and tangible property.
- Chief Information Officer (CIO)
    - Form a key part of any business that utilizes technology and data.
    - Manage IT resources and plan "ICT including policy and practice development, planning, budgeting, resourcing and training”.
    - Becoming increasingly important in calculating how to increase profits via the use of ICT frameworks, as well as the vital role of reducing expenditure and limiting damage by setting up controls and planning for possible disasters.

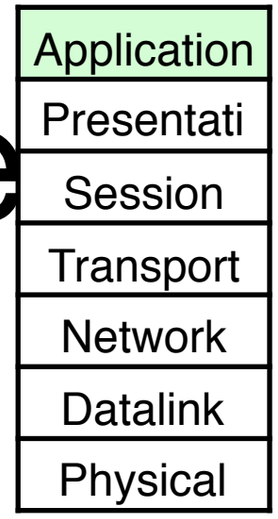
Outline

- Administravia
- Homework review
- User eXperience on Web service (cont'd)
  - IW10
  - HTTP
- Advanced Topic
  - QUIC
- NS2 simulation

Latency in network applications

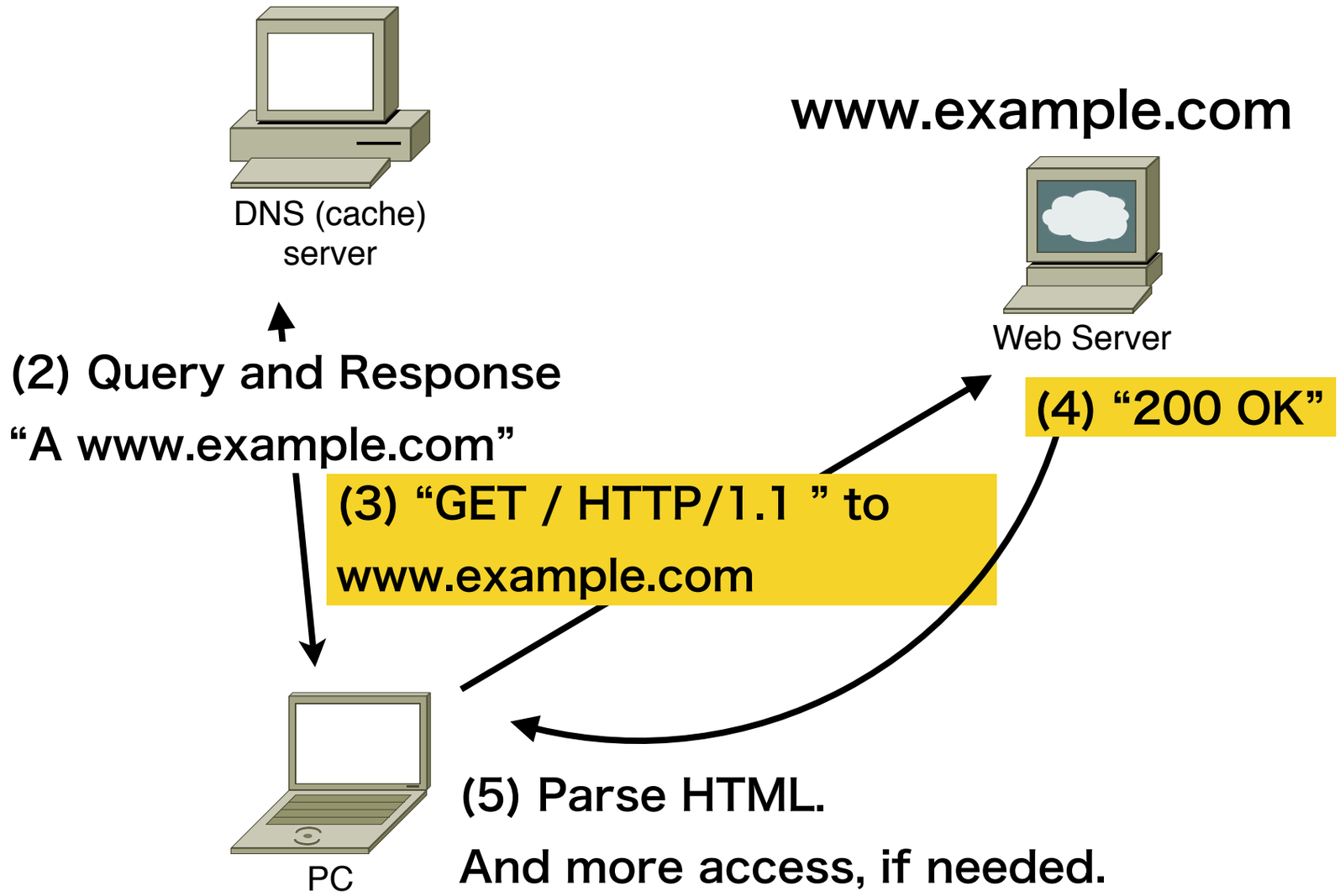
- Legacy:
    - Processing, Propagation, Queueing, De-jitter, Retransmission, Flow-control,
- Today:
    - Entire application process from the perspective of entire UX
        - Smooth interactive UI with Ajax

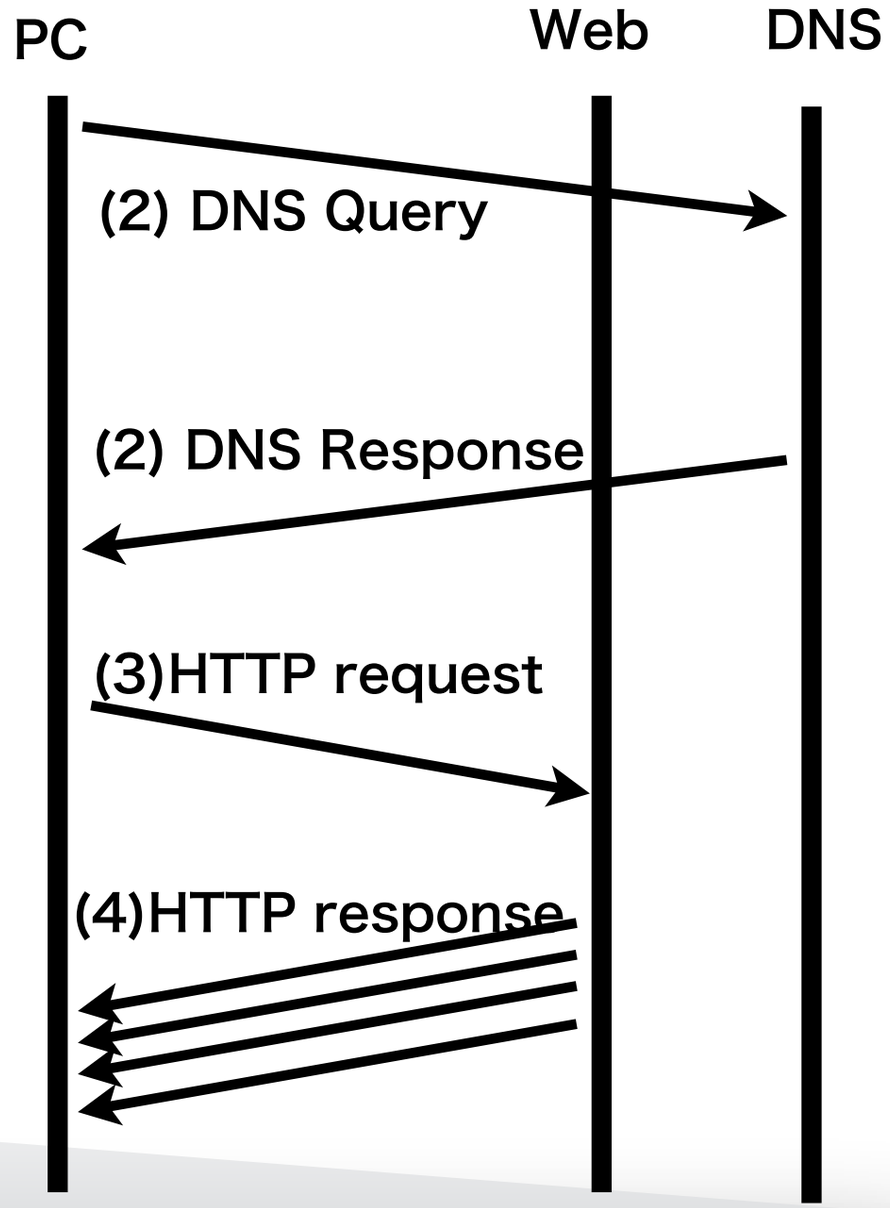
Latency of network service caused by …

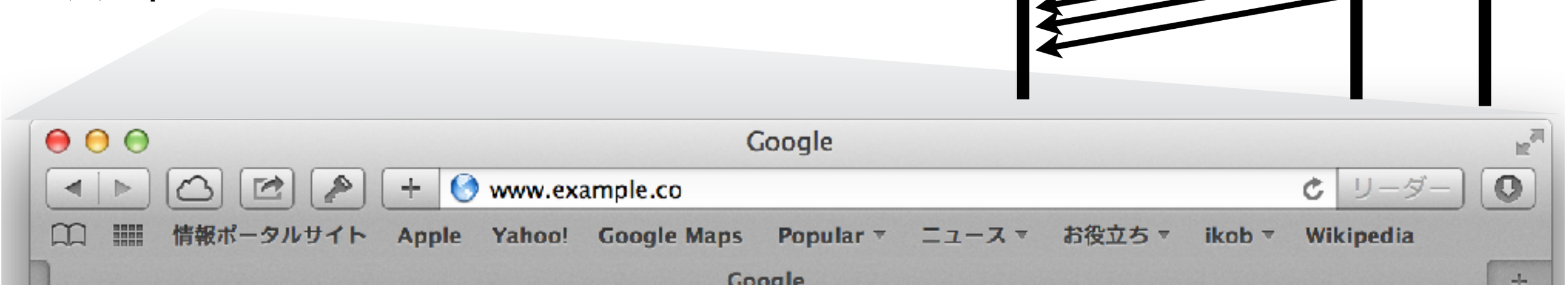


- Intra-node (server, client) :
  - Can be reduced by improving node performance and/or software, but not discussed in this class.
- Client - Server communication :
  - Global network, or the Internet
- Inter-node in server cluster :
  - Depending on communication in DC network

How client accesses Web service ?

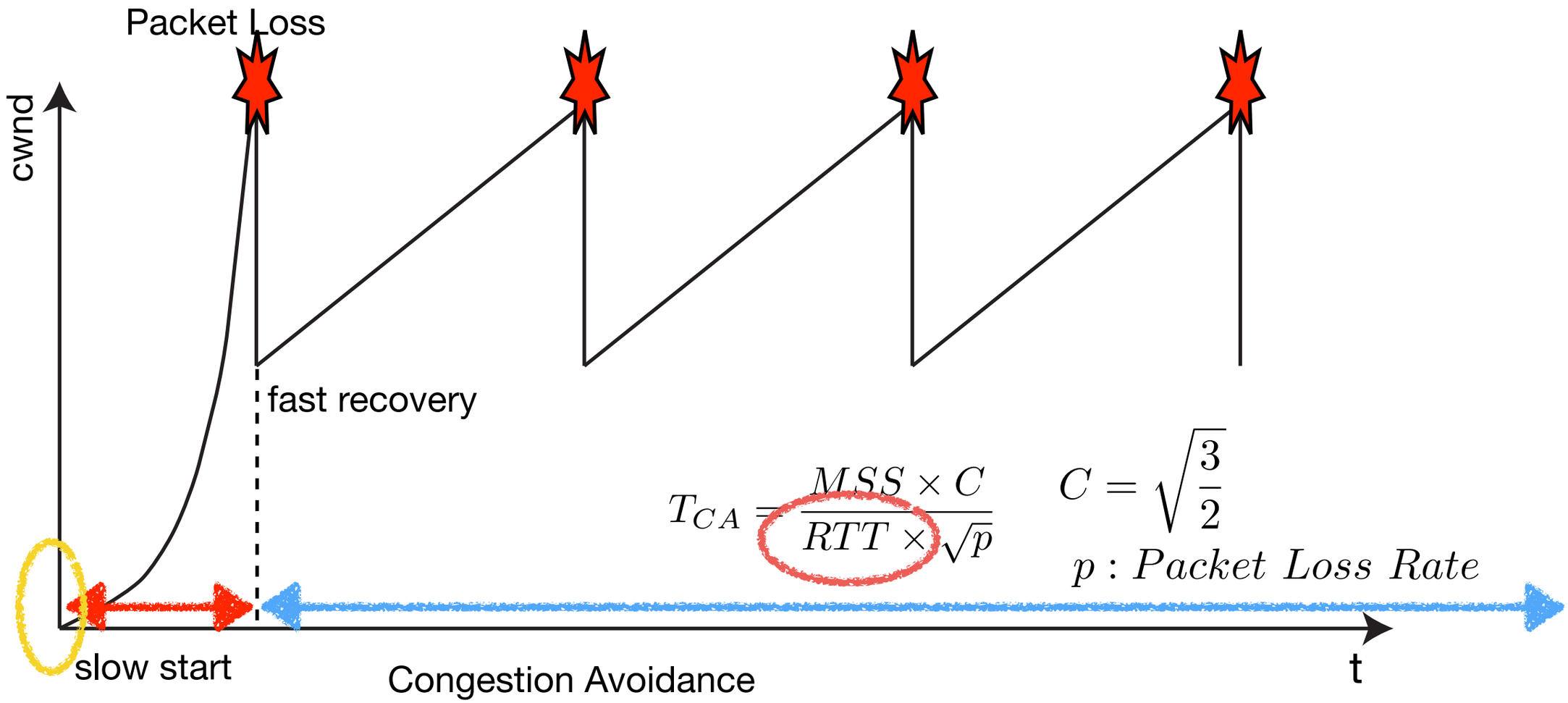






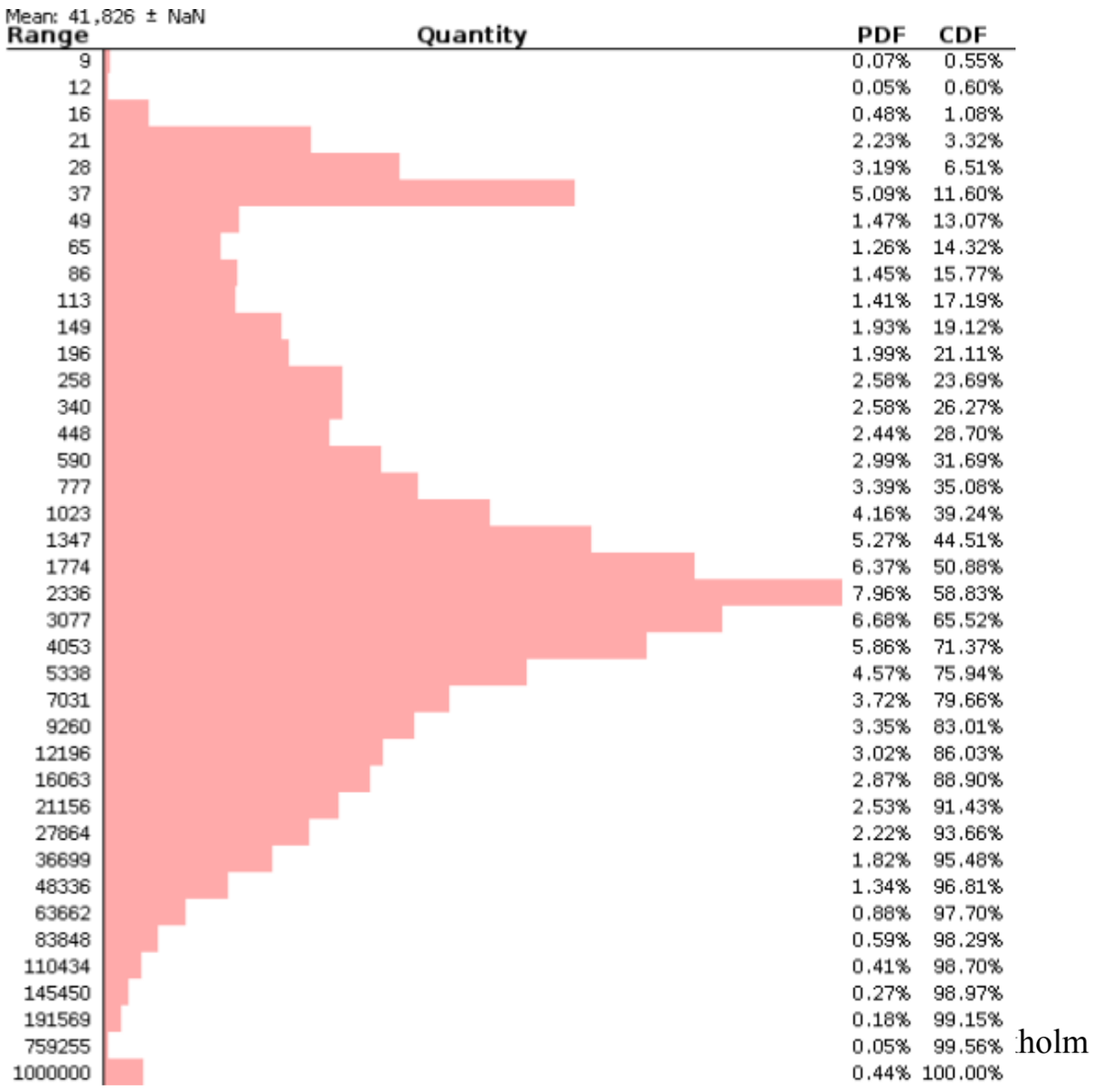
TCP window behavior on NewReno

Application
Presentatio
Session
Transport
Network
Datalink
Physical



Mathis, Matthew, et al. "The macroscopic behavior of the TCP congestion avoidance algorithm." *ACM SIGCOMM Computer Communication Review* 27.3 (1997): 67-82.

