Controlling Applications by Managing Network Characteristics

Vahab Pournaghshband
Leonard Kleinrock
Peter Reiher
Alexander Afanasyev

University of California, Los Angeles

June 14, 2012

UCLA

Motivation

- Administrators want to have some control over the data flow in their network
- Why? To prevent illegal or non-business activities to prioritize the network use
- How?
  - Restricting the application using firewalls
  - Deleting/disabling the application
  - Network dissuasion of use of the application

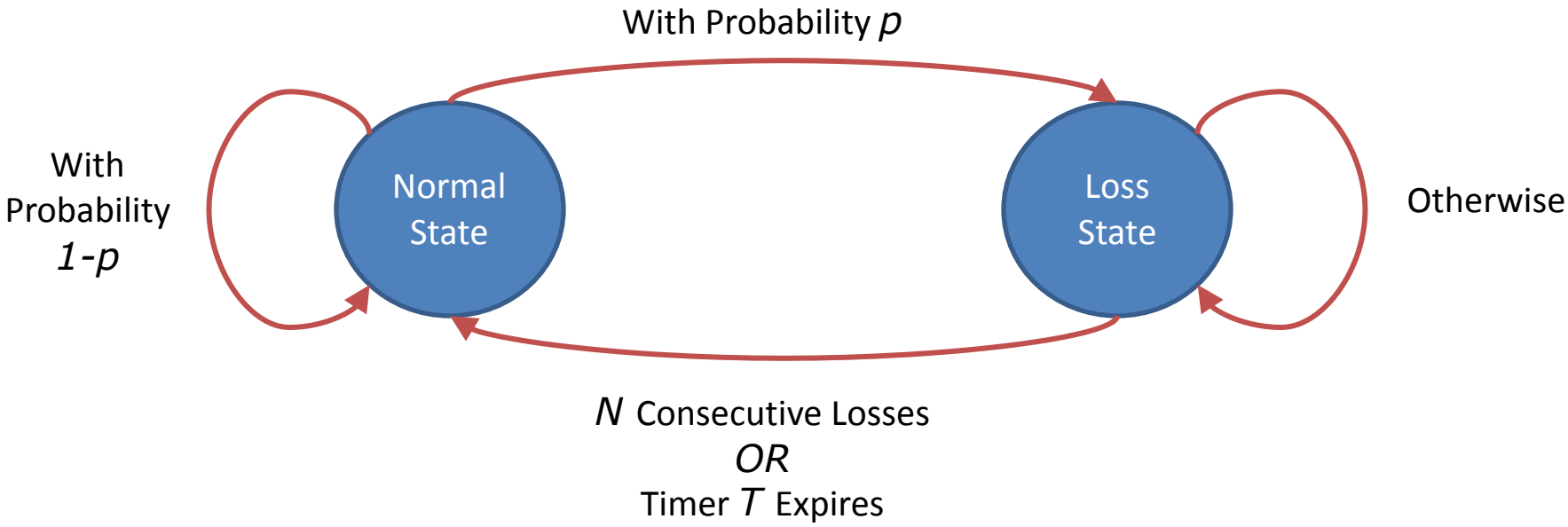
Network Dissuasion

- **Locally denying QoS requirements for the application**
  - **By managing network characteristics (loss, delay, jitter)**
- **Must leave *permitted applications* unaffected or little affected**

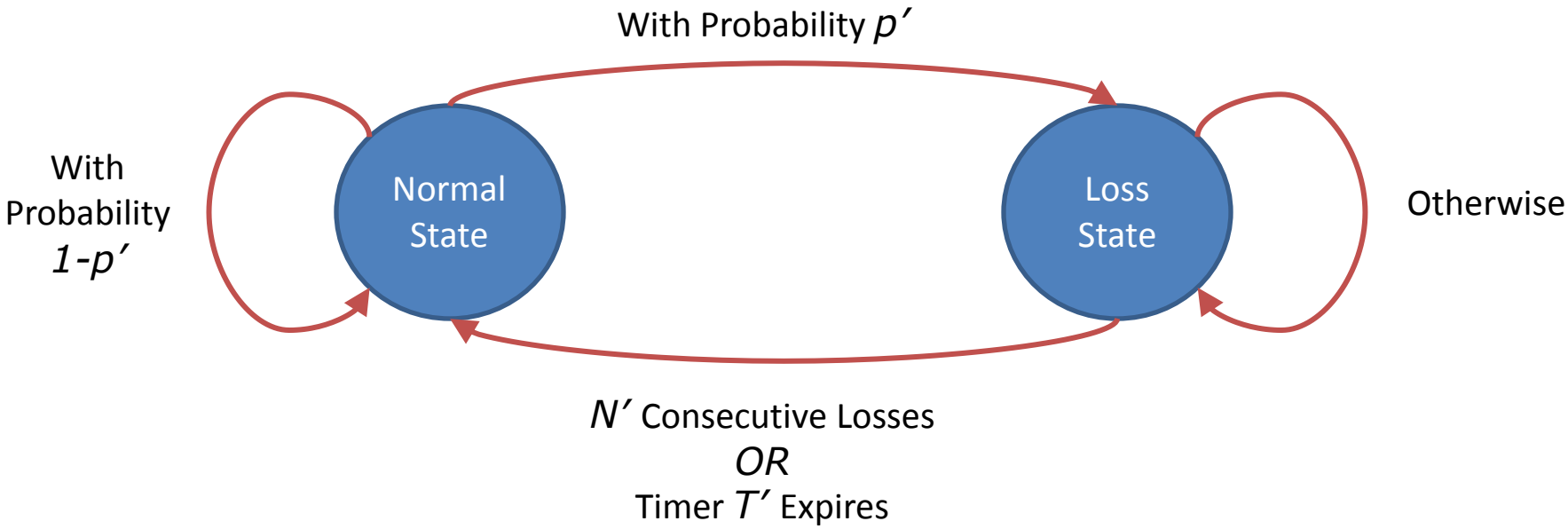
Problem Formulation

- Undesirable Application
  - VoIP
- Permitted Applications
  - DNS
  - HTTP
  - FTP
  - SSH

