TCP-Probleme und -Tuning im WAN

Peter Serocka
CAS-MPG Partner Institute
for Computational Biology, Shanghai

Bielefeld, 5. Januar 2010

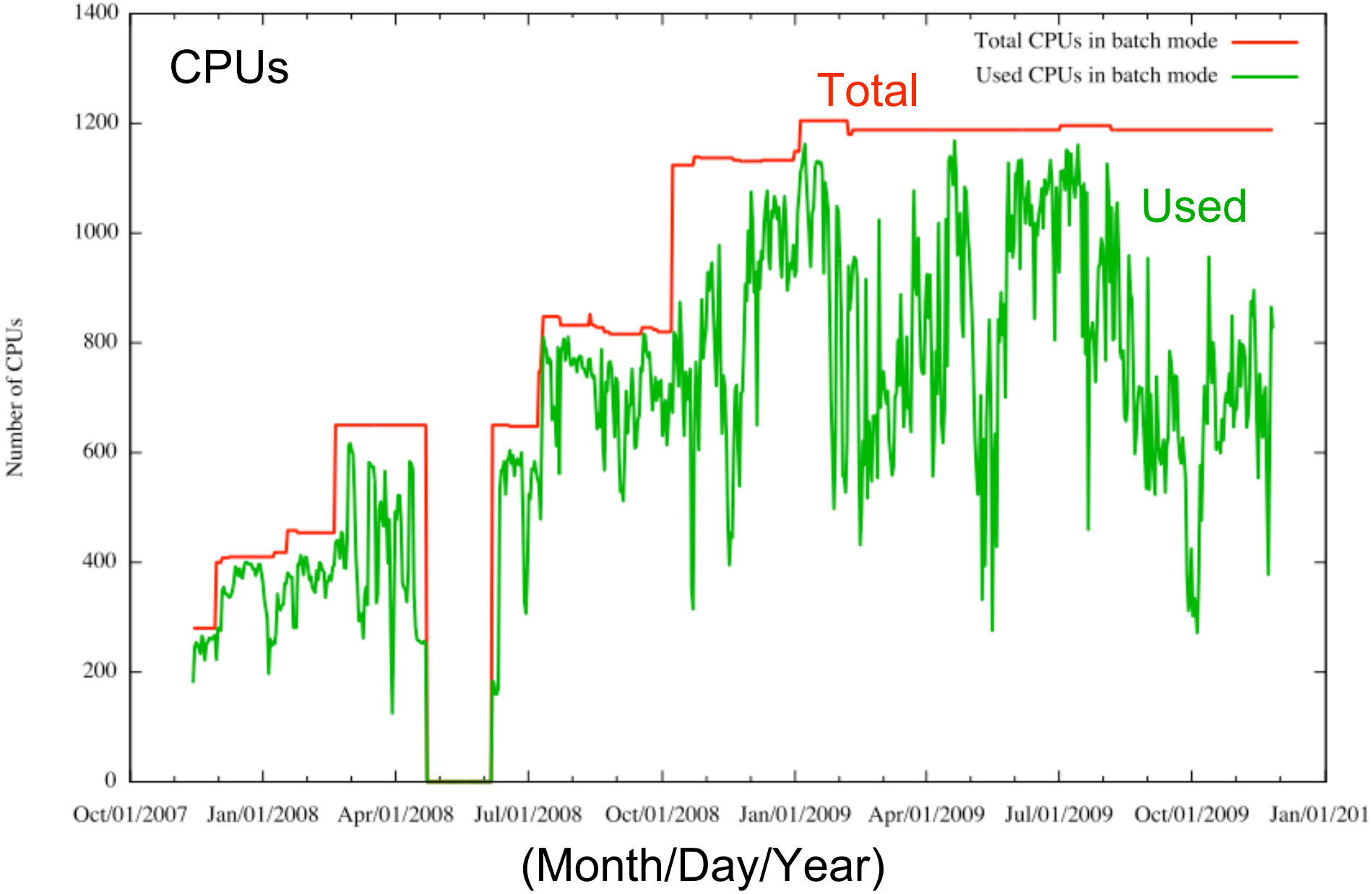
PICB

Chinese
Academy (CAS) &
Max-Planck (MPG)