HTTP Response Size Distribution



Range	Quantity	PDF	CDF
9		0.07%	0.55%
12		0.05%	0.60%
16		0.48%	1.08%
21		2.23%	3.32%
28		3.19%	6.51%
37		5.09%	11.60%
49		1.47%	13.07%
65		1.26%	14.32%
86		1.45%	15.77%
113		1.41%	17.19%
149		1.93%	19.12%
196		1.99%	21.11%
258		2.58%	23.69%
340		2.58%	26.27%
448		2.44%	28.70%
590		2.99%	31.69%
777		3.39%	35.08%
1023		4.16%	39.24%
1347		5.27%	44.51%
1774		6.37%	50.88%
2336		7.96%	58.83%
3077		6.68%	65.52%
4053		5.86%	71.37%
5338		4.57%	75.94%
7031		3.72%	79.66%
9260		3.35%	83.01%
12196		3.02%	86.03%
16063		2.87%	88.90%
21156		2.53%	91.43%
27864		2.22%	93.66%
36699		1.82%	95.48%
48336		1.34%	96.81%
63662		0.88%	97.70%
83848		0.59%	98.29%
110434		0.41%	98.70%
145450		0.27%	98.97%
191569		0.18%	99.15%
759255		0.05%	99.56%
1000000		0.44%	100.00%

holm

Data collected from Google Chrome (rough estimate with caveat!):

Median: ~2KB

Mean: ~41KB

99th percentile mean: 8.1KB due to heavy tail (0.5% in the 1MB + bucket)

67.5% < 3*mss (4380)

73% < 4*mss

77% < 5*mss

H.K. Jerry Chu, “Tuning TCP Parameters for the 21st Century”, 75th IETF, Stockholm, 2009

IW10 proposal

- Increase initial congestion window size from 3 to 10
  - improve UX on interactive Web to reduce latency.
- Concern from developing countries: ”Since we don’t have enough up-link, we are afraid congestion collapse caused by IW10.

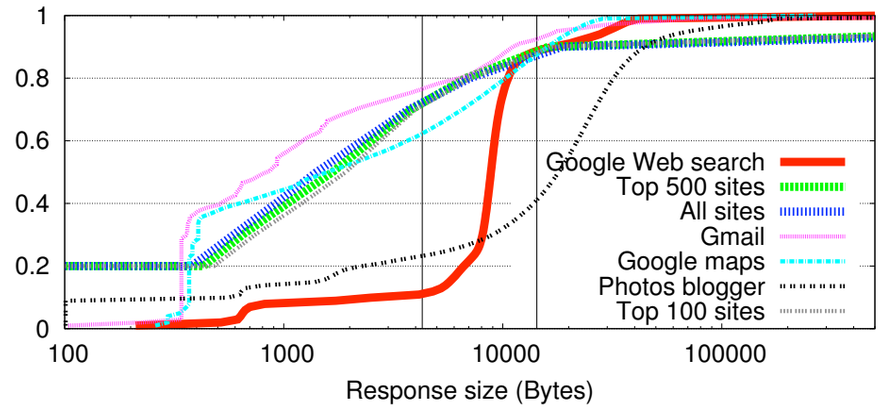


Figure 1: CDF of HTTP response sizes for top 100 sites, top 500 sites, all the Web, and for a few popular Google services. Vertical lines highlight response sizes of 3 and 10 segments.

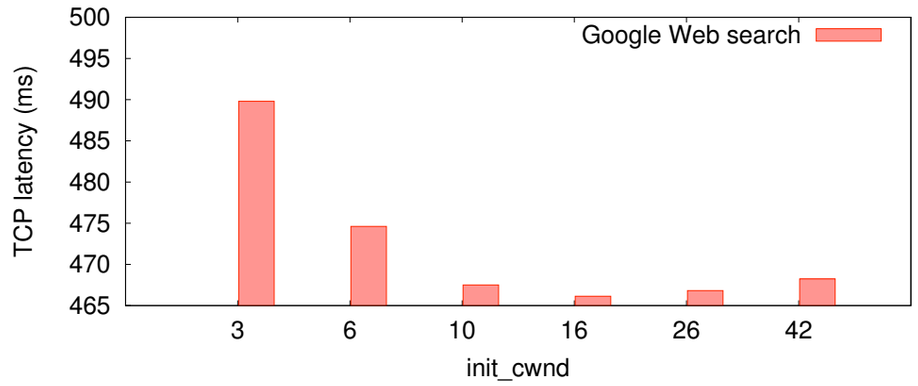
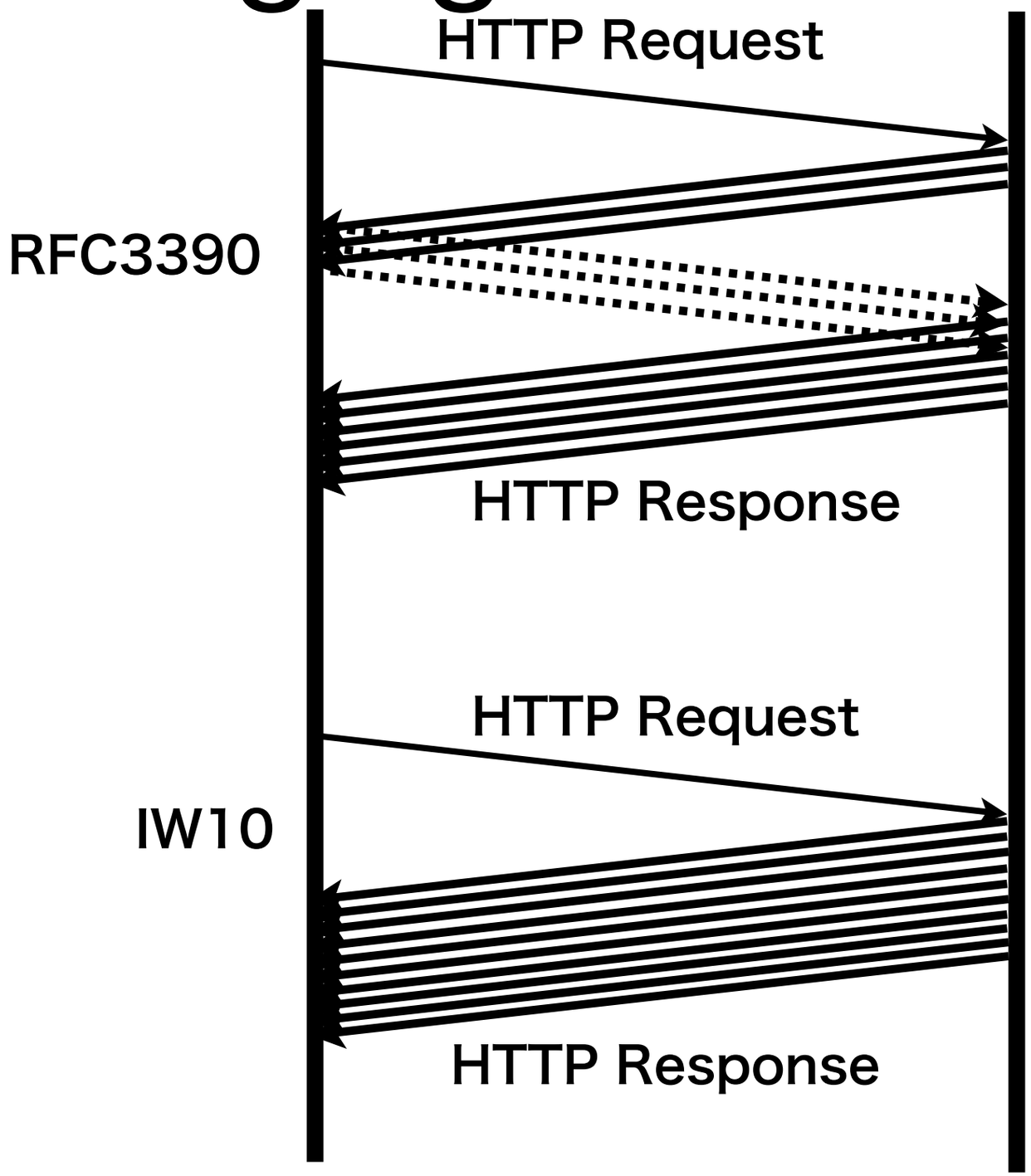


Figure 2: TCP latency for Google search with different *init_cwnd* values.

Allman, Mark, Sally Floyd, and Craig Partridge. "RFC 3390: Increasing TCP’s initial window." *Internet Eng. Task Force (IETF)-Request for Comments* (2002).

Dukkipati, Nandita, et al. "An argument for increasing TCP's initial congestion window." *Computer Communication Review* 40.3 (2010): 26-33.

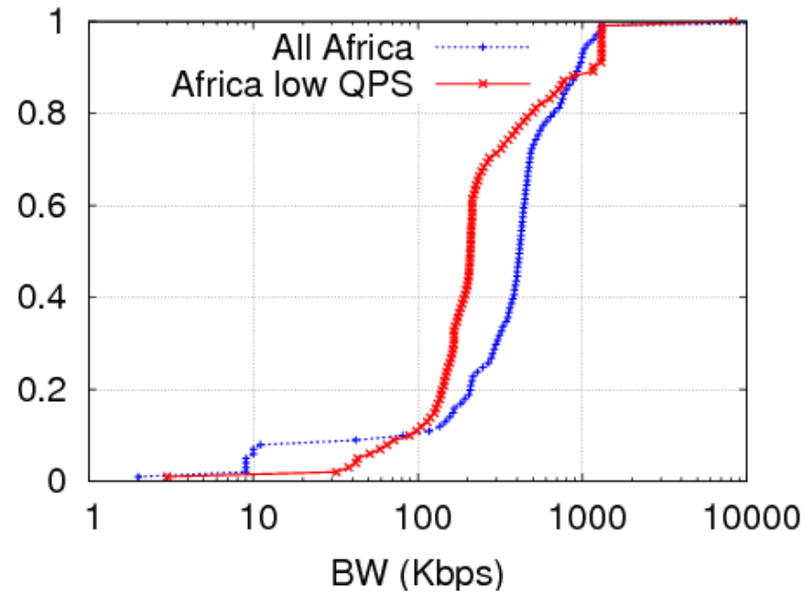
Reduce latency with enlarging Initial Window

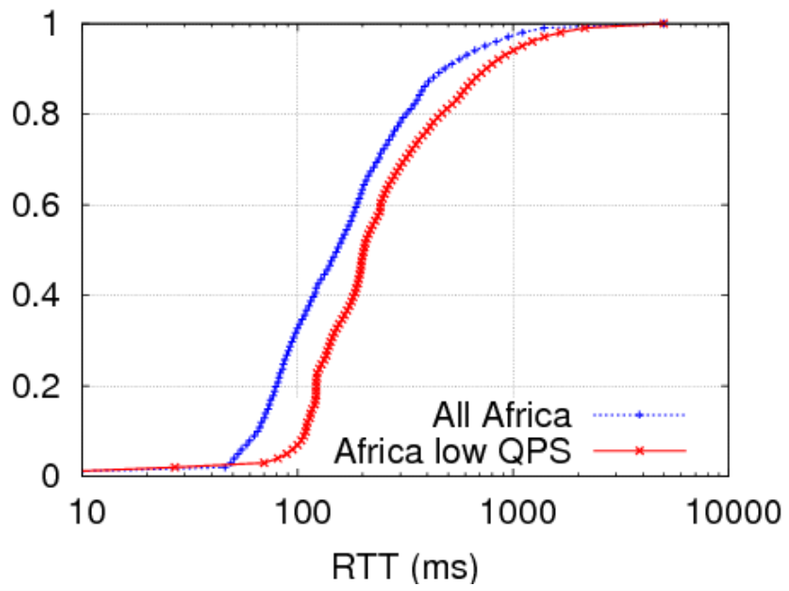


IW10 proposal (contd)

Analysis of IW10 on Africa traffic

Experiment for 1 week in June 2010





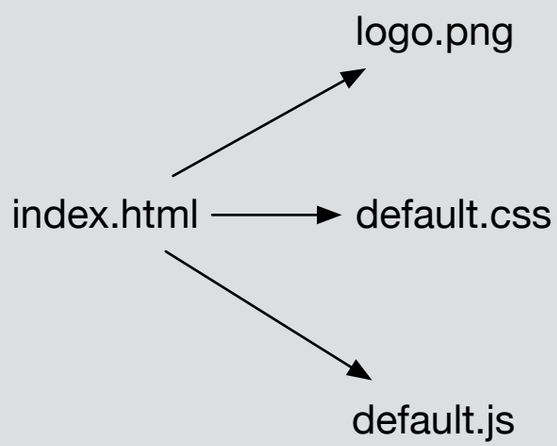
Outline

- Administravia
- Homework review
- User eXperience on Web service (cont'd)
  - IW10
  - HTTP
- Advanced Topic
  - QUIC
- NS2 simulation

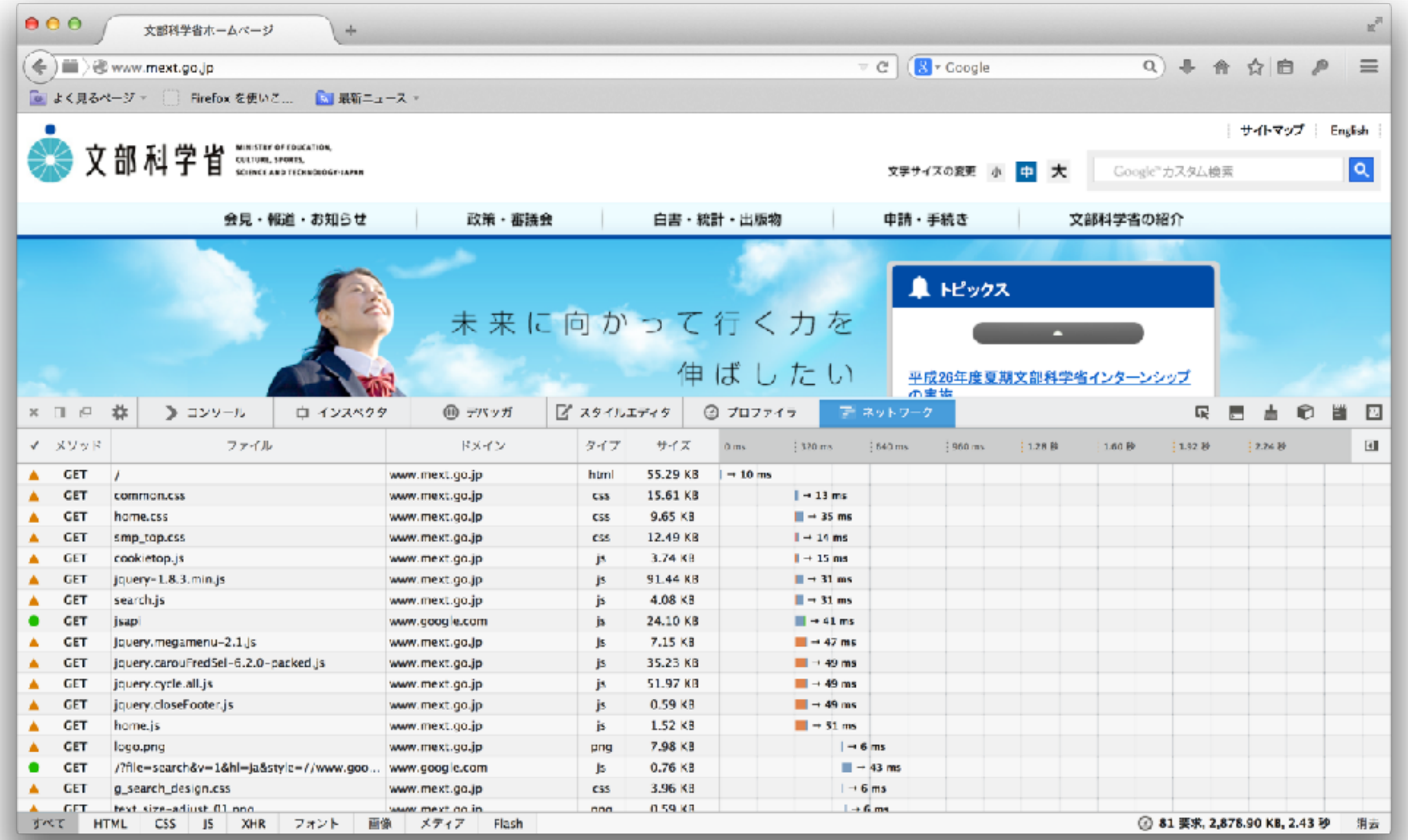
HTML

- With HTML constructs, images and other objects, such as interactive forms, may be embedded into the rendered page. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. (*Wikipedia*)