Our Design: UDP



Our Design: TCP

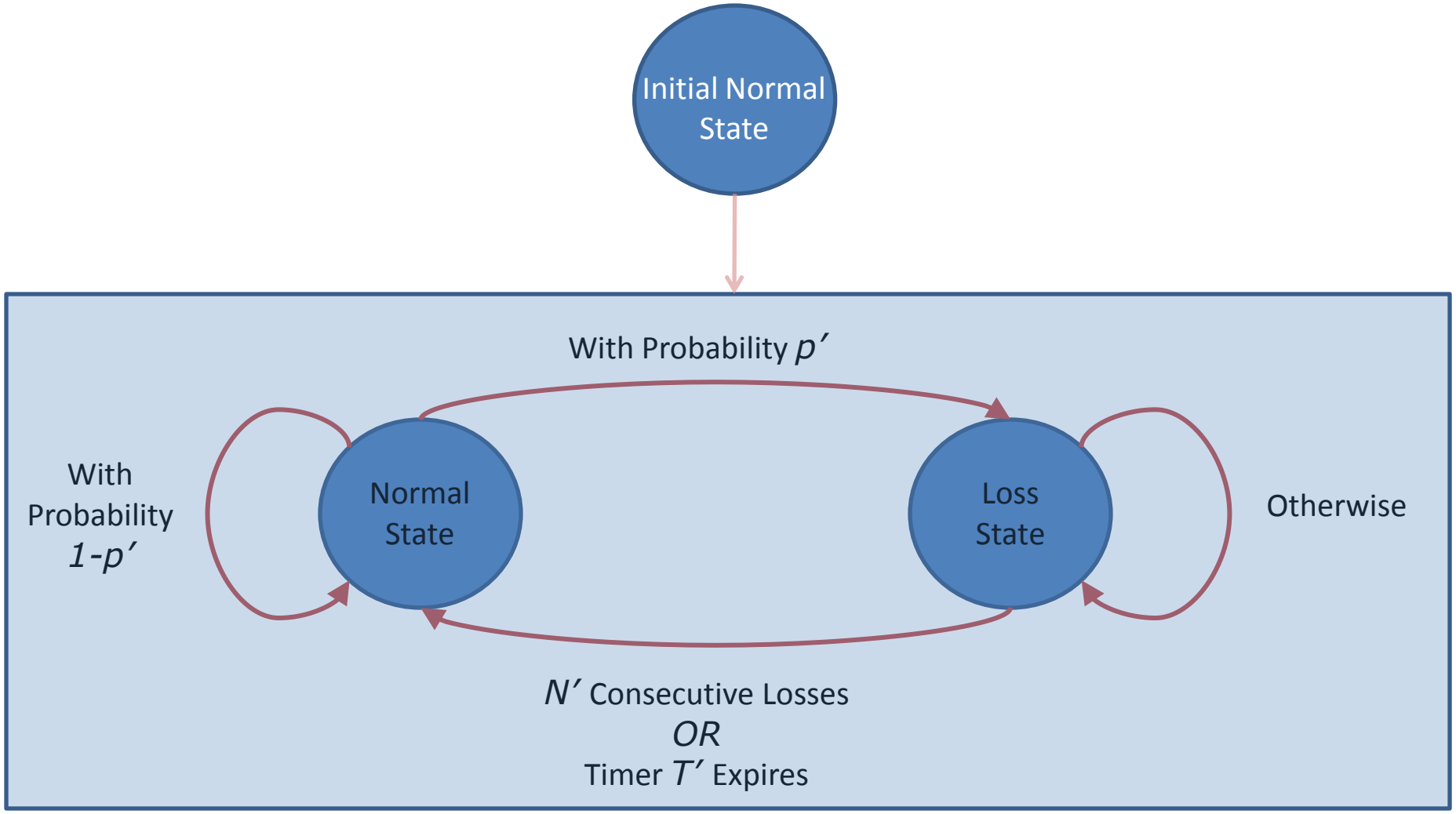


Why not the UDP design for TCP?