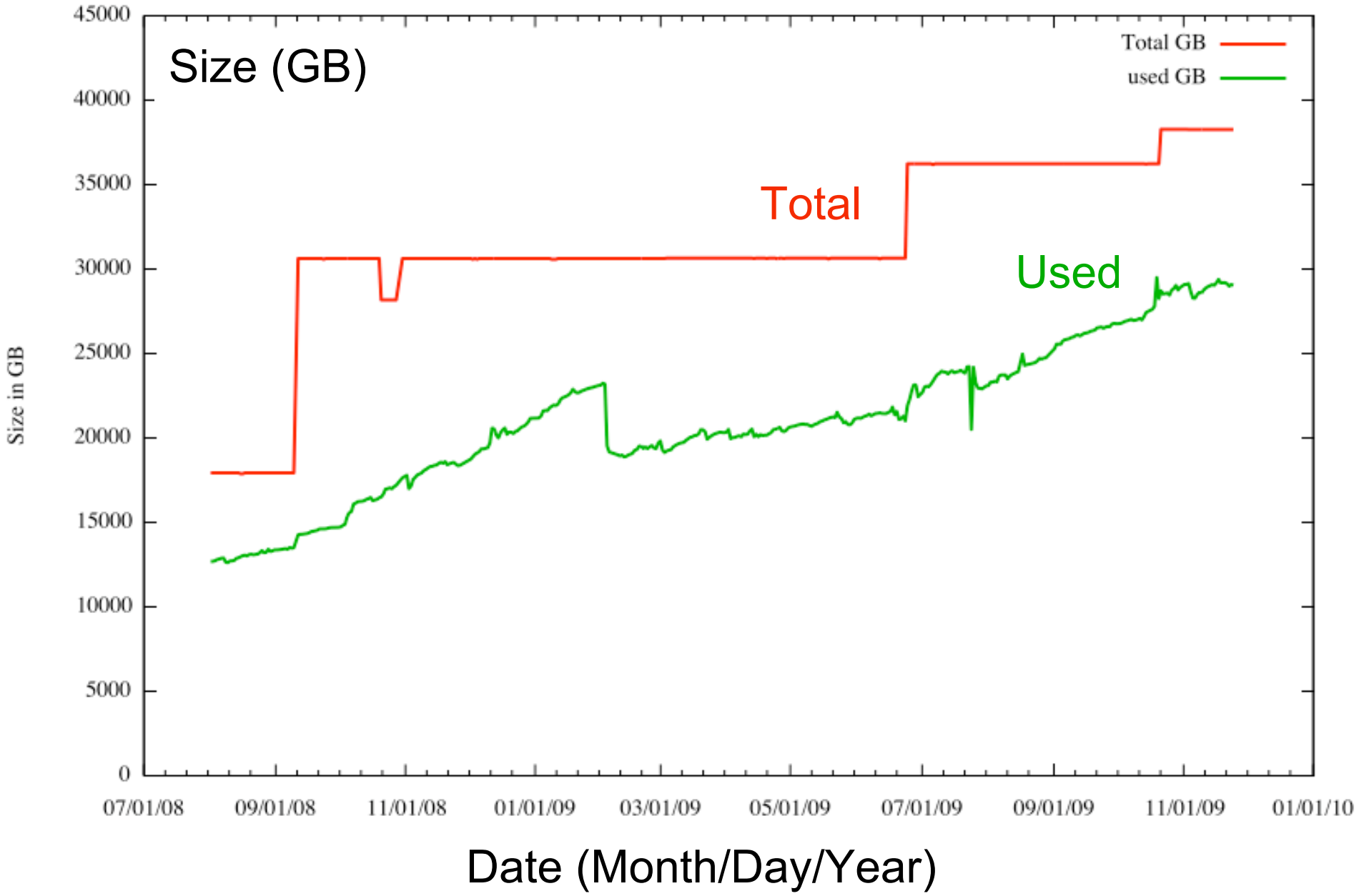
Partner
Institute for
Computational
Biology

Shanghai

PICB Cluster: Batch Usage (Nov 2007 - Nov 2009)



PICB Network Storage (Aug 2008 - Nov 2009)



Users‘ View of Data Transfer

```
(somewhere at Uni Bielefeld:)

$ scp data.dat gate1.picb.ac.cn:
data.dat                          1% 8656KB 288.1KB/s   39:43 ETA



$ ping www.picb.ac.cn
PING www.picb.ac.cn (202.127.25.195) 56(84) bytes of data.
64 bytes from www.picb.ac.cn (202.127.25.195):
icmp_seq=1 ttl=47 time=304 ms
```