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="default.css">
<script type="text/javascript" src="default.js"></script>
</head>
<body>
<h1> A sample page </h1>
<p>
<img src="logo.png" alt="logo">
</p>
</body>
</html>
```



Timelines on Web inspector



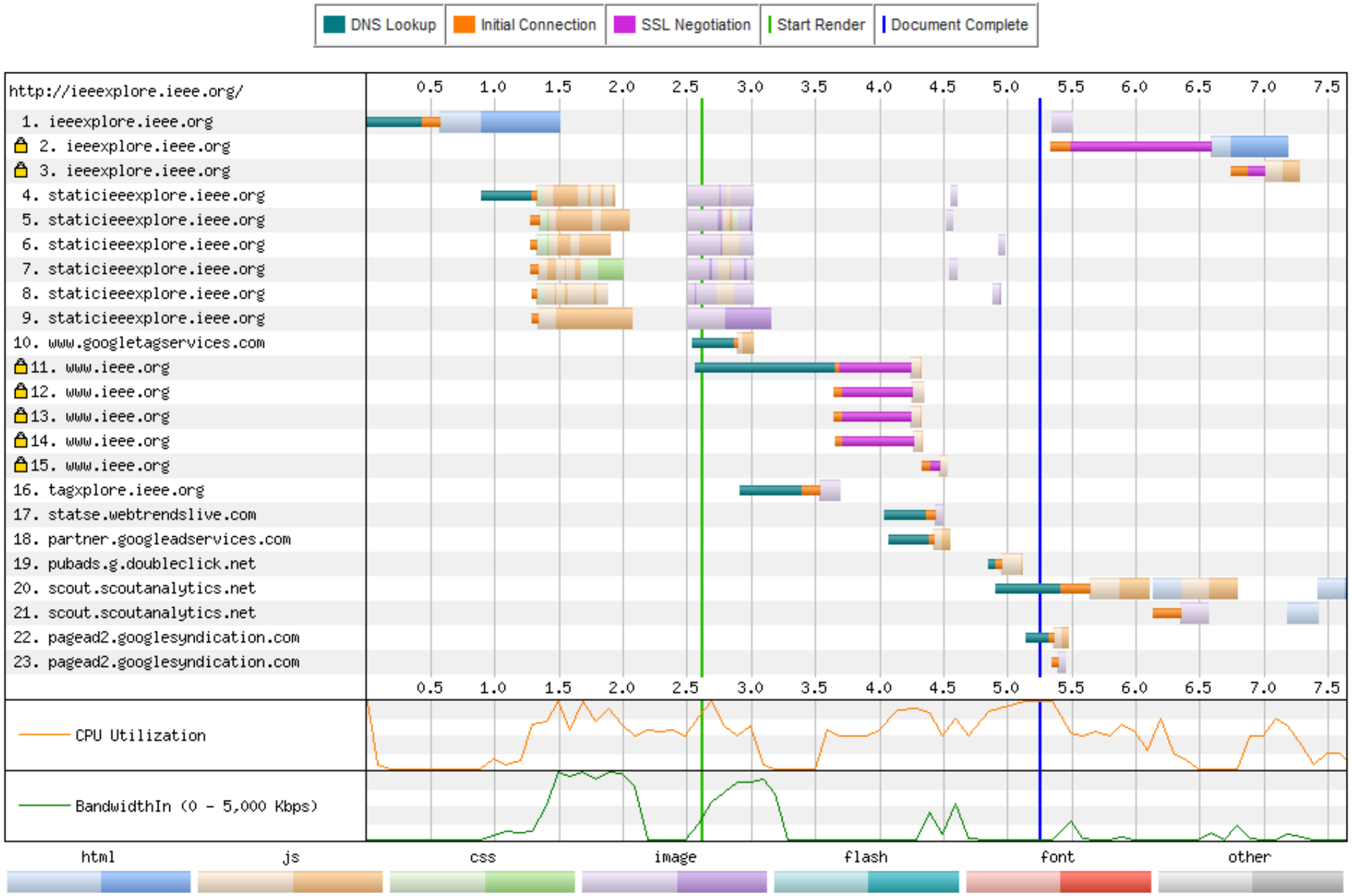
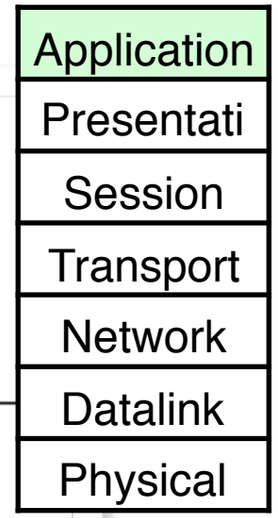


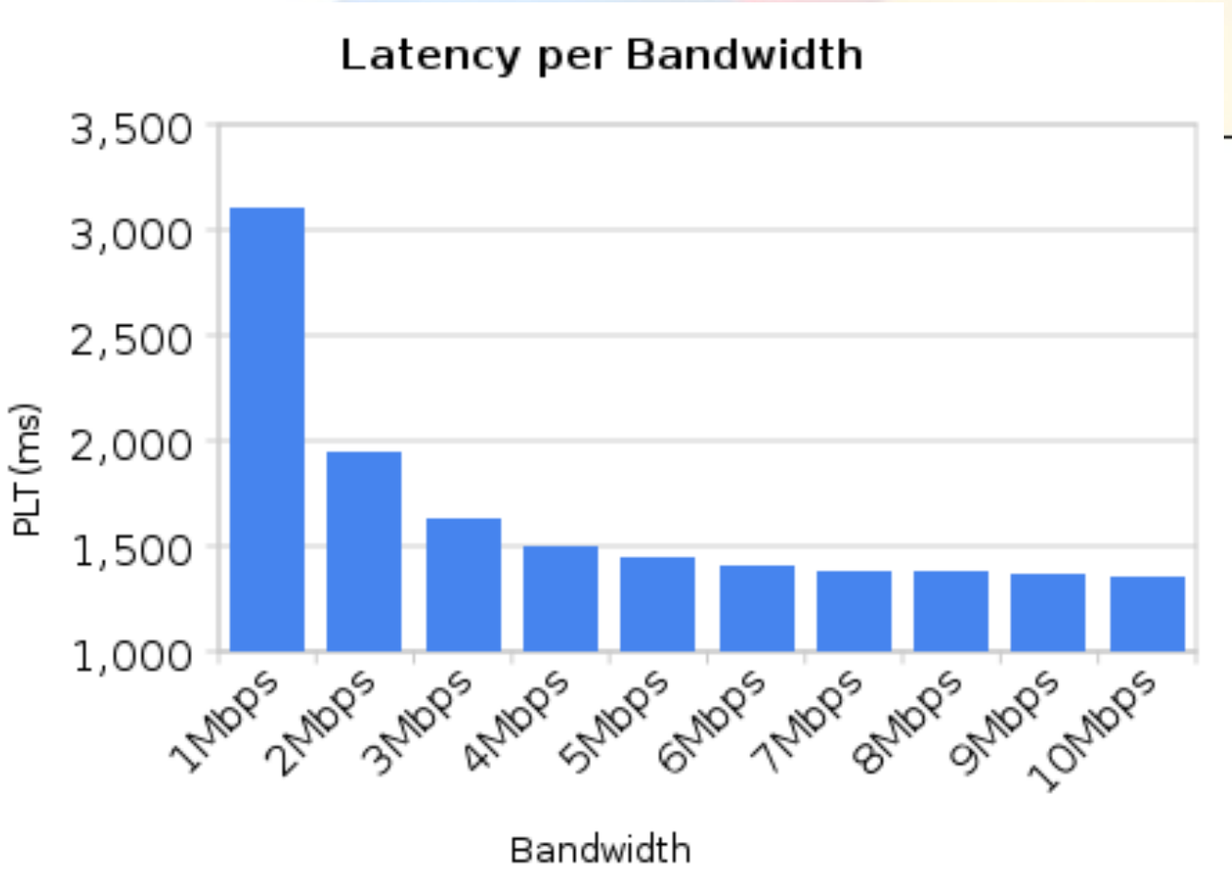
Fig. 1. Waterfall diagram showing the timing of download of an apparently uncluttered example Web page (ieeexplore.ieee.org), actually comprising over one hundred objects, transferred over 23 connections needing 10 different DNS look-ups. The horizontal scale is in seconds. This access was from Stockholm, Sweden, over a 28ms RTT 5 Mb/s down 1 Mb/s up cable access link, using Internet Explorer v8 without any prior cache warming. Source: www.webpagetest.org

Lesson #3: It's not the bandwidth



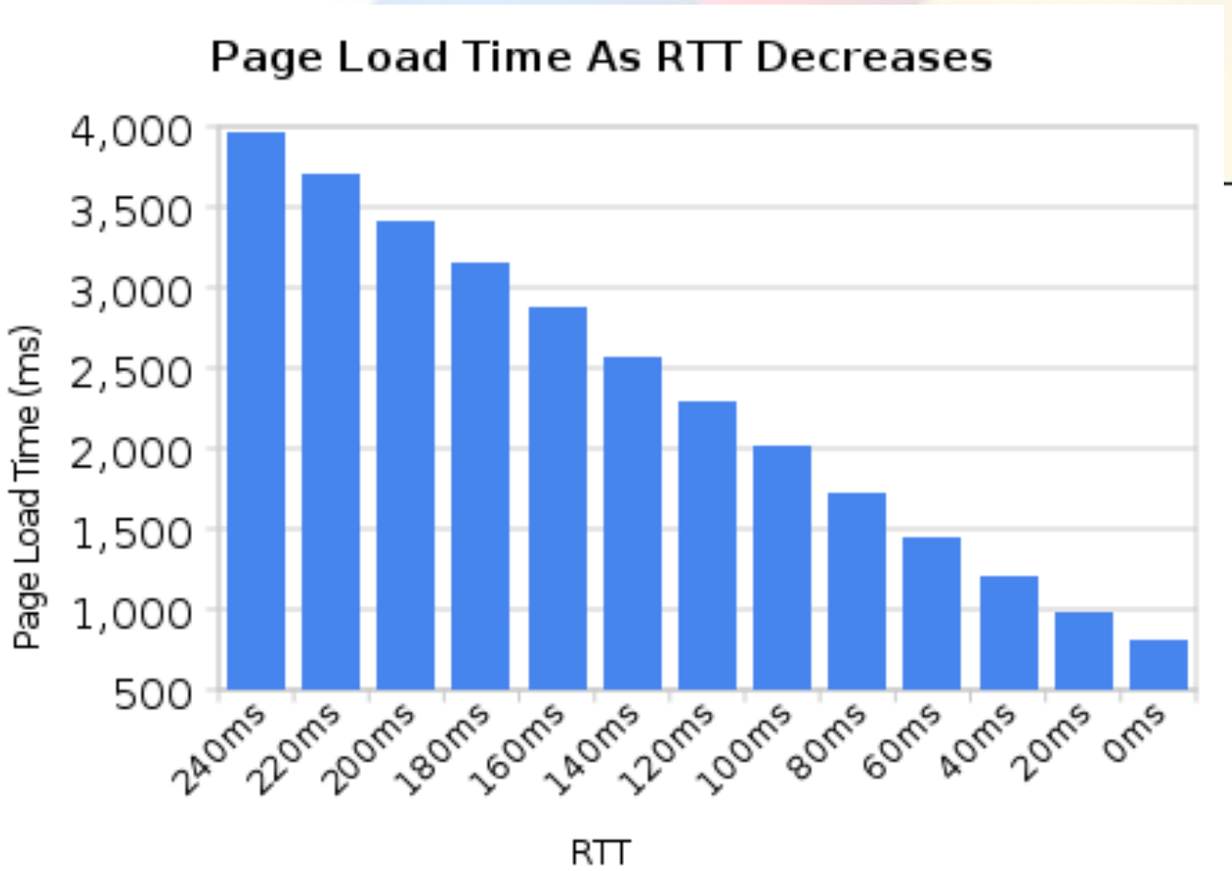
- Average bandwidth in the US is 3.9Mbps
  - Rapidly increasing.
  - 1Gbps projects are underway.
  - See also: http://www.akamai.com/stateoftheinternet/
- Average RTT to Google is 114ms
  - Not decreasing quickly
  - Mobile makes this worse
- Let's see an example:
  - Fix RTT at 60ms, 0% packet loss, vary bandwidth
  - Fix bandwidth at 5Mbps, 0% loss, vary RTT

Application
Presentati
Session
Transport
Network
Datalink
Physical



Latency (page load time) decreases as the bandwidth increases. Note the diminishing returns.

Application
Presentati
Session
Transport
Network
Datalink
Physical



Now we vary the RTT for a fixed bandwidth.

Reducing RTT, always helps reduce PLT.

Google™

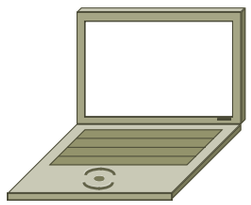
RTT Barrier

Application
Presentati
Session
Transport
Network
Datalink
Physical

www.example.com

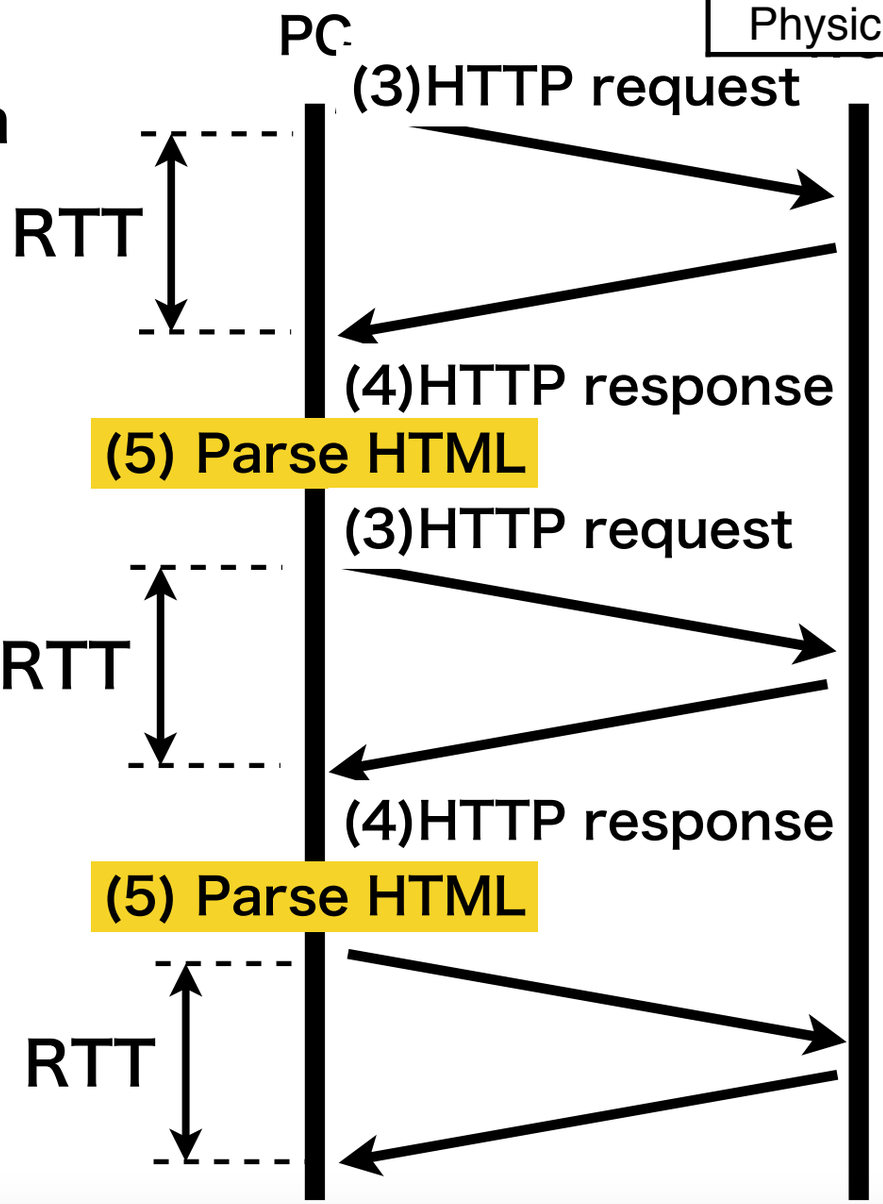
(3) “GET / HTTP/1.1 ” to
www.example.com

(4) “200 OK”



(5) Parse HTML.
And more access, if needed.

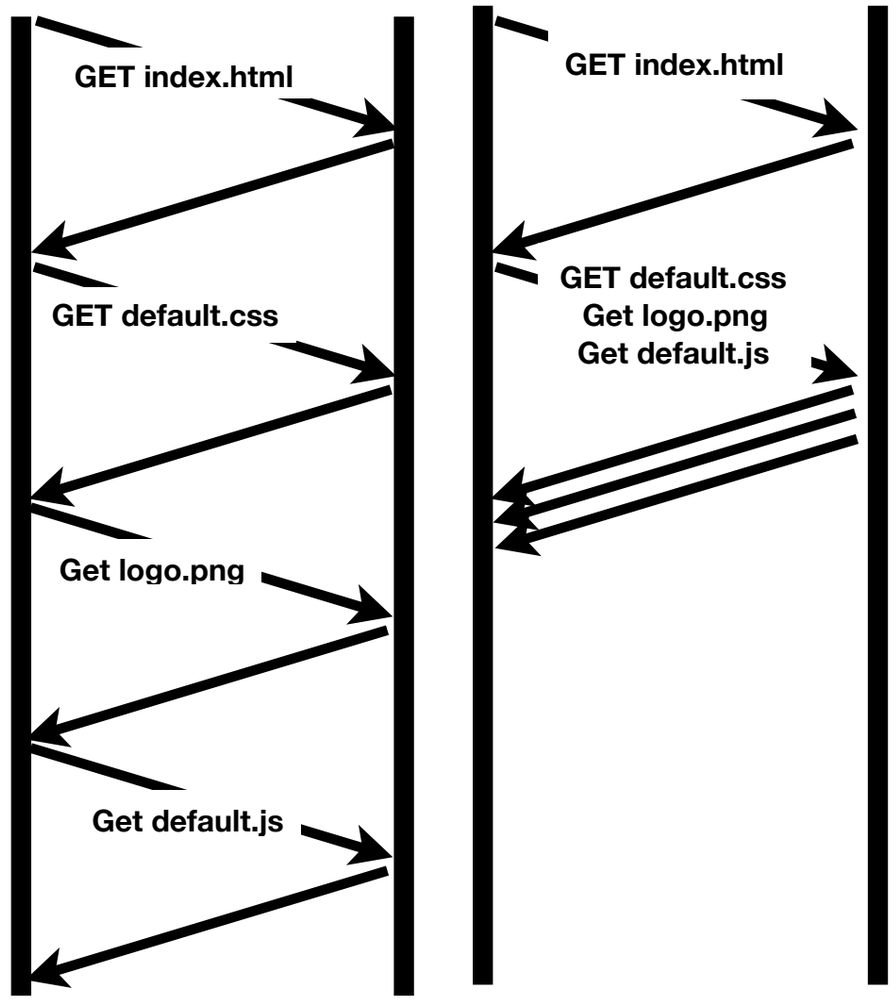
(1) Input URL or click anchor



HTTP pipelining

Application
Presentati
Session
Transport
Network
Datalink
Physical

- More than one requests on a single http connection.
- But not deployed, because
  - Contents/Services provided by more than one servers.
  - Large contents, such as image, occupy connection pipe and block more significant ones. (Head of Line Blocking (HoL)).

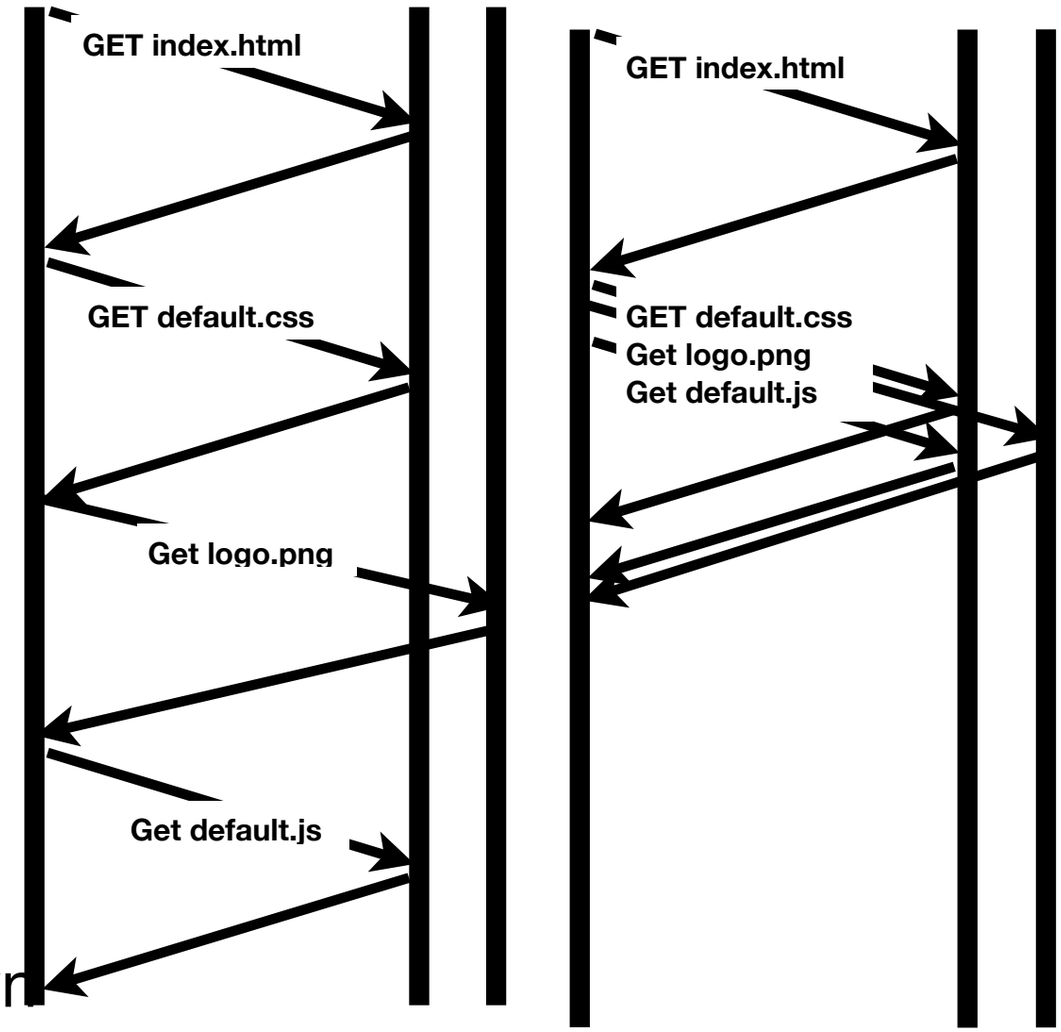


HTTP multiplex

Parallel HTTP connection

Application
Presentati
Session
Transport
Network
Datalink
Physical

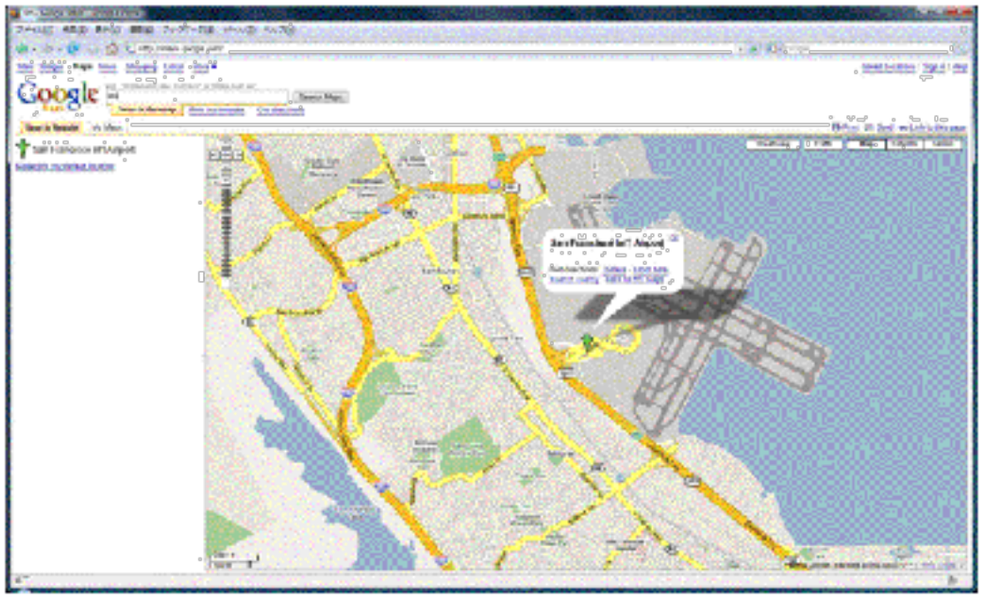
- Multiple HTTP connections
  - Most browsers use up to 6 conns.
- Pros:
  - Able to improve performance with different sites.
- Cons:
  - Consume flow tables on NAT or IPS/IDS
  - ISPs provide NAT service to save own IPv4 address spaces.



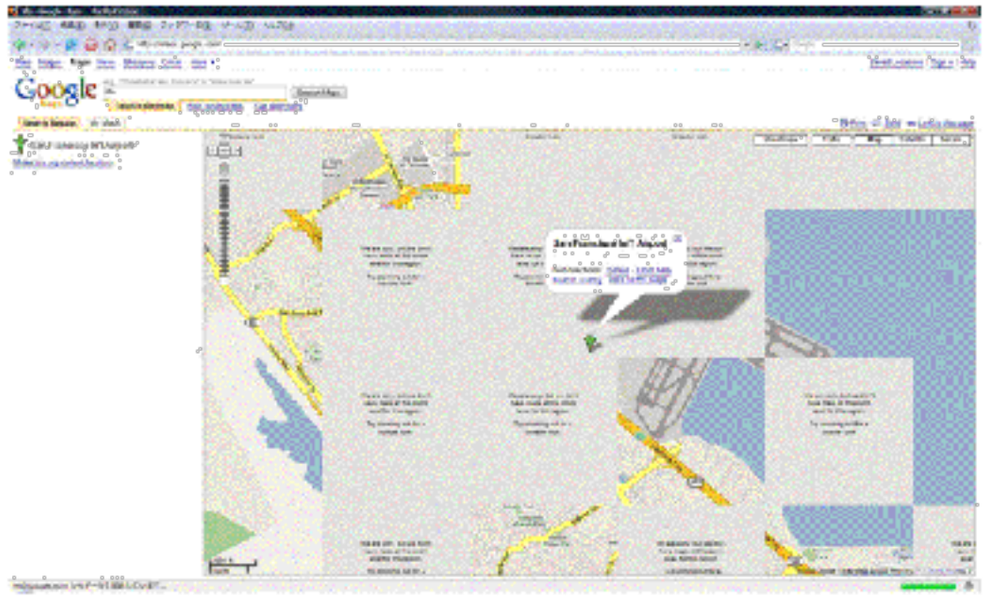
Concurrent

Google Maps with Parallel HTTP connections

Max 30 Connections



Max 15 Connections



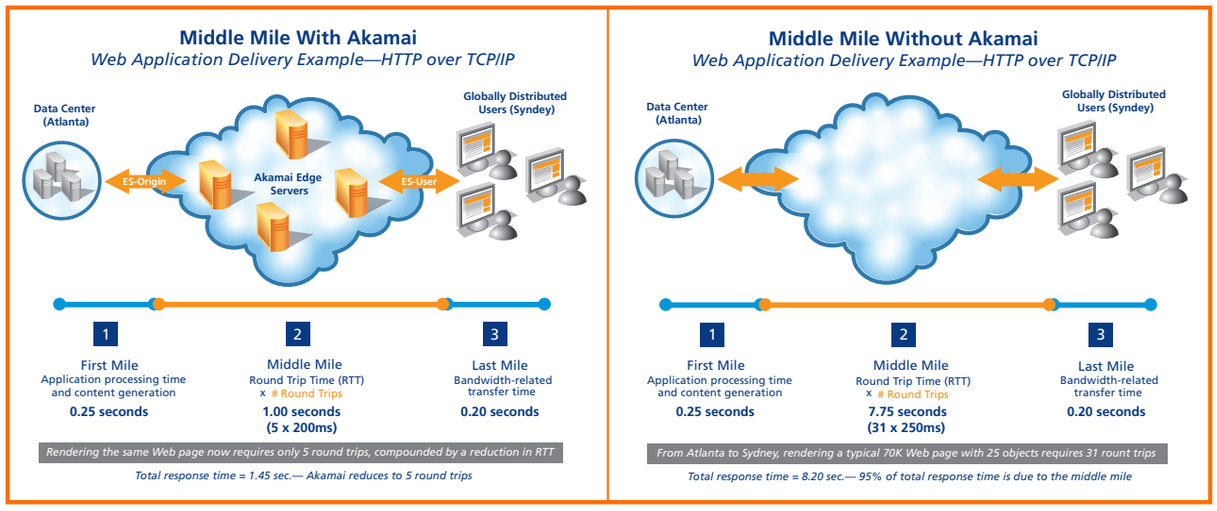
Shin Miyakawa, “From IPv4 only to v4/v6 Dual Stack”, IETF72

Contents Distribution Network (CDN) and RTT

- CDN is developed to serve content to Internet users with optimized availability and performance. A CDN uses servers that are geographically distributed, helping to accelerate the delivery of content by caching it in multiple locations and then using the closest server to fulfill a request for content from each particular user.
  Many CDN providers compete Akamai, Cloudflare, AWS CloudFront compete with each other.

  - CDN offers “Best” server using metrics such as:
    - Small RTT / Bandwidth Capacity / Stable connectivity each other.

  - Akamai deploys more than ~~100,000~~ 200,000 edge servers in order to improve UX incl. to reduce RTT.



Akamai Corp., “Akamai’s EdgePlatform for Application Acceleration”

	Country/Region	Q1 2017 Avg. Mbps	QoQ Change	YoY Change
–	Global	7.2	2.3%	15%
1	South Korea	28.6	9.3%	-1.7%
2	Norway	23.5	-0.4%	10%
3	Sweden	22.5	-1.3%	9.2%
4	Hong Kong	21.9	-0.2%	10%
5	Switzerland	21.7	2.1%	16%
6	Finland	20.5	-0.7%	15%
7	Singapore	20.3	0.8%	23%
8	Japan	20.2	3.1%	11%
9	Denmark	20.1	-2.9%	17%
10	United States	18.7	8.8%	22%

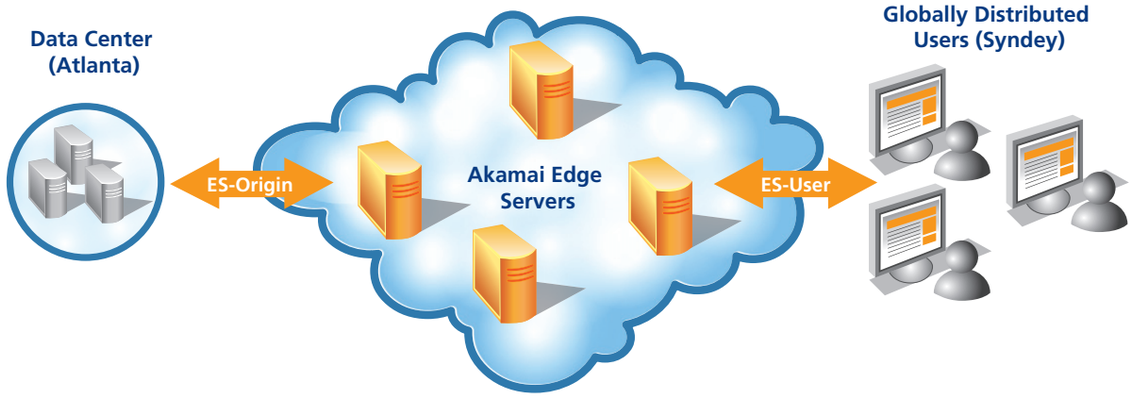
Figure 6: Average Connection Speed (IPv4) by Country/Region

Akamai Corp., “Akamai’s [state of the internet] Q1 2017 report ”

“Middle Mile” by Akamai

Middle Mile With Akamai

Web Application Delivery Example—HTTP over TCP/IP



1 **First Mile**
Application processing time and content generation
0.25 seconds

2 **Middle Mile**
Round Trip Time (RTT) x # Round Trips
1.00 seconds (5 x 200ms)

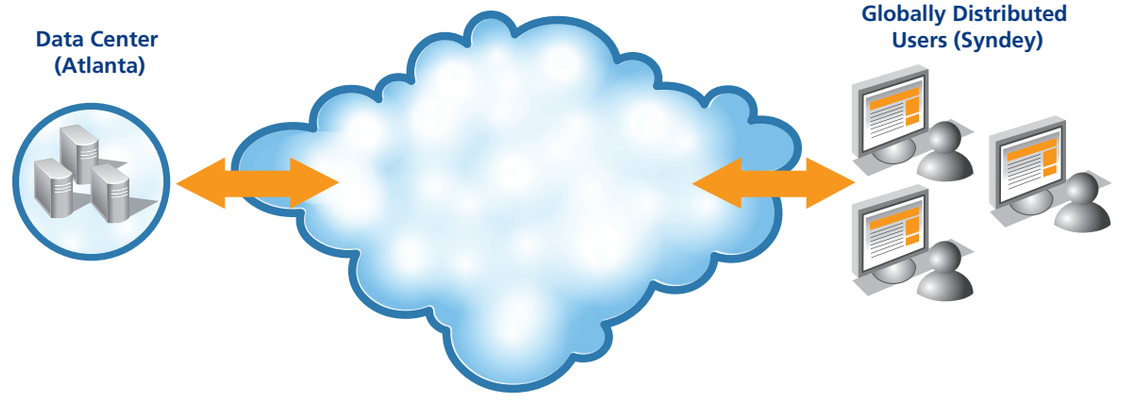
3 **Last Mile**
Bandwidth-related transfer time
0.20 seconds

Rendering the same Web page now requires only 5 round trips, compounded by a reduction in RTT

Total response time = 1.45 sec.— Akamai reduces to 5 round trips

Middle Mile Without Akamai

Web Application Delivery Example—HTTP over TCP/IP



1 **First Mile**
Application processing time and content generation
0.25 seconds

2 **Middle Mile**
Round Trip Time (RTT) x # Round Trips
7.75 seconds (31 x 250ms)

3 **Last Mile**
Bandwidth-related transfer time
0.20 seconds

From Atlanta to Sydney, rendering a typical 70K Web page with 25 objects requires 31 rount trips

Total response time = 8.20 sec.— 95% of total response time is due to the middle mile

Akamai, “Akamai’s EdgePlatform for Application Acceleration”, 2007

Akamai sells (small) RTT

- Akamai deploys 100,000 edge servers for CDN (Contents Delivery Network) business.
- Offer “Best” server with DNS.
  - stable connectivity
  - TCP optimization
  - prefetching

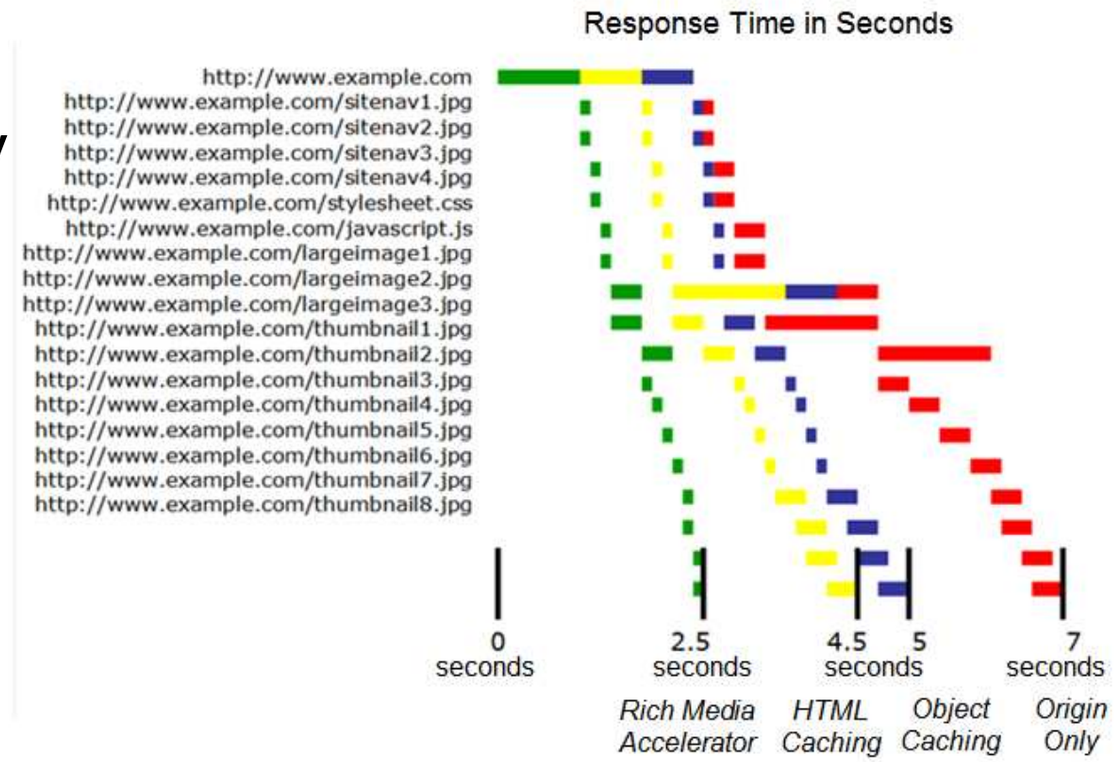
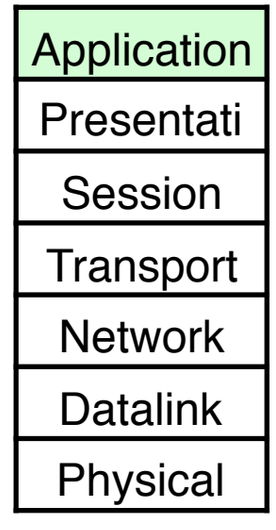


Figure 6. Akamai’s path and connection optimization techniques work together to significantly boost performance

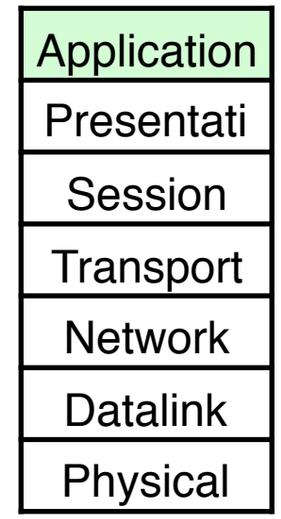
Akamai Corp., “Beyond Caching: The User Experience Impact of Accelerating Rich, Dynamic Content across the Internet”, White Paper. (2013)

Google SPDY (Obsoleted)



- Multiplexing (known as pipelining)
  - More than one requests within a single connection.
- Prioritized Request
  - Avoid Head-of-Line blocking (HoL) by non-critical contents.
- Header Compression
  - Avoid redundant header information.
- Server Push
  - Bundled contents set delivered without requests, same as pre-fetch in CDN.
- SPDY is not an authorized STD, but rapidly deployed even both server/client replace are required: Google, Facebook, Akamai, LINE, Apple, MS…

HTTP 2.0 (RFC7540)

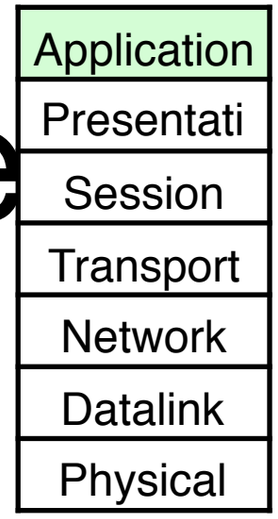


- Based on SPDY/3
- Published at May 2015
  - Google announced SPDY fade out plan :
    - Chrome browser will not support SPDY after 2016.
    - Others also followed Google.
- Not mandate “encryption” on specification due to strong arguments, such as, middle box, lawful inspection, inline compression.

Outline

- Administravia
- Homework review
- User eXperience on Web service (cont'd)
  - IW10
  - HTTP
- Advanced Topic
  - QUIC
- NS2 simulation

Latency of network service caused by …

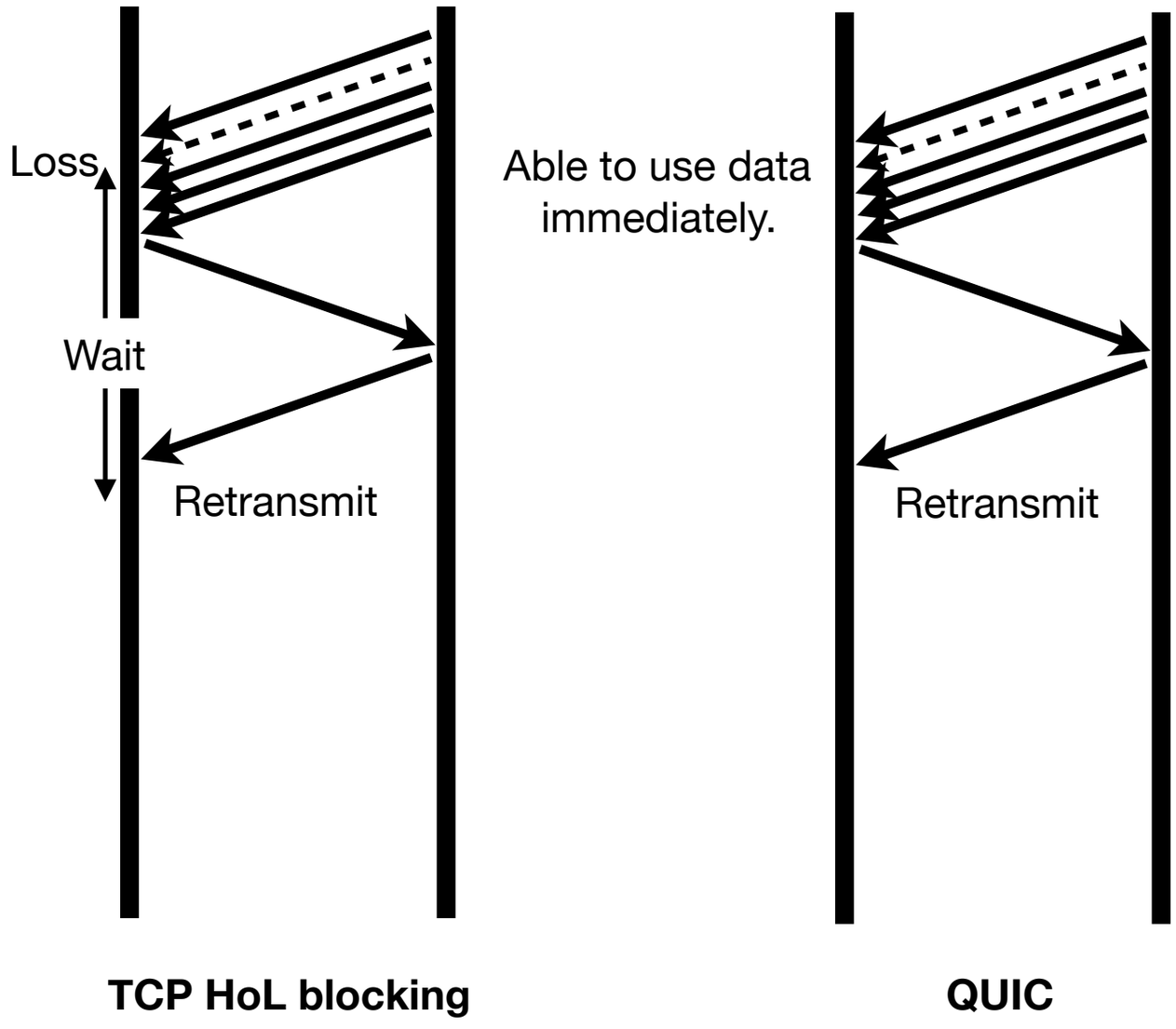


- Intra-node (server, client) :
  - Can be reduced by improving node performance and/or software, but not discussed in this class.
- Client - Server communication :
  - Global network, or the Internet
- Inter-node in server cluster :
  - Depending on communication in DC network

HoL Blocking : TCP

- If a segment is discarded, the later sequence packets must wait until receiving the retransmitted one, because TCP is a byte-stream transport.

- Message-oriented transport like SCTP can avoid such HoL blocking.



TCP HoL blocking **QUIC**

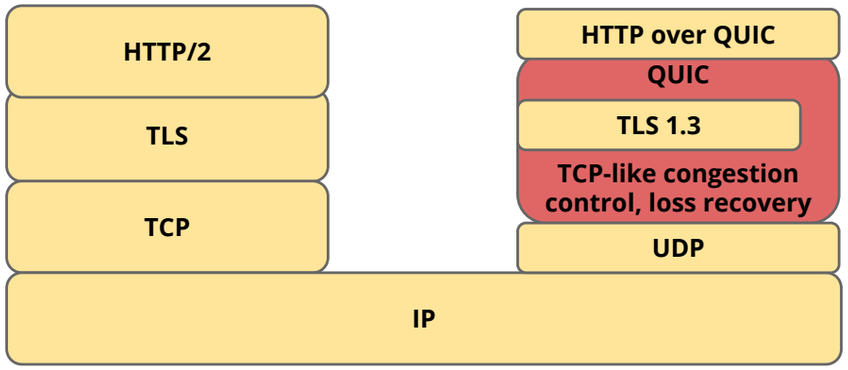
QUIC UDP Internet Connections (QUIC)

A QUIC history

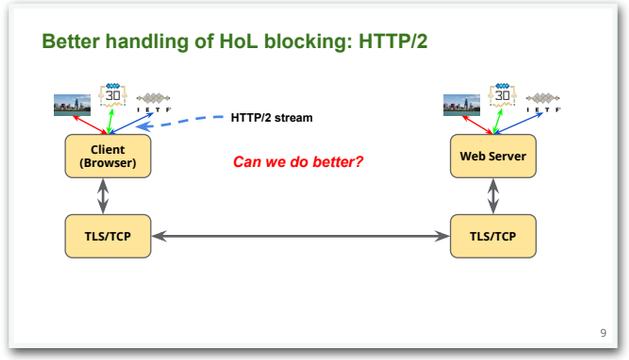
- Experimental protocol, deployed at Google starting in 2014
  - Between Google services and Chrome
  - Improved page load latency, video rebuffer rate
  - Successful experiment today
  - ~35% of Google's egress traffic (~7% of Internet traffic)
  - Akamai deployment in 2016
- QUIC wg formed in Oct 2016
  - Modularize and standardize QUIC in parts
  - HTTP as initial application

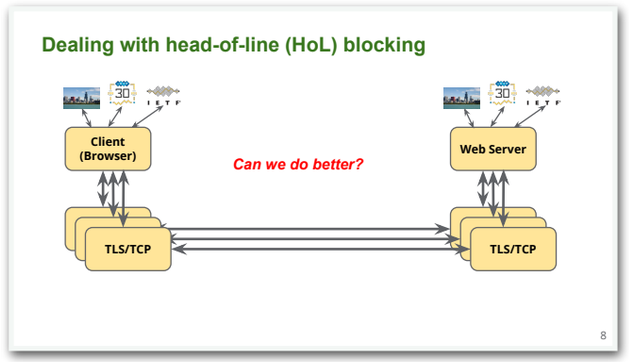
4

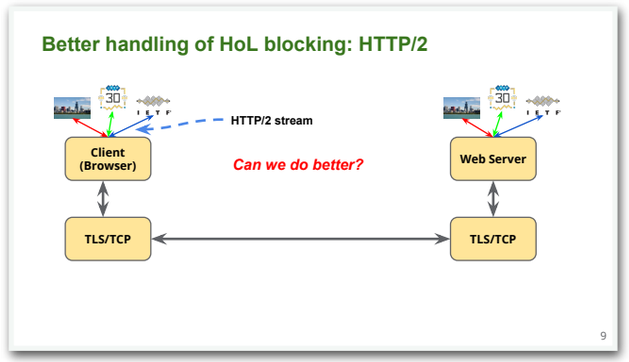
QUIC working group

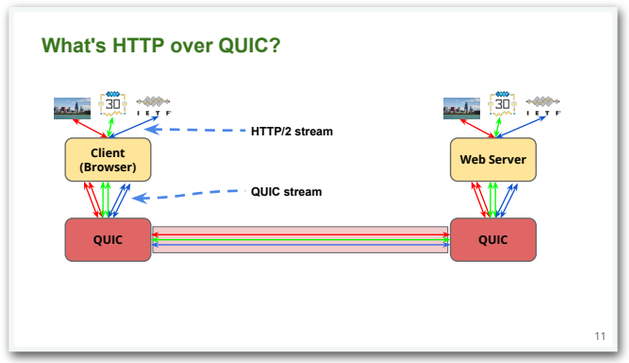


14







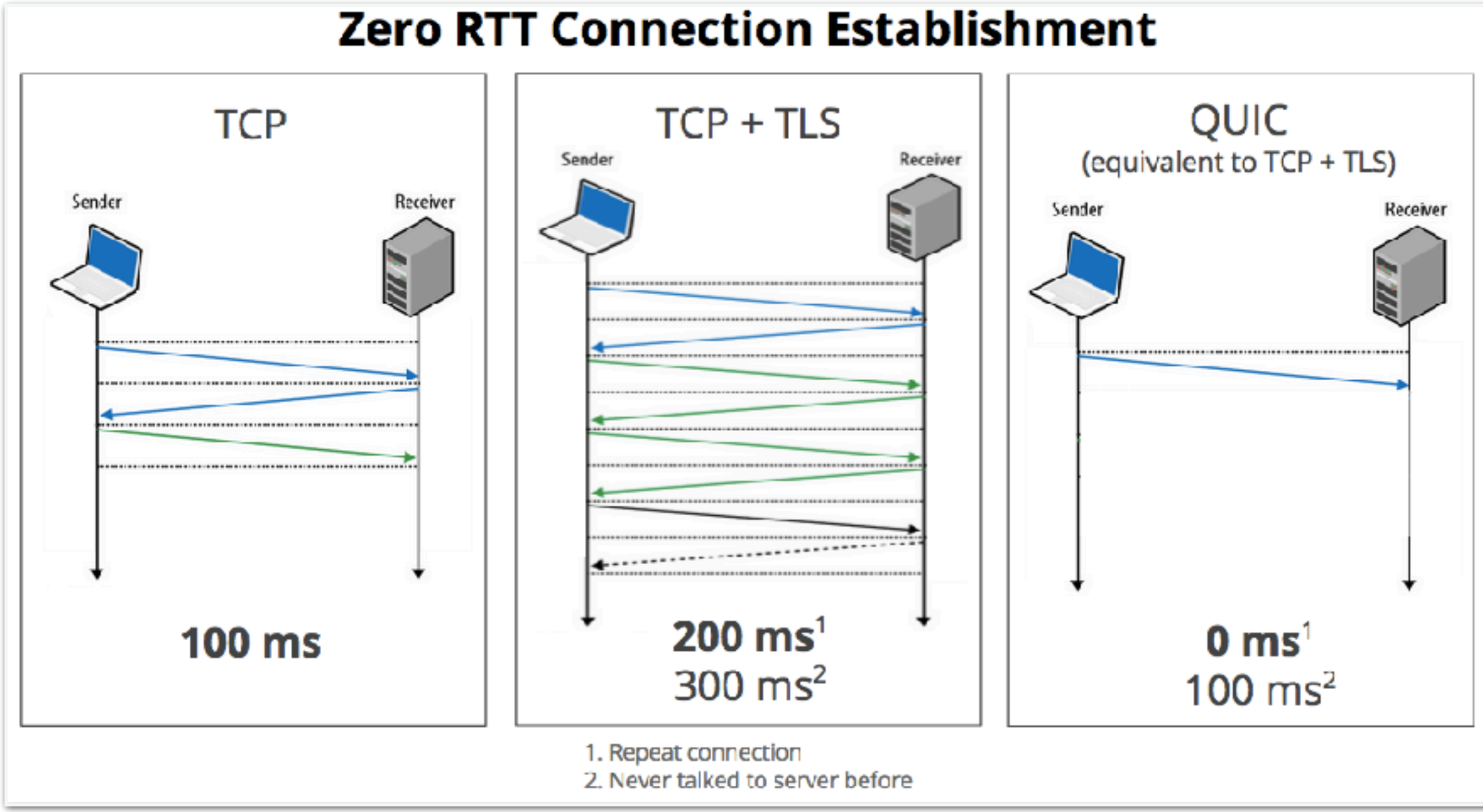


Jana Iyenger, “QUIC Tutorial A New Internet Transport”, IETF98 2017

QUIC Design Aspirations

- Deployability and evolvability
- Low latency connection establishment
- Multistreaming
- Better loss recovery and flexible congestion control
- Resilience to NAT-rebinding (Connection IDs vs. 4-tuple)
- Multipath for resilience and load sharing

Jana Iyenger, “QUIC Tutorial A New Internet Transport”, IETF98
2017



A QUIC update on Google's experimental transport, April 17, 2015

Outline

- Administravia
- Homework review
- User eXperience on Web service (cont'd)
  - IW10
  - HTTP
- Advanced Topic
  - QUIC
- NS2 simulation

NS-2 script

- Describe two type of components with oTCL

  1.Event and Schedules

    - Scenario with time-line

  2.System Configuration

    - Network Topology
    - Transport Agent
    - Application

NS2 (Network Simulator 2)

- Has been developed by VINT (Virtual InterNetwork Testbed) Project funded by US DARPA(DefenseAdvanced Research Projects Agency)
- Simulates packet switched network with discrete event driven simulation approach.
  - Transport, Routing, Wireless link, mobile
- A lot of remarkable outputs from NS2
  - esp. to improve transport behavior, such as, TCP.

NS2 (Network Simulator 2) (cont'd)

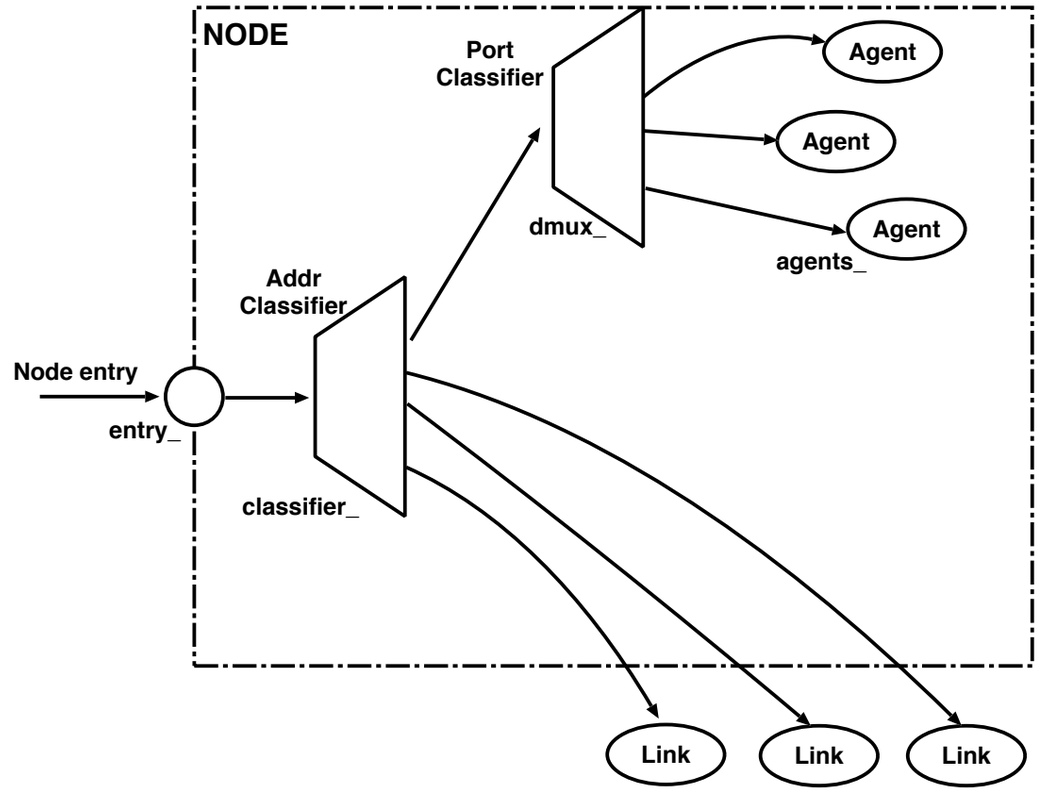
- Written by oTCL (object oriented extension for TCL) and C++
  - Easy to implement and to evaluate new ideas.
- Simulation scenarios are described by oTCL program.
  - A network specified as a graph which consists of nodes and links.
  - Transport stacks and applications are agents attached to nodes .
- Though NS3 project successor of NS2 has been started, this lecture uses NS2 due to a visualization issue in NS3.

NS2 : ex-1.tcl
Event and Schedules

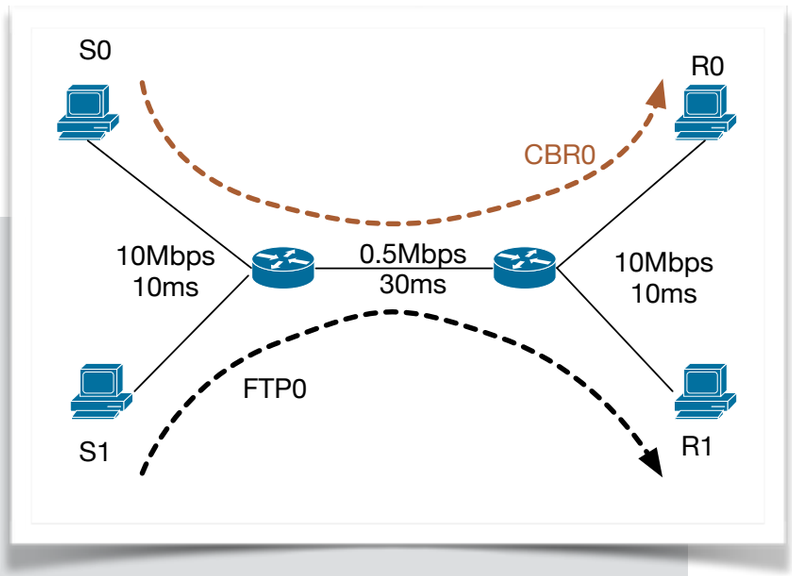
```
1 set ns [new Simulator]
……
109 $ns at 1.0 "$ftp0 start; $ns trace-annotate \"Time:[$ns now] Start FTP\""
110 #$ns at 51.0 "$cbr0 start;
111 #$ns trace-annotate \"Time:[$ns now] Start CBR interval …
112 #$ns at 101.0 "$cbr0 stop; $ns trace-annotate \"Time:[$ns now] Stop CBR\""
113 $ns at 200.0 "$ftp0 stop; $ns trace-annotate \"Time:[$ns now] Stop FTP\""
114 $ns at 200.0 “finish"
```

NS2 : Network Topolog

- Node
  - Interface, Addressing, Routing, …
- Link
  - Delay, Bandwidth, Queue, Loss, ..
- Queue (packet buffer)
  - Absorb differences of link speed
  - **Size**
  - Scheduling policy (FCFS/DropTail)



NS2: Network Topology (cont'd)