- HTTP
- SSH
- FTP

p (%)	.jp
0	9min
35	112min
5	87min
1	60min

Building TCP Model: HTTP



Experiments

- VoIP
- HTTP
- FTP
- SSH

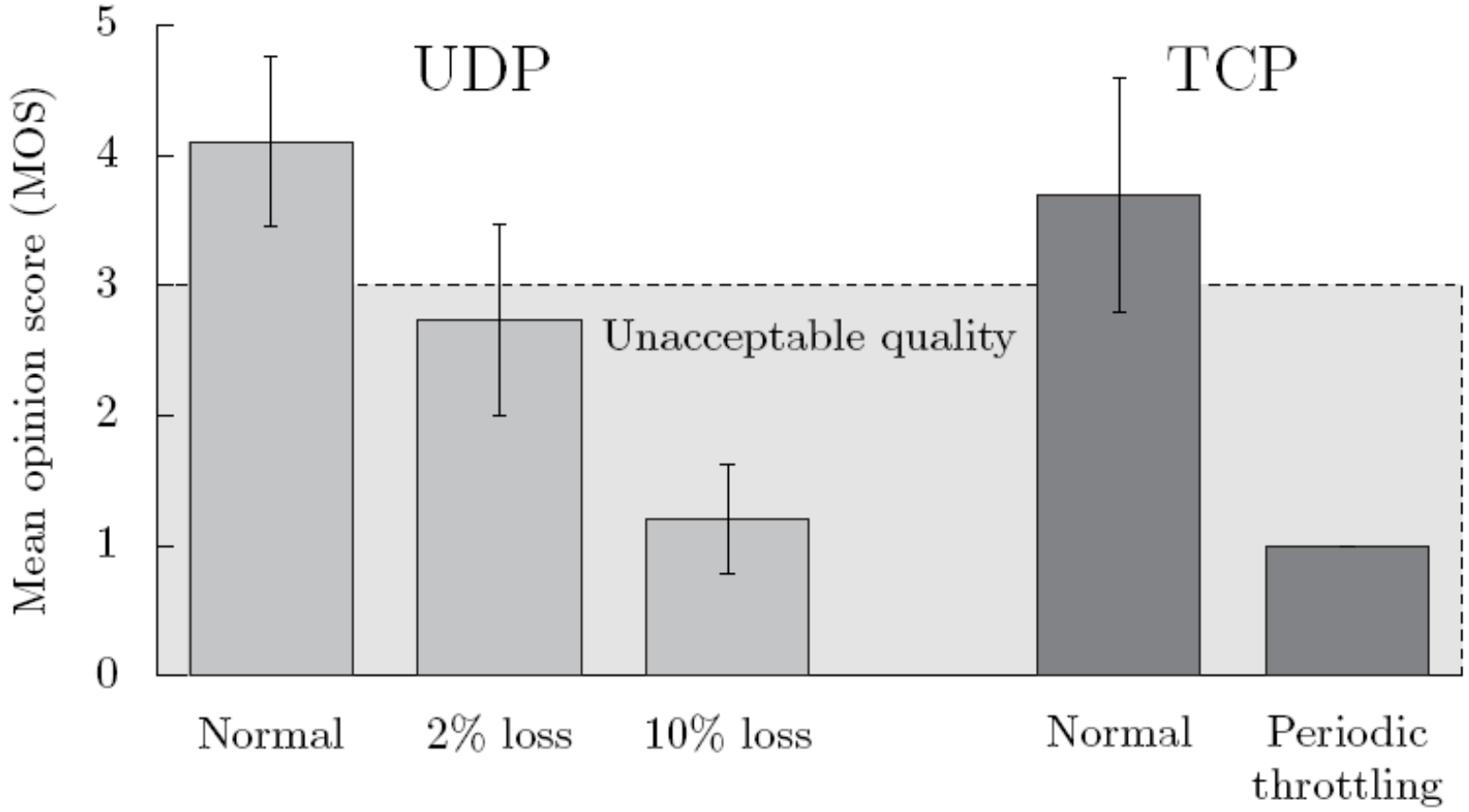
Mean Opinion Score

MOS	Quality	Impairment
5	Excellent	Imperceptible
4	Good	Perceptible but not annoying
3	Fair	Slightly annoying
2	Poor	Annoying
1	Bad	Very annoying

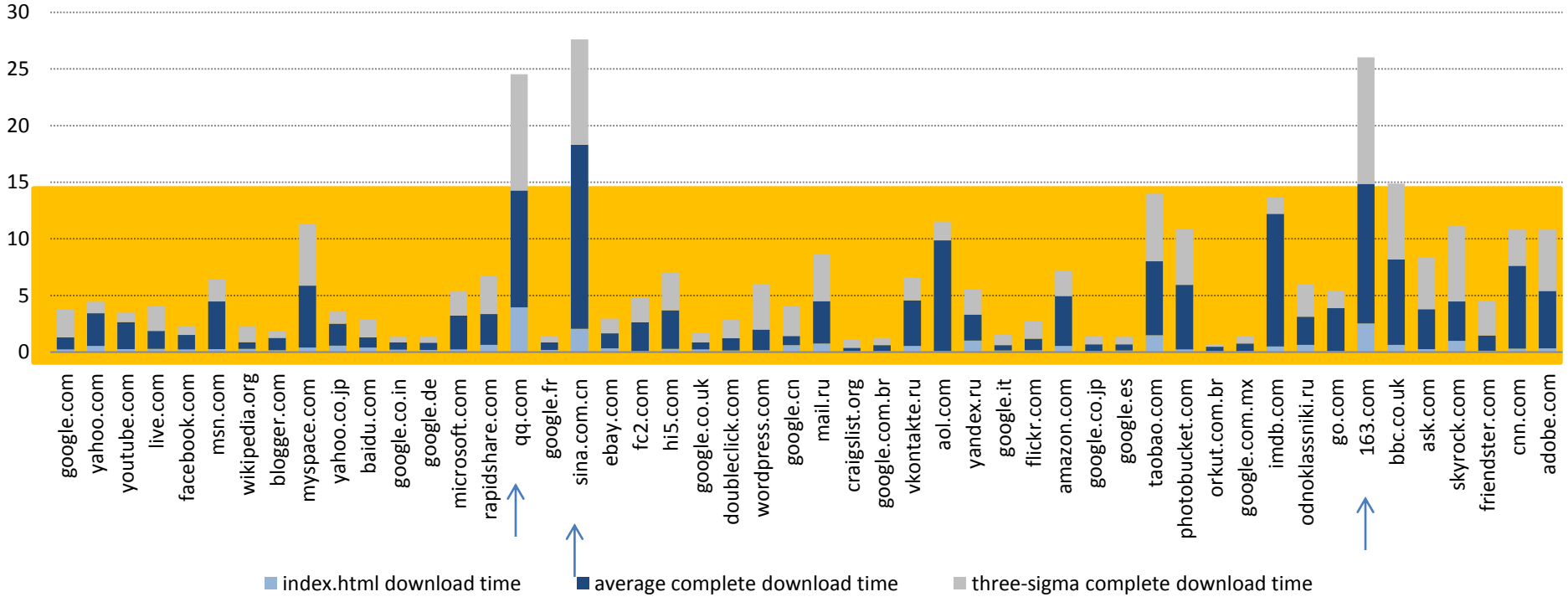
≥4.0: Public Switched Telephone Network (PSTN) Standard