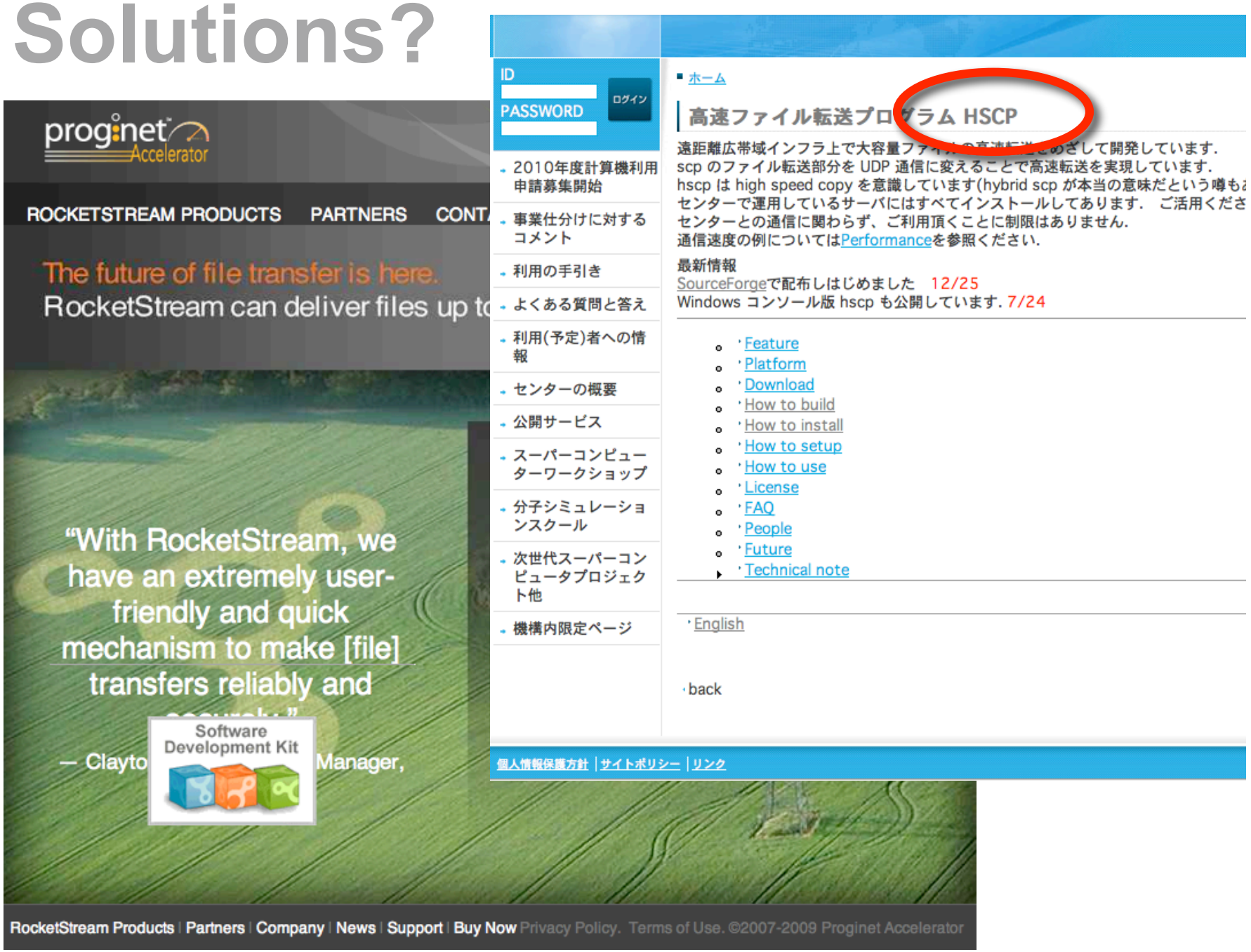
Route Analysis

```
traceroute to gate1.picb.ac.cn (202.127.22.195), 30 hops max, 40 byte packets
 1  s01-cat6500-1.hrz.uni-bielefeld.de (129.70.15.253)  0.342 ms  0.463 ms  0.537 ms
 2  s01-cat6500-2.if.hrz.uni-bielefeld.de (129.70.188.69)  0.435 ms  0.430 ms  0.506 ms
 3  v01-cat6500-3.if.hrz.uni-bielefeld.de (129.70.188.65)  0.493 ms  0.594 ms  0.586 ms
 4  v01-cat6500-2.if.hrz.uni-bielefeld.de (129.70.188.97)  0.739 ms  0.881 ms  0.866 ms
 5  xr-bie1-ge8-3.x-win.dfn.de (188.1.86.209)  1.203 ms  1.345 ms  1.034 ms
 6  zr-han1-te0-7-0-6.x-win.dfn.de (188.1.145.9)  3.114 ms  2.891 ms  2.851 ms
 7  zr-fra1-te0-0-0-4.x-win.dfn.de (188.1.145.213)  15.629 ms  15.431 ms  15.418 ms
 8  dfn.rt1.fra.de.geant2.net (62.40.124.33)  14.622 ms  14.613 ms  14.585 ms
 9  so-6-0-0.rt2.cop.dk.geant2.net (62.40.112.50)  28.732 ms  28.810 ms  28.722 ms
10  bj-so-01-v4.bb.tein3.net (202.179.241.41)  204.529 ms  204.626 ms  204.515 ms
11  cn-pr-2-v4.bb.tein3.net (202.179.241.51)  204.469 ms  204.545 ms  204.452 ms
12  202.112.53.17 (202.112.53.17)  204.422 ms  204.342 ms  204.334 ms
13  202.112.61.157 (202.112.61.157)  204.949 ms  204.713 ms  204.705 ms
14  202.112.53.178 (202.112.53.178)  204.963 ms  204.319 ms  204.311 ms
15  210.25.128.6 (210.25.128.6)  204.986 ms  204.434 ms  204.430 ms
16  210.25.129.14 (210.25.129.14)  206.985 ms  206.976 ms  206.966 ms
17  210.25.128.66 (210.25.128.66)  206.954 ms  207.147 ms  206.963 ms
18  8.201 (159.226.254.25)  281.218 ms  280.996 ms  280.974 ms
19  8.208 (159.226.254.110)  281.381 ms  281.374 ms  281.356 ms
20  * * *
21  ACA80012.ipt.aol.com (172.168.0.18)  306.837 ms  304.278 ms  304.377 ms
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Solutions?


prog:net Accelerator
ROCKETSTREAM PRODUCTS
PARTNERS
CONTACT US
NEWS
SUPPORT
BUY NOW
RocketStream can deliver files up to 200x faster