```
25 # Dumbbell topology
26 #
27 # s0              r0
28 #   \            /
29 #    \          /
30 #     n0------n1
31 #    /          \
32 #   /            \
33 # s1              r1
34 #
35 set s0 [$ns node]
36 set s1 [$ns node]
37 set n0 [$ns node]
38 set n1 [$ns node]
39 set r0 [$ns node]
40 set r1 [$ns node]
41
42 $ns duplex-link $s0 $n0 10Mb 10ms DropTail
43 $ns duplex-link $s1 $n0 10Mb 10ms DropTail
44
45 $ns duplex-link $n0 $n1 0.5Mb 30ms DropTail
46 $ns duplex-link-op $n0 $n1 queuePos 0.5
47
48 $ns duplex-link $n1 $r0 10Mb 10ms DropTail
49 $ns duplex-link $n1 $r1 10Mb 10ms DropTail
50
51 $ns duplex-link-op $n0 $s0 orient up-left
52 $ns duplex-link-op $n0 $s1 orient down-left
53 $ns duplex-link-op $n0 $n1 orient right
54 $ns duplex-link-op $n1 $r0 orient up-right
55 $ns duplex-link-op $n1 $r1 orient down-right
56
57 $ns queue-limit $s0 $n0 50
58 $ns queue-limit $s1 $n0 50
59
```

← Dumbbell topology

← Link definitions
Bottleneck link of 0.5Mbps,
Other link of 10Mbps

← Queue definitions at bottleneck link

NS2: Transport Agents

- Middle between Applications and Nodes
- TCP (Various CC algorithms, NewReno, CUBIC, …)
  - a lot of internal variables, e.g., ssthresh_, cwnd_…
  - Agent/TCP/NewReno -> Agent/TCPSink
    - One-way TCP data stream
- UDP
  - Agent/UDP -> Agent/Null
    - One-way TCP data stream, Just sends packets (datagrams)

NS2: Applications

- On a top of Transport Agents
  - FTP : Data transfer application
    - Simulate bulk data transfer
  - CBR : Constant Bit Rate
    - Simulate voice / video stream
      - packetSize_, interval_, …
      - BW = packetSize_ / interval_

NS2: NS2 : ex-1.tcl
Applications and Transports