Source: ITU-T P.800

Experiments: VoIP



Experiments: Web



Did It Really Work?

- Did our chosen experiment work?
  - Succeeded for our set of applications
  - QoS metrics are debatable
- Did we validate the concept of network dissuasion?

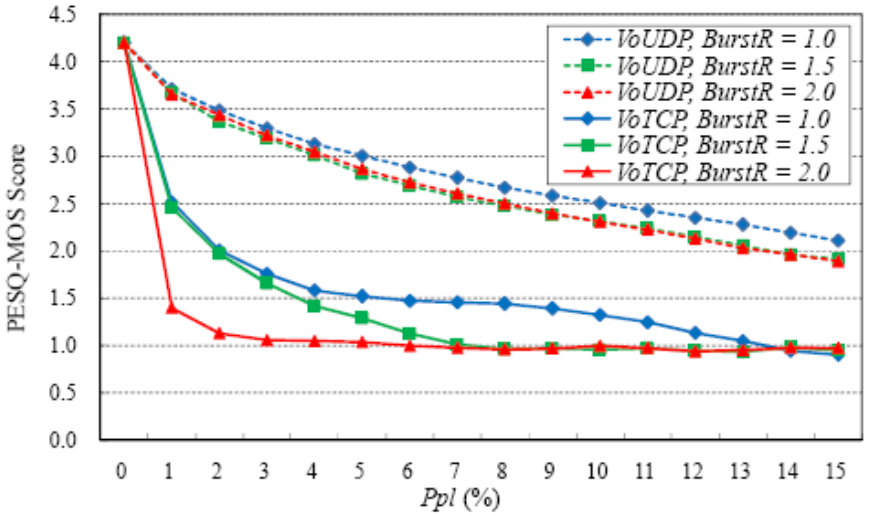
  Yes, but:
  - Shutting down VoIP was not easy
  - Generalizing it to an arbitrary set of applications maybe difficult

Thank You

Questions?

Voice Over TCP (VoTCP)

- Skype: Only when UDP doesn't work
- Highly Sensitive to Losses
  - Burst Ratio of 1.0, Loss rate of 1.6% MOS < 3.0



Design Assumptions

- No UDP/IP Spoofing
- All "Permitted" Applications Use TCP (Except DNS)

Application	Protocol	RFC
HTTP	Not UDP (usually TCP)	2616
Telnet	TCP	854
SSH	Not UDP (usually TCP)	4251
FTP	TCP	959
DNS	TCP/UDP (mainly UDP)	1035

Evaluation Methodology

- **Subjective**
  - Mean Opinion Score
- **Objective**
  - PSQM (ITU-T P.861)
  - Measuring Normalized Blocks PESQ (ITU-T P.862)
  - Perceptual Evaluation of Speech Quality PAMS
  - Perceptual Analysis Measurement System
  - The E-Model (ITU-T G.107)

Overview

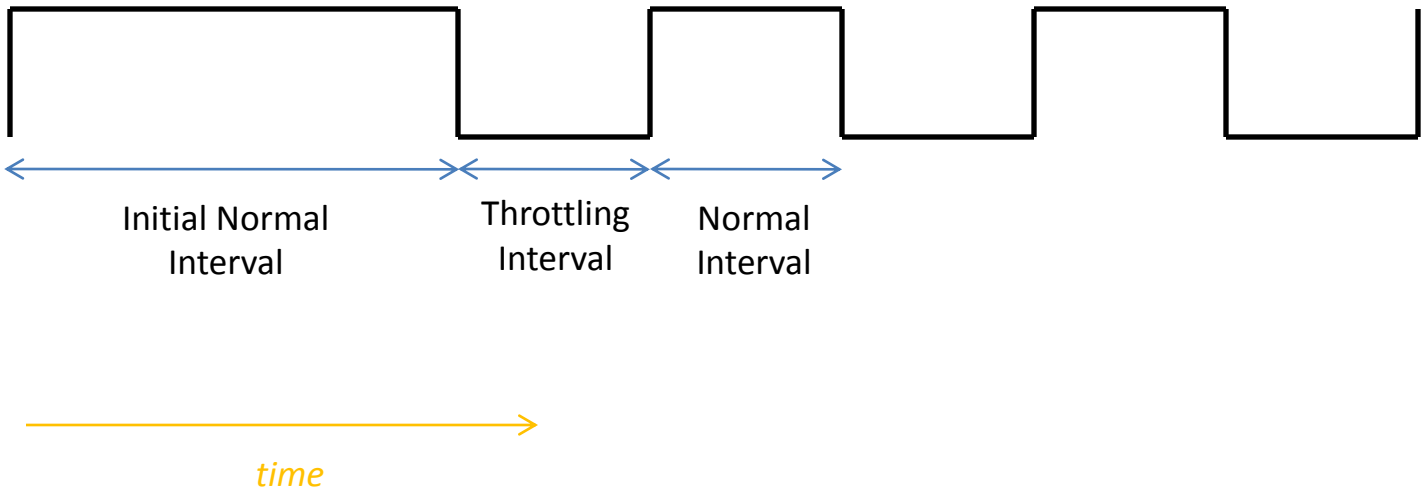
- UDP Design
- TCP Design
- Experiments
- Discussion