```
62 # FTP sender: Transport and Application
63 Agent/TCP set window_ 200000
64 Agent/TCP set ssthresh_ 200000
65 set tcp0 [new Agent/TCP/Newreno]
66 $tcp0 set class_ 1
67 $ns attach-agent $s1 $tcp0
68
69 set sink0 [new Agent/TCPSink]
70 $ns attach-agent $r1 $sink0
71
72 set ftp0 [new Application/FTP]
73 $ftp0 attach-agent $tcp0
74
75 $ns connect $tcp0 $sink0
76
77 $tcp0 trace cwnd_
78 $tcp0 trace ssthresh_
79 $tcp0 attach-trace $ftcp
80
81 # CBR sender: Transport and Application
82 set udp0 [new Agent/UDP]
83 $ns attach-agent $s0 $udp0
84
85 set cbr0 [new Application/Traffic/CBR]
86 $cbr0 set packetSize_ 1000
87 $cbr0 set interval_ 0.05
88 $cbr0 attach-agent $udp0
89 $udp0 set class_ 0
90
91 # CBR receiver: Transport and Application
92 set null0 [new Agent/Null]
93 $ns attach-agent $r0 $null0
94
95 $ns connect $udp0 $null0
96
```

TCP sender side
Transport: TCP NewReno
Class ID（shown as red packets in nam）

TCP receiver side

FTP application

Connecting TCP agents

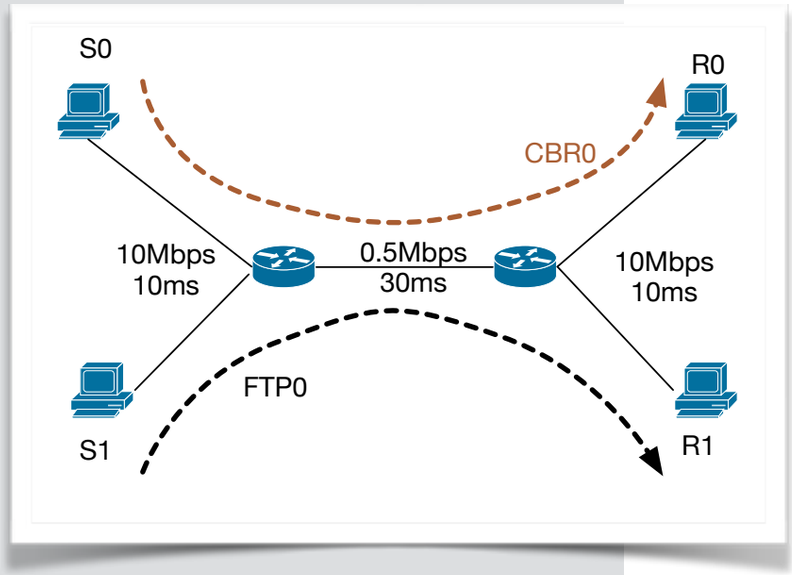
Specify trace parameters

UDP sender side

CBR application
Class ID(shown as black in nam)

UDP receiver side

Connect UDP sender and receiver agents



NS2: Tracing TCP window behavior

- TCP window parameters are recorded in trace files, in case of changing parameter.
  - In ex1-sample.tcl, cwnd and ssthresh are recored into ex1-sample.tcp.

ex1-sample.tcp

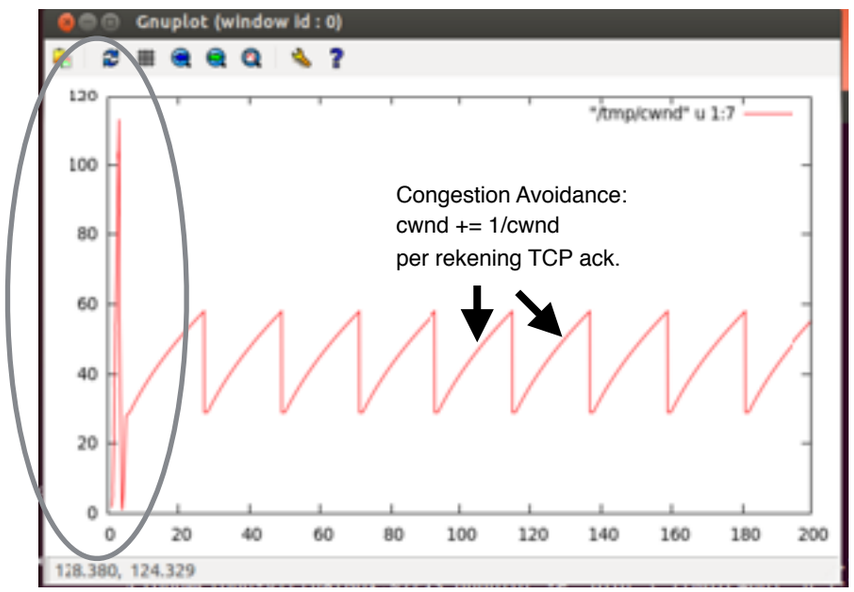
```
105 3.07251  1  0  5  0  cwnd_ 106.000
106 3.08915  1  0  5  0  cwnd_ 107.000
107 3.10579  1  0  5  0  cwnd_ 108.000
108 3.12243  1  0  5  0  cwnd_ 109.000
109 3.13907  1  0  5  0  cwnd_ 110.000
110 3.15571  1  0  5  0  cwnd_ 111.000
111 3.17235  1  0  5  0  cwnd_ 112.000
112 3.18899  1  0  5  0  cwnd_ 113.000
113 3.23891  1  0  5  0  ssthresh_ 56
114 3.23891  1  0  5  0  cwnd_ 56.500
115 4.02891  1  0  5  0  ssthresh_ 28
116 4.02891  1  0  5  0  cwnd_ 1.000
117 4.17075  1  0  5  0  cwnd_ 2.000
118 4.22067  1  0  5  0  cwnd_ 3.000
119 4.30640  1  0  5  0  cwnd_ 4.000
120 4.33968  1  0  5  0  cwnd_ 5.000
```

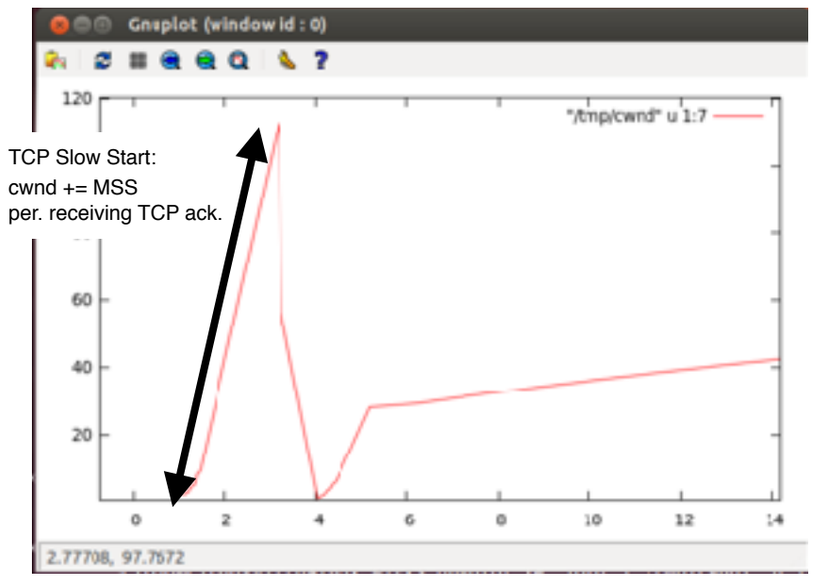
Colum	
1	time in simulation
2	source node #
3	source port #
4	destination node #
5	destination port #
6	parameter updated by the event cwnd:congestion windwow size ssthresh :slow-start threshold
7	parameter value

Today’s quiz1/2

- Goal: To understand slow-start and congestion avoidance behaviors.
- After download “ex1-sample.tcl” script from the course Web, Run it using “ns”
- Plot the dependency of ***cwnd*** on simulation time.
- Upload captured snapshot via the course web **during this class**.
- In case of using docker, you are recommended to run gnuplot on server side.

```
ns@ns-VirtualBox:~/Ex$ grep "1  0  5  0  cwnd" ex1-sample.tcp > /tmp/cwnd
ns@ns-VirtualBox:~/Ex$ gnuplot -e "plot \"/tmp/cwnd\" u 1:7 w l; pause -1"
```

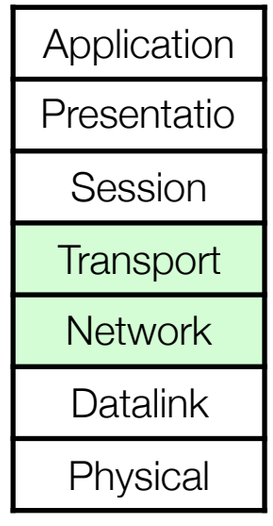




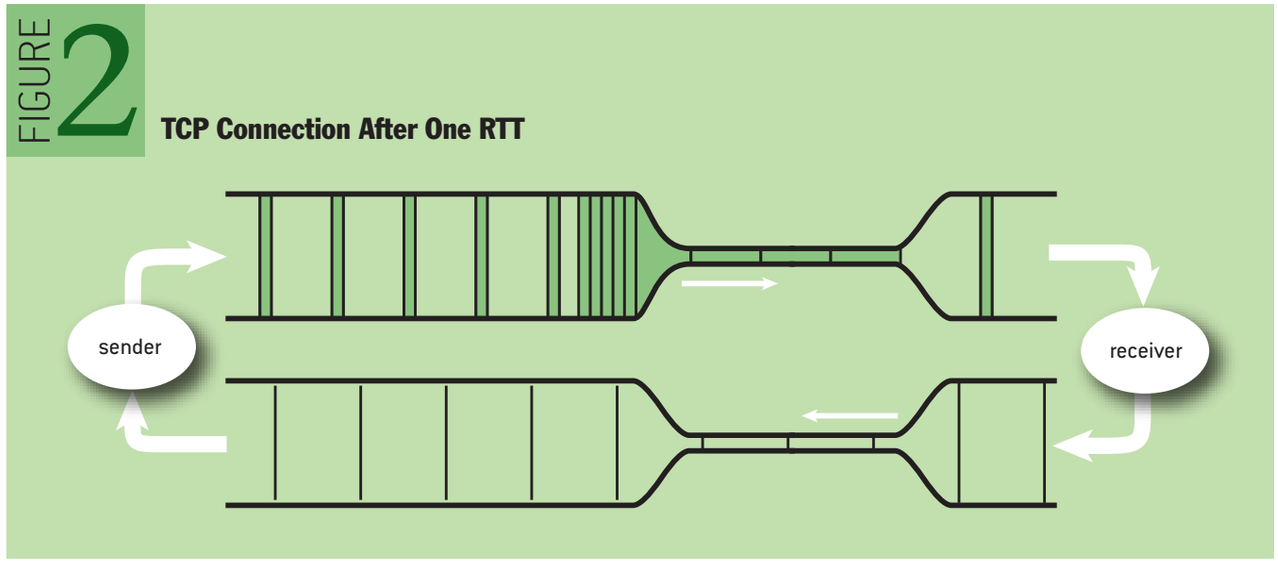
Today’s quiz 2/2

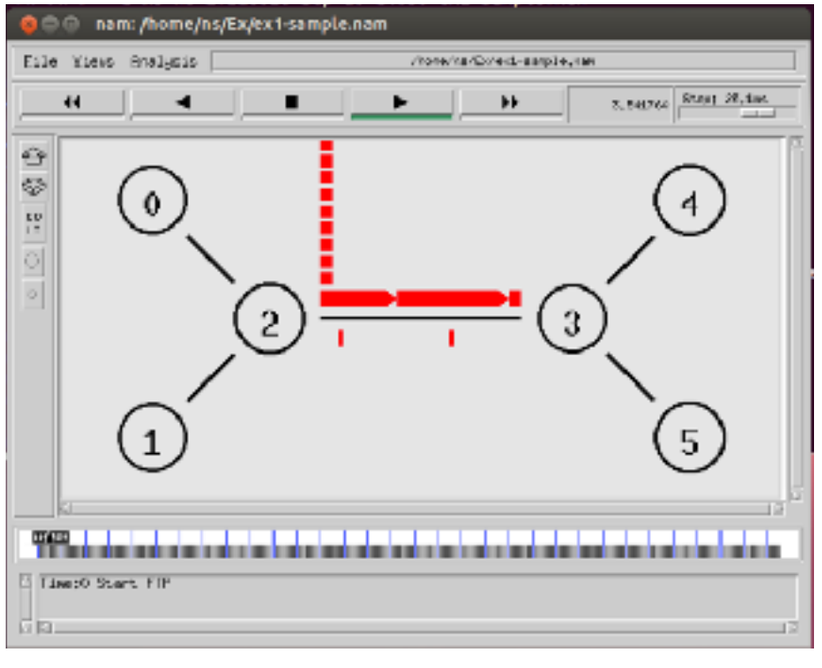
- In ex1-sample.tcl, the queue size of the bottleneck link is bigger than optimal.
  How much queue size, C, is optimal in the number of pkts ?
  Note that the size of 1pkt is 1000Bytes.
    - The end-to-end throughput should be more than or equal to the bottleneck link capacity.
    - In terms of reducing latency, queue size should be small as possible.
- Plot the dependency of ***cwnd*** on simulated time.
- Upload captured snapshot via the course web **during this class**.

Queue / packet buffer at router interface



- Absorb burst traffic caused by different rate links.
  - In case of small sizes: Unable to absorb large burst.
  - Large: longer queuing delays. Buffer space does not overflow as quickly but the buffers become full due to (greedy) TCP's behavior
  - C = BW x RTT (C: Optimal buffer capacity in case of single TCP flow)





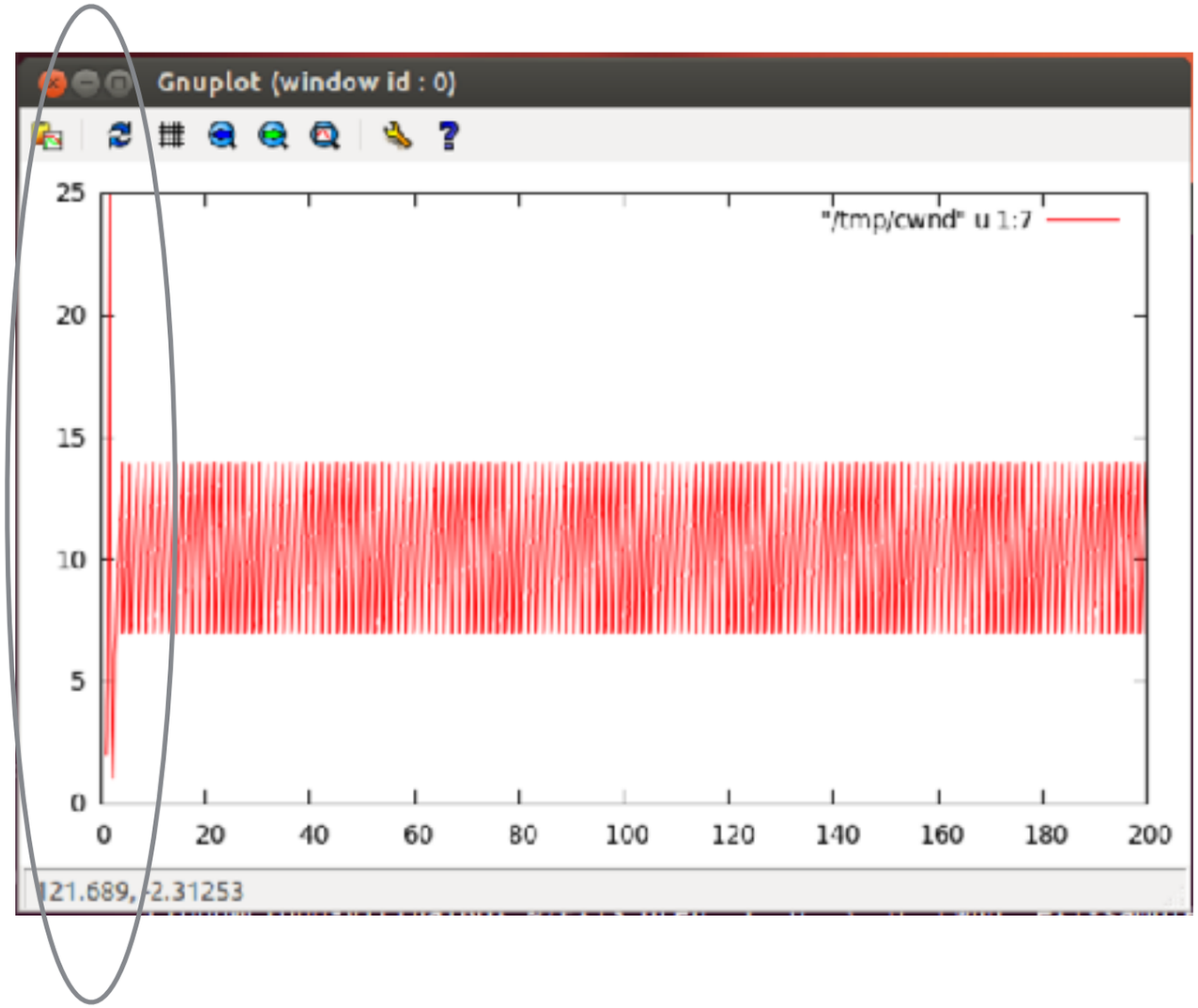
Optimal Queue size ?

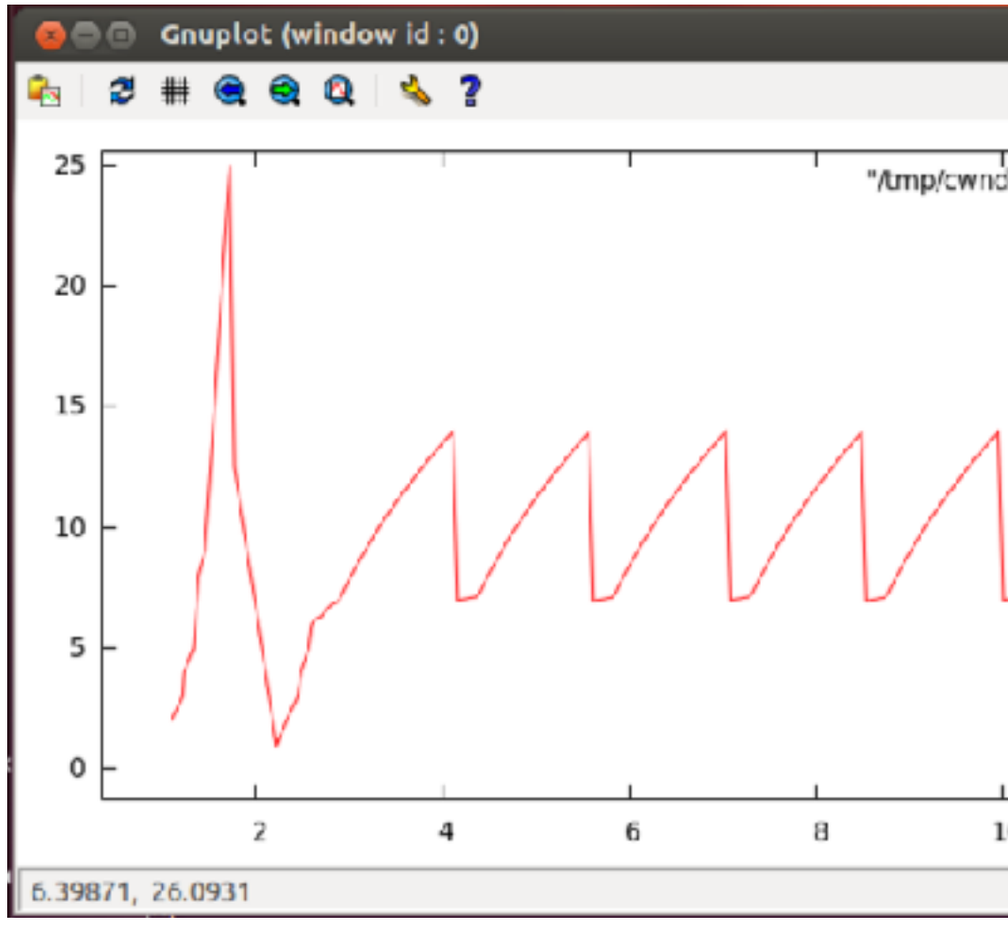
- Too big queue size at bottle neck in ex1-sample.tcl
- How much is the Optimal Queue size, C ? (1000Bytes = 1pkt)
- C = BW x RTT = 0.5Mbps x 100ms