Permitted Applications: QoS Requirements

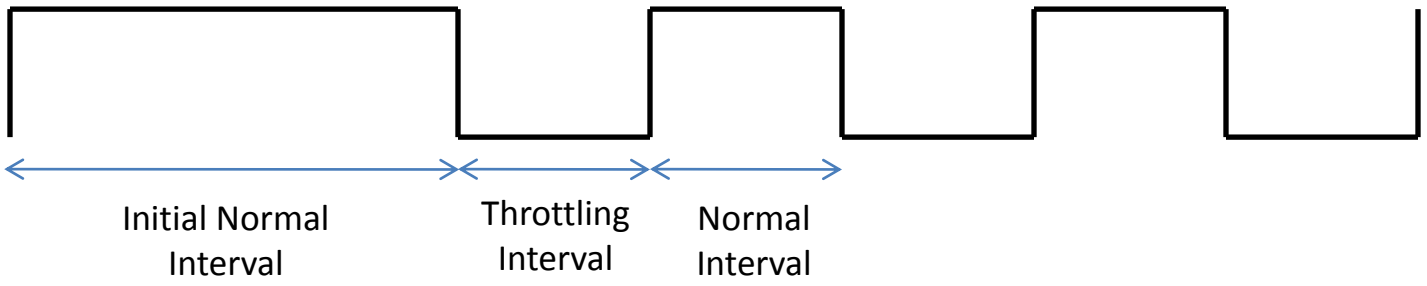
Category	One-way delay	Req/resp delay	Loss	Dur.	Jitter
email (srv/srv)		whole, RTT $<$ 4 h			
Usenet		whole, RTT $<$ 4 h			
Chat, typing		RTT $<$ 4 s			
Chat, audio	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Chat, video	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		
Web		part, RTT $<$ 4 s		$<$ 60 s	
FTP Data		part, RTT $<$ 10 s		$<$ 300%	
FTP Control		part, RTT $<$ 4 s			
FPS games	$<$ 150 ms		$<$ 3 %		
RTS games	$<$ 500 ms				
Telnet		part, RTT $<$ 250 ms			
email (usr/srv)		part, RTT $<$ 4 s		$<$ 300%	
DNS		whole $<$ 4 s			
Ping		whole $<$ 4 s			
	media	control	media		media
Audio, conv.	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Audio, messg.	$<$ 2 s	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Audio, stream	$<$ 10 s	whole, RTT $<$ 4 s	$<$ 1%		$<$ 50 ms
Videophone	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		
Video, stream	$<$ 10 s	whole, RTT $<$ 4 s	$<$ 1%		

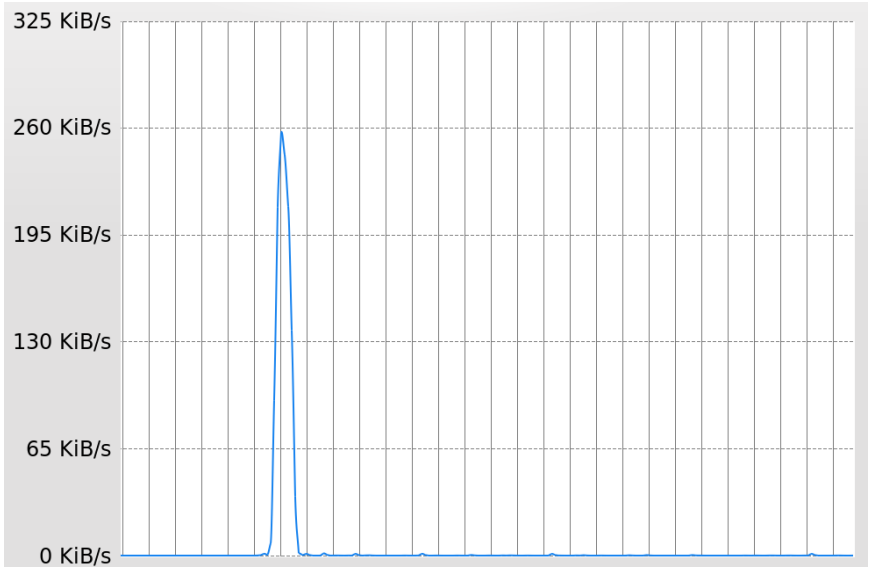
Mirkovic et al.

Building TCP Model: FTP



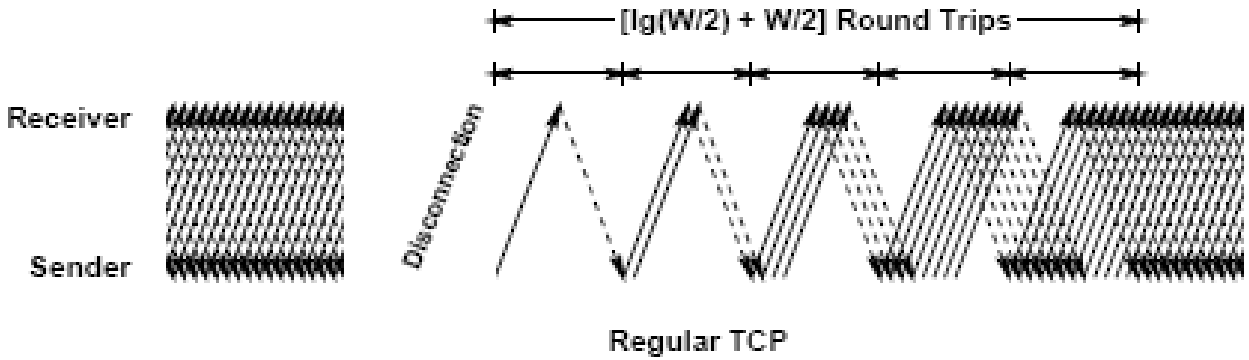
Building TCP Model: FTP?



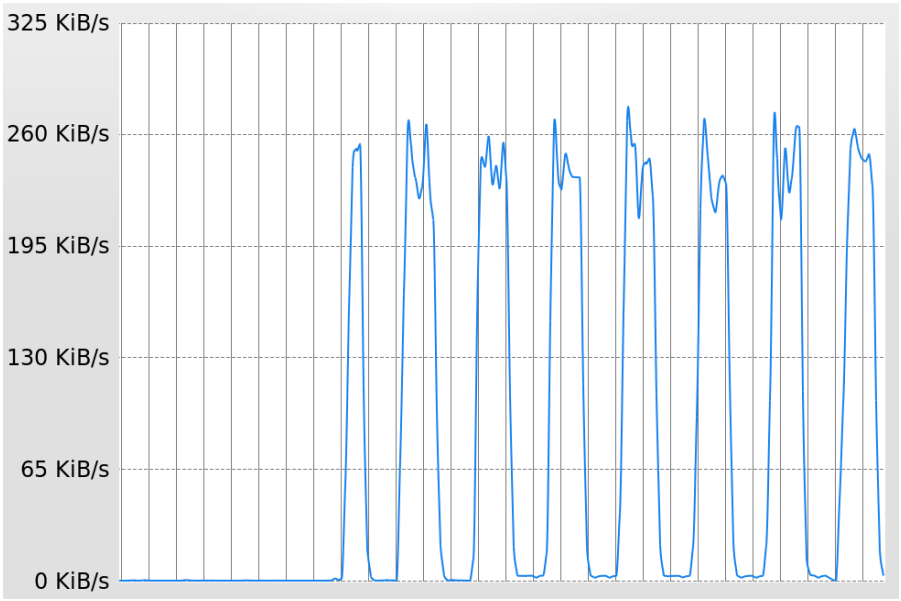


mirrors.vcu.edu

Freeze-TCP


[lg(W/2) + W/2] Round Trips
Receiver
Sender
Disconnection
Regular TCP
Receiver
Sender
Disconnection
Freeze-TCP