```
25 # Dumbbell topology
26 #
27 # s0              r0
28 #   \            /
29 #    \          /
30 #     n0------n1
31 #    /          \
32 #   /            \
33 # s1              r1
34 #
.....
41
42 $ns duplex-link $s0 $n0 10Mb 10ms DropTail
43 $ns duplex-link $s1 $n0 10Mb 10ms DropTail
44
45 $ns duplex-link $n0 $n1 0.5Mb 30ms DropTail
46 $ns duplex-link-op $n0 $n1 queuePos 0.5
47
48 $ns duplex-link $n1 $r0 10Mb 10ms DropTail
49 $ns duplex-link $n1 $r1 10Mb 10ms DropTail
50
.........
56
57 $ns queue-limit $s0 $n0 50
58 $ns queue-limit $s1 $n0 50
```

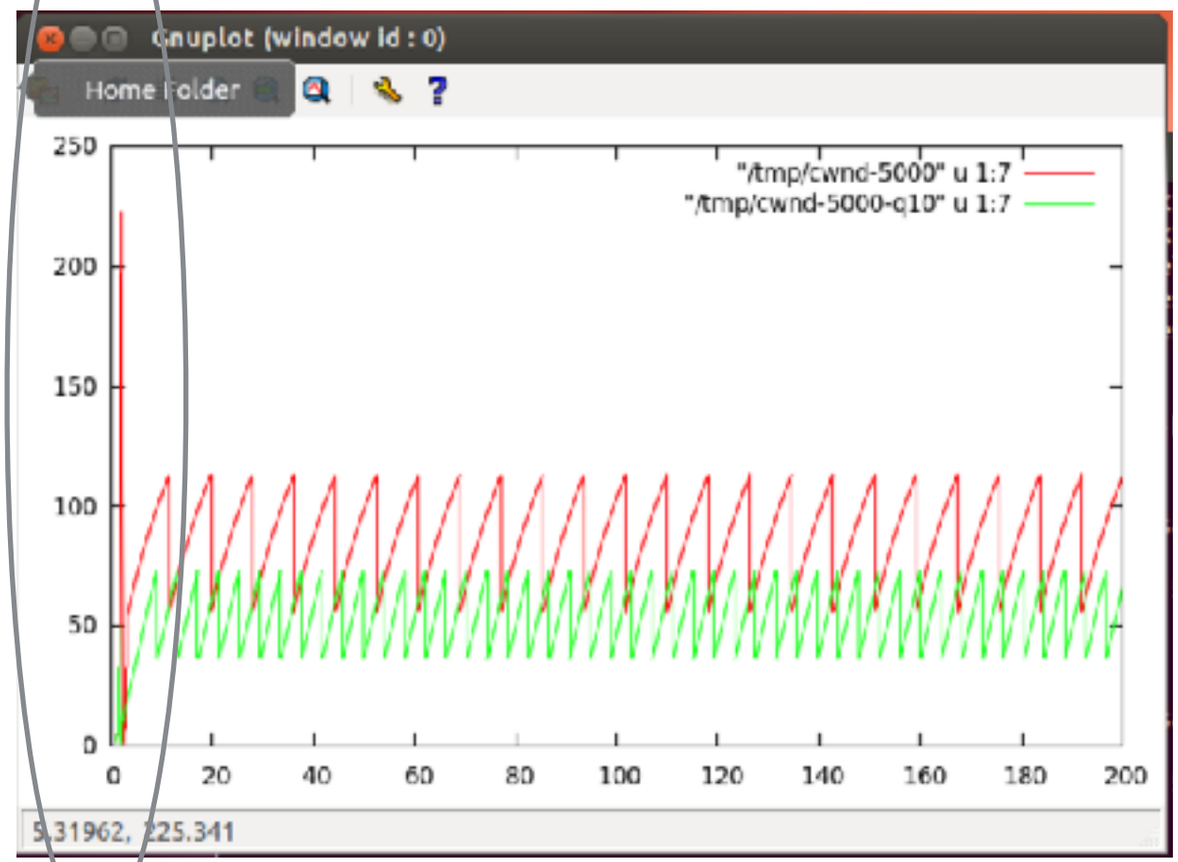
← Queue size definition at the bottleneck.

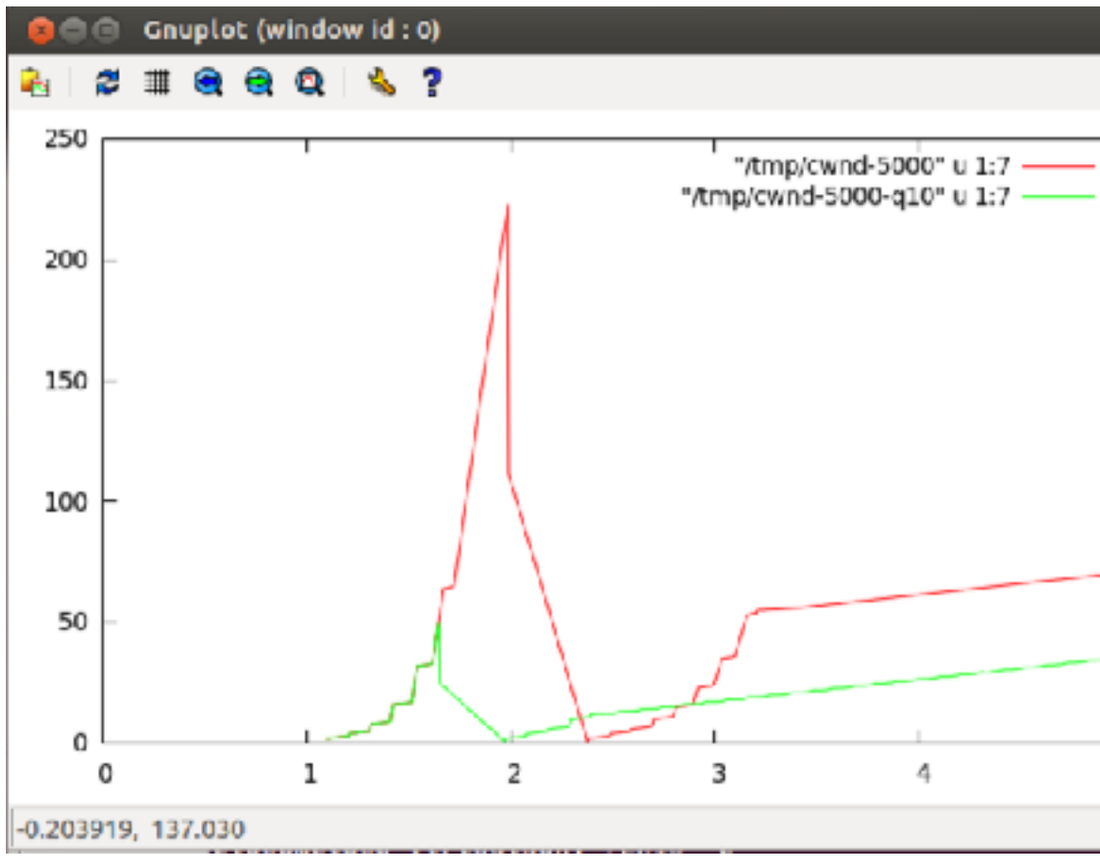
Optimal Queue size ? (cont'd)





Poor Queue size





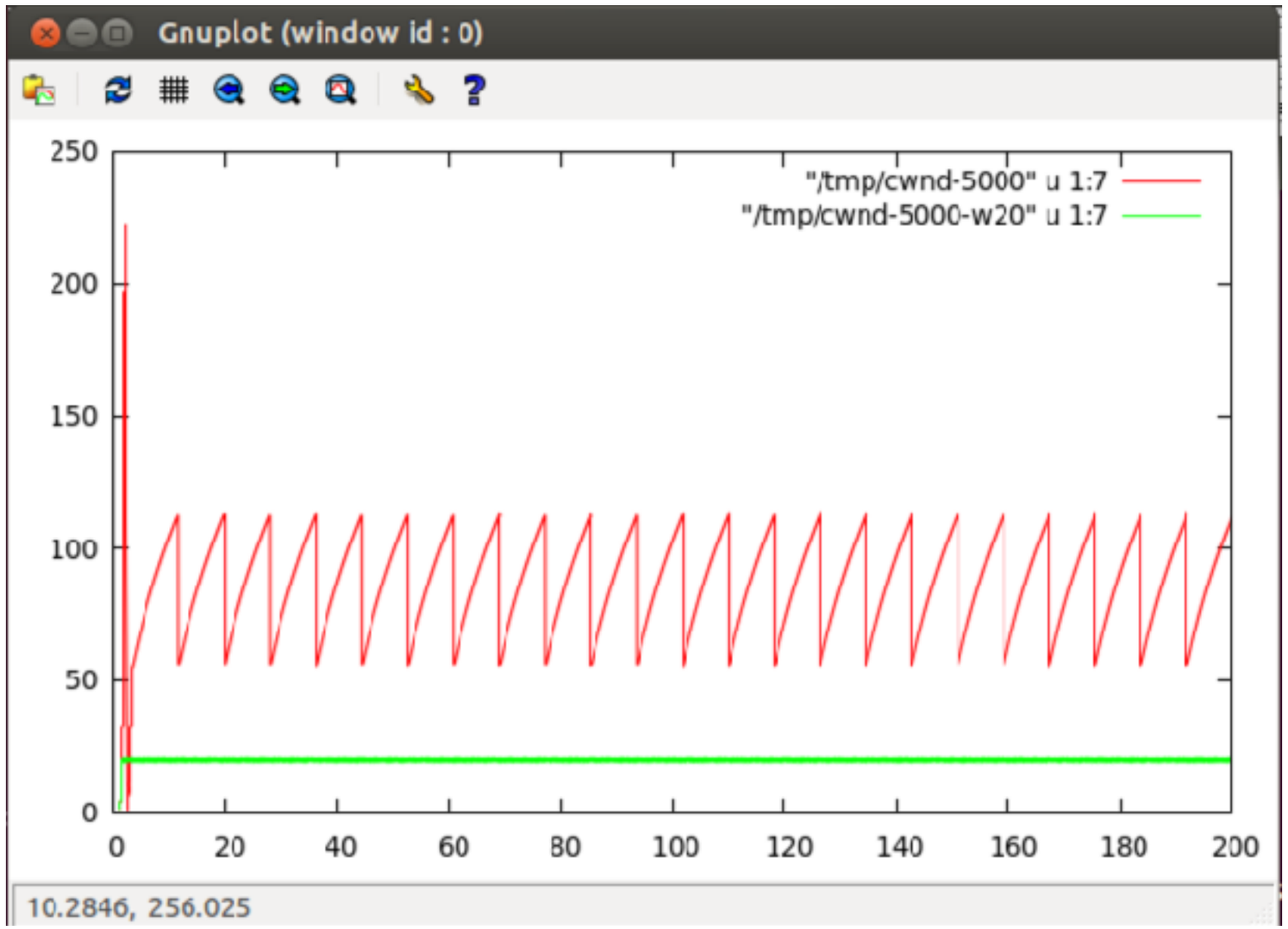
5Mbps @ Bottleneck
Red: 50pkts queue
Green: 10pkts queue

Note: Poor Throughput

- In case of insufficient Congestion Window Size:
- By end system
  - Receiver: TCP Advertised window (rwnd)
    - **SO_RCVBUF** or /proc/sys/net/ipv4/tcp_rmem @Linux
  - Sender:
    - **SO_SNDBUF** or /proc/sys/net/ipv4/tcp_wmem @ Linux
- By network - packet loss
- Note: Linux caches TCP parameters.
  - B(Loss) ≒ 1.22 x MSS/(RTT x sqrt(Loss)) , Loss = packet loss rate
    - TCP parameters, RTT, RTTVAR, cwnd, are cached and reused.