FTP Improvement



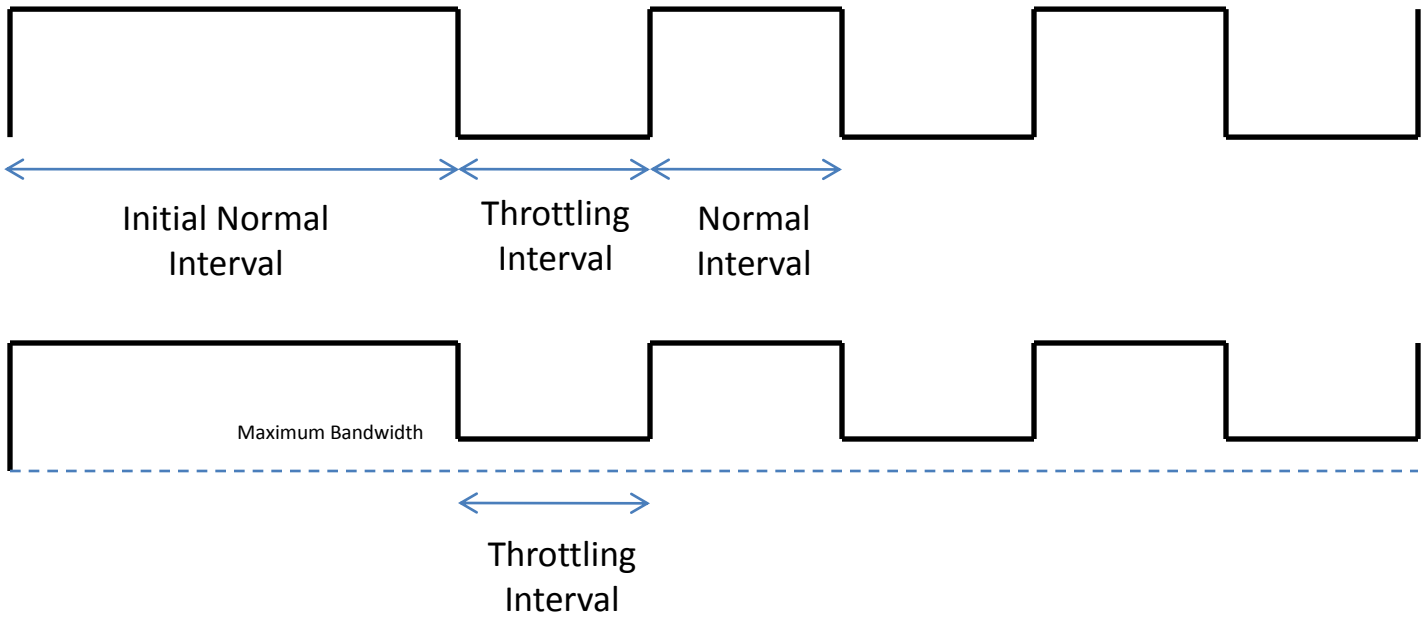
mirrors.vcu.edu

Permitted Applications: QoS Requirements

Category	One-way delay	Req/resp delay	Loss	Dur.	Jitter
email (srv/srv)		whole, RTT $<$ 4 h			
Usenet		whole, RTT $<$ 4 h			
Chat, typing		RTT $<$ 4 s			
Chat, audio	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Chat, video	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		
Web		part, RTT $<$ 4 s		$<$ 60 s	
FTP Data		part, RTT $<$ 10 s		$<$ 300%	
FTP Control		part, RTT $<$ 4 s			
FPS games	$<$ 150 ms		$<$ 3 %		
RTS games	$<$ 500 ms				
Telnet		part, RTT $<$ 250 ms			
email (usr/srv)		part, RTT $<$ 4 s		$<$ 300%	
DNS		whole $<$ 4 s			
Ping		whole $<$ 4 s			
	media	control	media		media
Audio, conv.	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Audio, messg.	$<$ 2 s	whole, RTT $<$ 4 s	$<$ 3%		$<$ 50 ms
Audio, stream	$<$ 10 s	whole, RTT $<$ 4 s	$<$ 1%		$<$ 50 ms
Videophone	$<$ 150 ms	whole, RTT $<$ 4 s	$<$ 3%		
Video, stream	$<$ 10 s	whole, RTT $<$ 4 s	$<$ 1%		

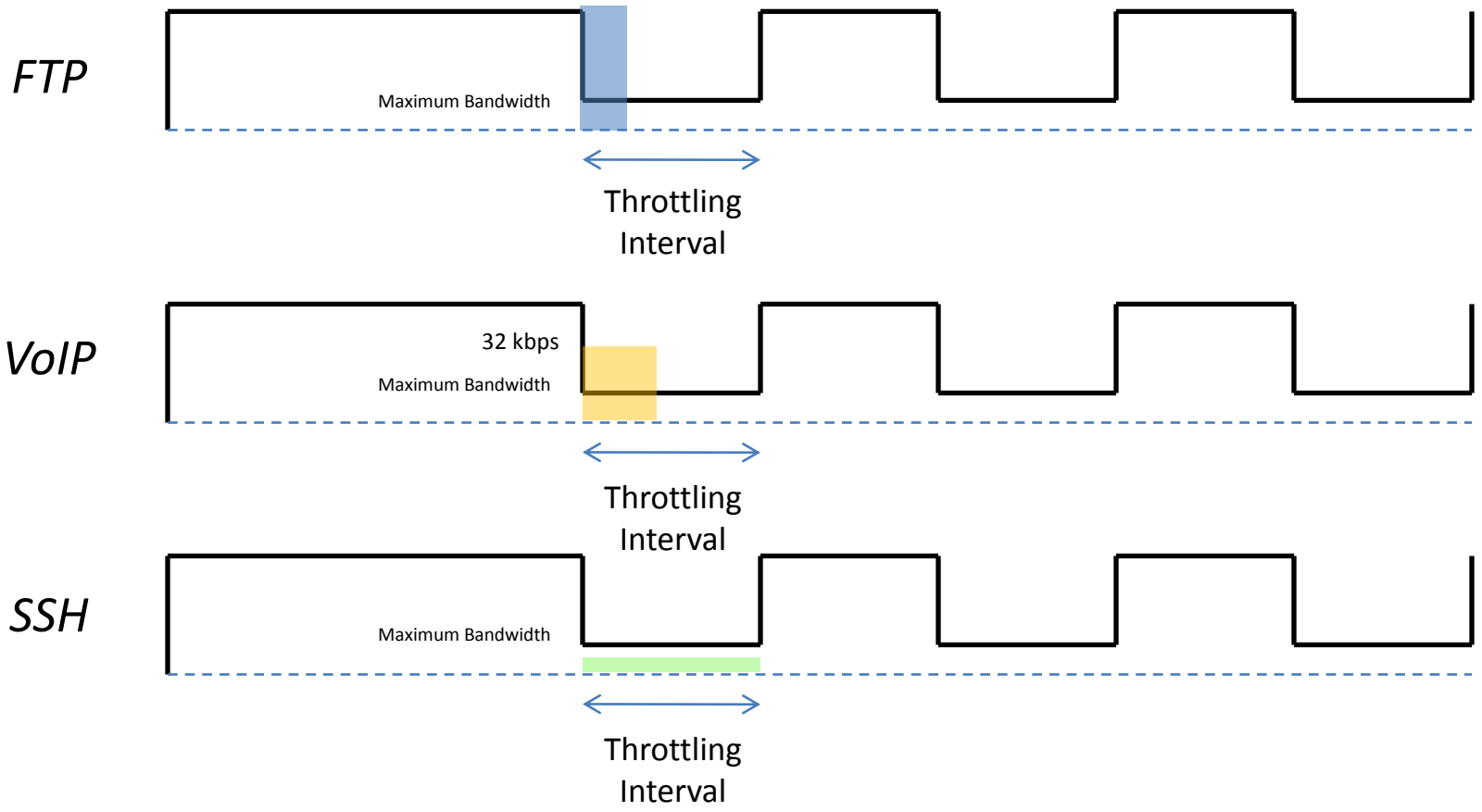
Mirkovic et al.

Building TCP Model: SSH



Quota = Throttling Interval x Maximum Bandwidth

Building TCP Model: SSH



Experiments: FTP

- File Size
- RTT
- Bandwidth
- Congestion Control Mechanism

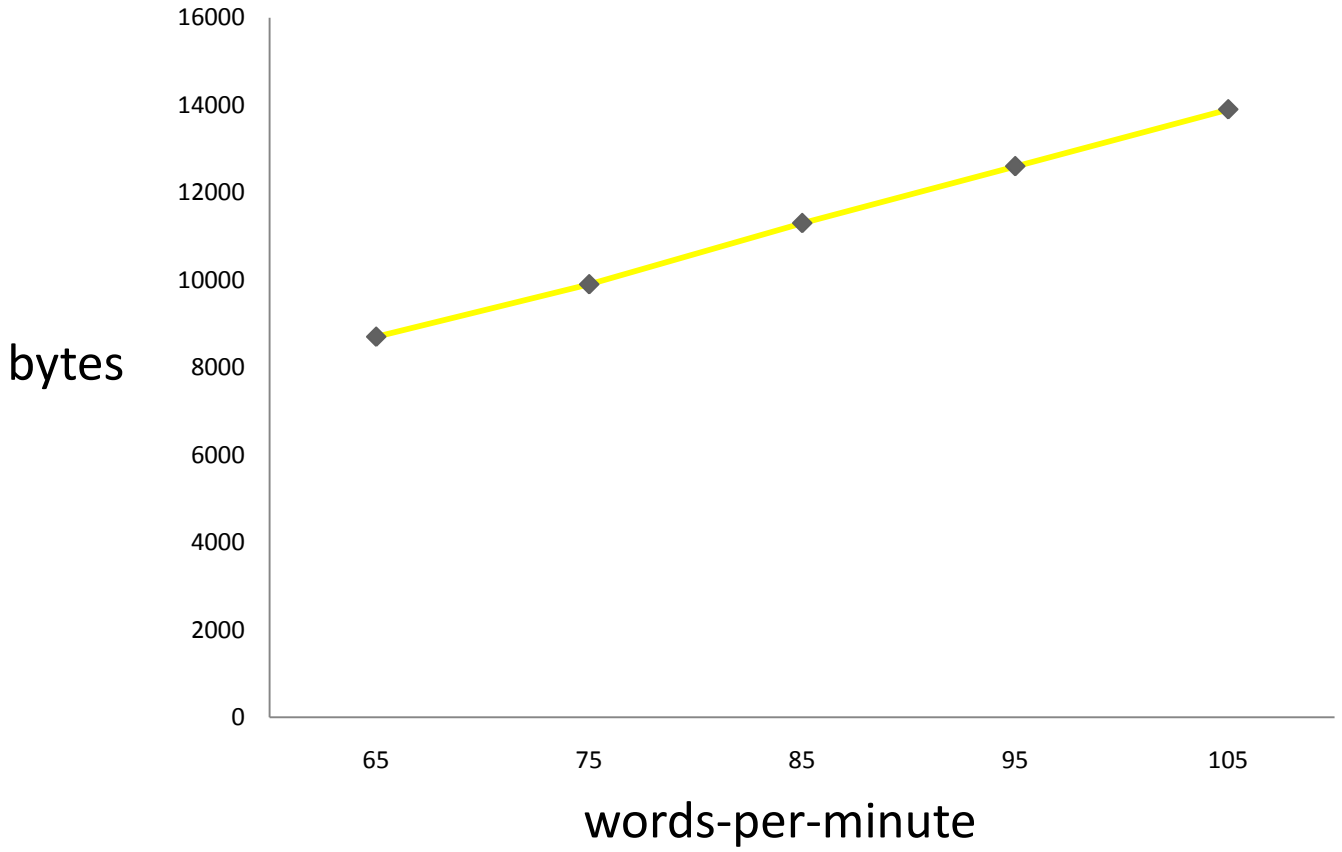
Experiments: FTP

- Large File (~700MB)
- Long RTT (~200ms)
- High Bandwidth (100kbps)
- Congestion Control Mechanism

Popular TCP variants: CUBIC and Compound TCP

TCP Variant	Normal (s)	Dissuade (s)	Accepted Delay (s)
C-TCP	683	2006	2049
Cubic	660	1994	1980

Experimentation Plan: SSH



Quota = 16 (s) x 10 (kbps) = 20000 bytes

Experiments: Web

Website: Search

Today's Most Popular Websites on the Internet:

Ads by Google | Most Visited | 2009 Auto | Renta 2009 | Leilão 2009

The following list of the **Most Popular Websites** was updated on 2010-04-23 .

1	Google.com	www.google.com
2	Yahoo.com	www.yahoo.com
3	Youtube.com	www.youtube.com
4	Live.com	www.live.com
5	Facebook.com	www.facebook.com
6	Msn.com	www.msn.com
7	Wikipedia.org	www.wikipedia.org
8	Blogger.com	www.blogger.com
9	Myspace.com	www.myspace.com
10	Yahoo.co.jp	www.yahoo.co.jp
11	Baidu.com	www.baidu.com
12	Google.co.in	www.google.co.in
13	Google.de	www.google.de
14	Microsoft.com	www.microsoft.com