```
# ip route show cache 157.82.6.xxx
157.82.6.xxx from xxx.xxx.xxx.xxx via xxx.xxx.xxx.1 dev eth0
    cache  ipid 0x0f1c rtt 44ms rttvar 39ms cwnd 20
#
```

Insufficient ***cwnd***…



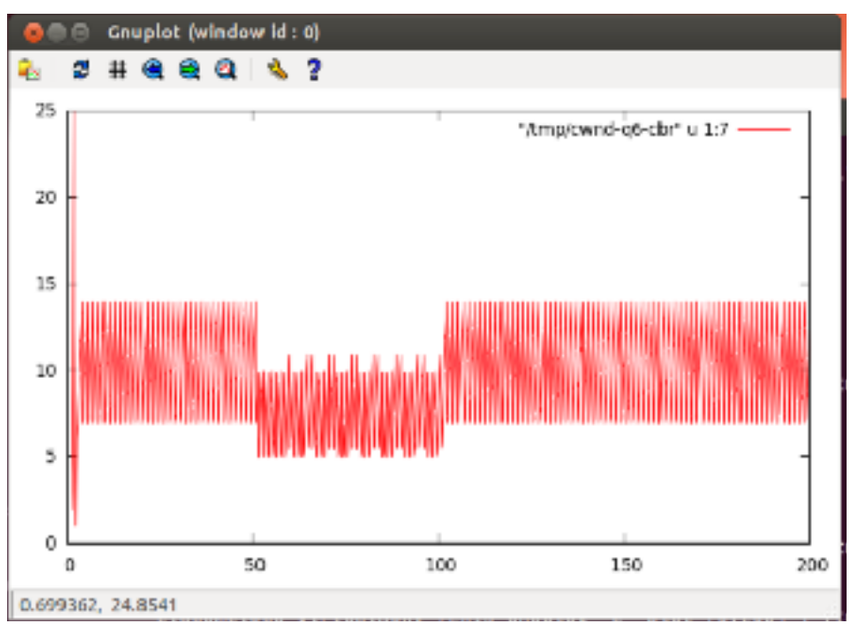
5Mbps @ Bottleneck

Green: maxcwnd of 20 limits the throughput.

TCP with UDP flow

- Enable “cbr0” actions commented out. (160Kbps) with the optimal bottleneck size.

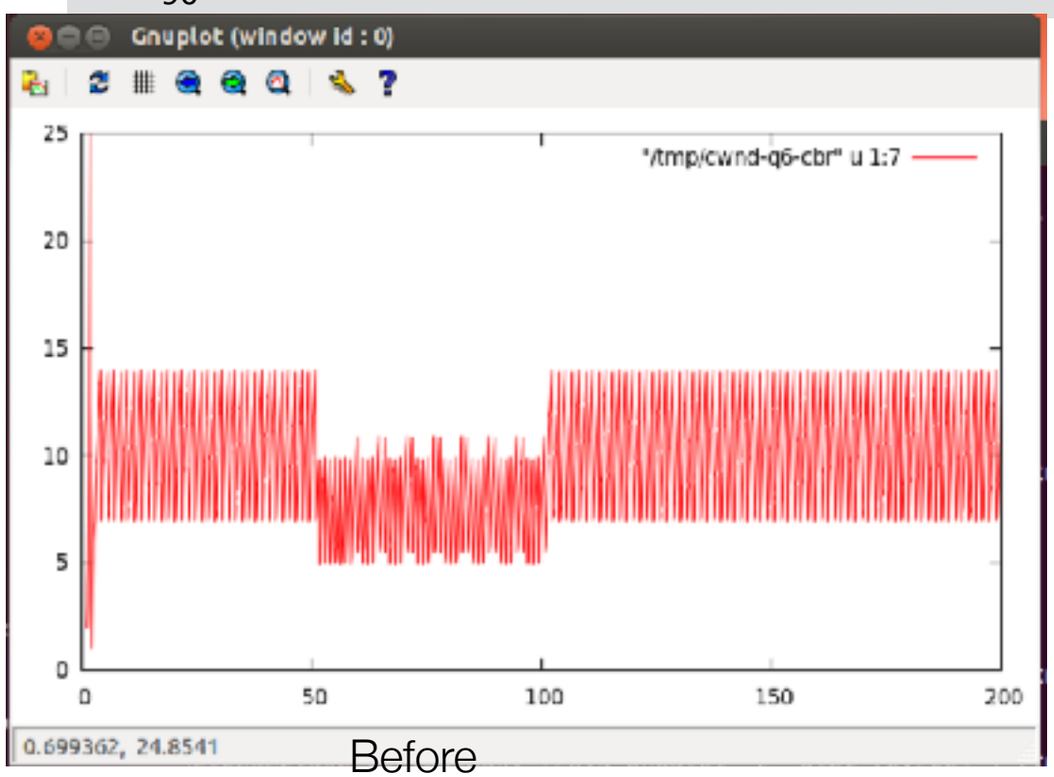
```
1 set ns [new Simulator]
……
109 $ns at 1.0 "$ftp0 start; $ns trace-annotate \"Time:[$ns now] Start FTP\""
110 $ns at 51.0 "$cbr0 start;
111 $ns trace-annotate \"Time:[$ns now] Start CBR interval …
112 $ns at 101.0 "$cbr0 stop; $ns trace-annotate \"Time:[$ns now] Stop CBR\""
113 $ns at 200.0 "$ftp0 stop; $ns trace-annotate \"Time:[$ns now] Stop FTP\""
114 $ns at 200.0 "finish"
```



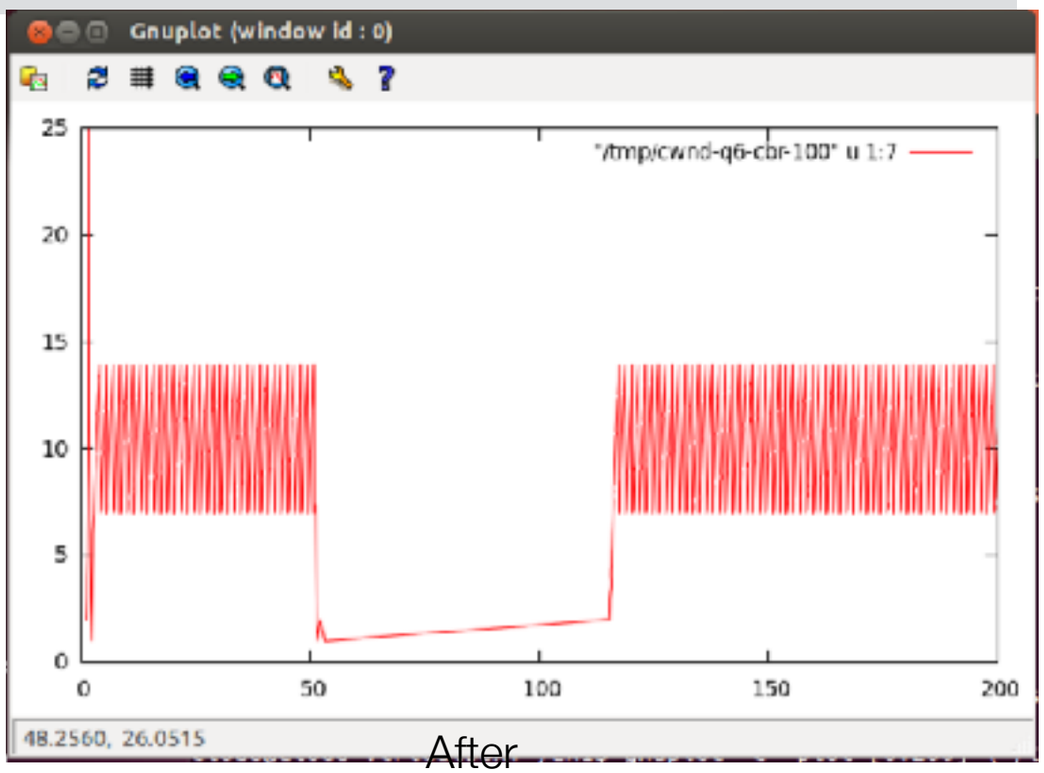
UDP flow unresponsible against congestion, or packet losses.

- Increase UDP flow-rate by decreasing “interval_”
  - Bottleneck queue = Optimal

```
81 # CBR sender: Transport and Application
82 set udp0 [new Agent/UDP]
83 $ns attach-agent $s0 $udp0
84
85 set cbr0 [new Application/Traffic/CBR]
86 $cbr0 set packetSize_ 1000
87 $cbr0 set interval_ 0.05
88 $cbr0 attach-agent $udp0
89 $udp0 set class_ 0
90
```



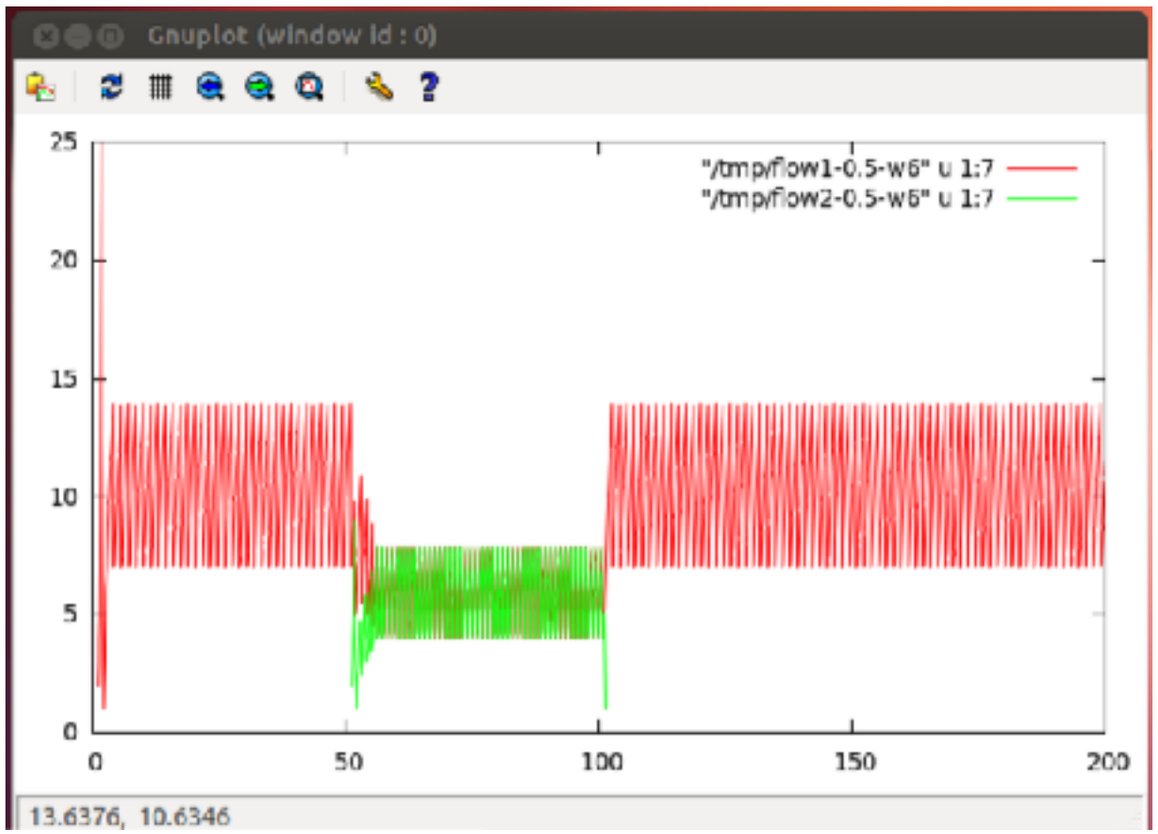
Before



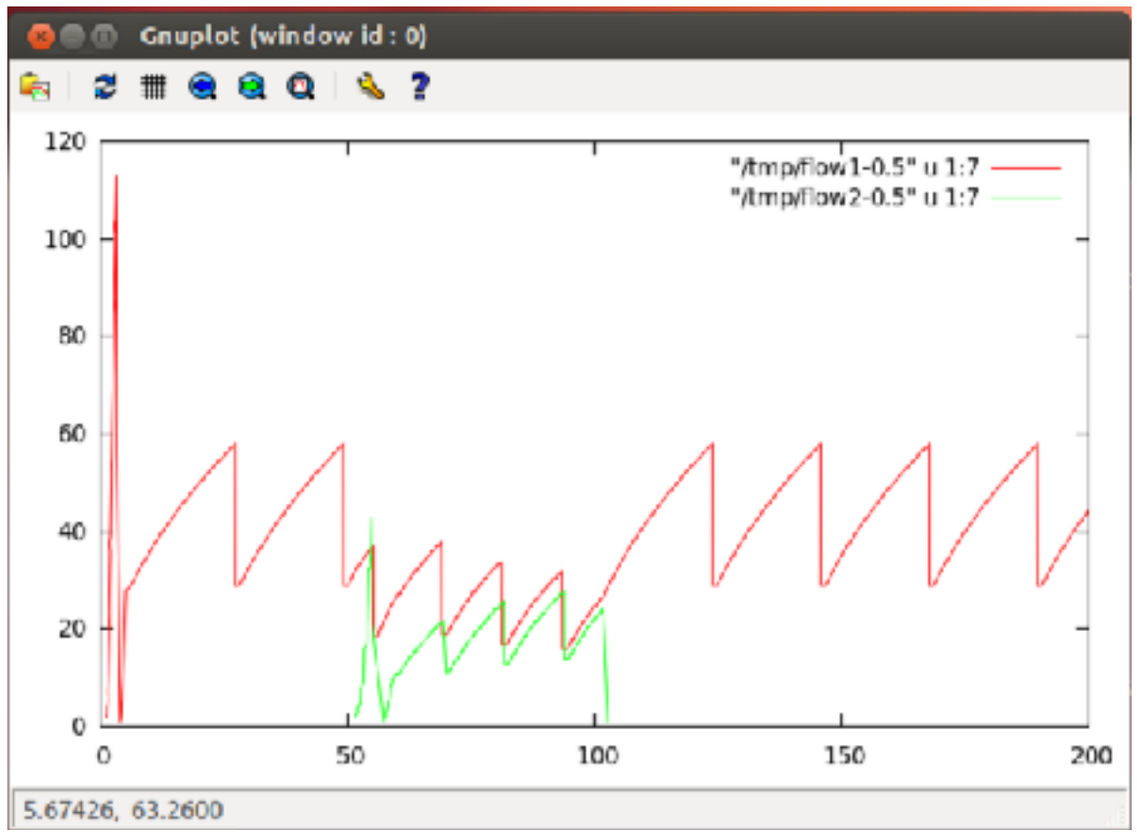
After

TCP flows share a bottleneck

- Add TCP flows instead of UDP



Bottleneck queue : Optimal



Bottleneck queue : Original

Today’s assignment

- Add one more FTP application from node s0 to r0 into the sample script ex1-sample.tcl. (with optimal queue size)
  - Hint: Replace CBR application and UDP agent with FTP and TCP, respectively.
- Plot **cwnd** dependency of two TCP connections’ on the simulated time, and upload it.
- Submit via the course Web.

本日の課題

- サンプルスクリプト ex1-sample.tcl を変更し、s0 -> r0 でデータ転送するもう一つの FTP アプリケーションを追加せよ。
  （最適なキュー長を利用すること）
  - ヒント：CBR アプリケーションおよび UDP agent をそれぞれ FTP および TCP に置き換えれば良い。

- ２つの TCP コネクションの cwnd の時間変化をプロット、アップロードする。

- 講義　Web から提出すること。