RocketStream is a software-based data acceleration solution that makes file transfers fast, easy, secure, and reliable. RocketStream is the ideal way to transfer large files over IP networks. If network latency is slowing you down, just RocketStream it!
RocketStream Hot Topics
Jump straight to our News Section »
Proginet Announces the Acquisition of the RocketStream Software Suite - 12.22.2009
RocketStream Releases SDK for Ultra-Fast File Transfer Capability - 05.11.2009
Software Development Kit
RocketStream Products | Partners | Company | News | Support | Buy Now Privacy Policy. Terms of Use. ©2007-2009 Proginet Accelerator

Solutions?


ID
PASSWORD
ログイン
ホーム
高速ファイル転送プログラム HSCP
遠距離広帯域インフラ上で大容量ファイルの高速転送をめざして開発しています.
scp のファイル転送部分を UDP 通信に変えることで高速転送を実現しています.
hscp は high speed copy を意識しています(hybrid scp が本当の意味だという噂も
センターで運用しているサーバにはすべてインストールしてあります. ご活用くださ
センターとの通信に関わらず、ご利用頂くことに制限はありません.
通信速度の例については Performance を参照ください.
最新情報
SourceForge で配布しはじめました 12/25
Windows コンソール版 hscp も公開しています. 7/24
Feature
Platform
Download
How to build
How to install
How to setup
How to use
License
FAQ
People
Future
Technical note
English
back
2010年度計算機利用申請募集開始
事業仕分けに対するコメント
利用の手引き
よくある質問と答え
利用(予定)者への情報
センターの概要
公開サービス
スーパーコンピューターワークショップ
分子シミュレーションスクール
次世代スーパーコンピュータプロジェクト他
機構内限定ページ
個人情報保護方針 | サイトポリシー | リンク
prog:net Accelerator
ROCKETSTREAM PRODUCTS
PARTNERS
The future of file transfer is here.
RocketStream can deliver files up
"With RocketStream, we have an extremely user-friendly and quick mechanism to make [file] transfers reliably and
Software Development Kit
Manager,
RocketStream Products | Partners | Company | News | Support | Buy Now Privacy Policy. Terms of Use. ©2007-2009 Proginet Accelerator

Understanding the Principles

- BDP = **B**andwidth x **D**elay **P**roduct
- Congestion Avoidance

Bandwidth Delay Product

BDP: data-in-flight before ACK

-> window buffer ~ 2 * BDP

- 10 Mbit/s * 10 ms * 2 = 15.0 KByte
- 100 Mbit/s * 1 ms * 2 = 15.0 Kbyte
- 40 Mbit/s * 300 ms * 2 = 3.0 Mbyte
- 1 Gbit/s * 10 ms * 2 = 2.5 Mbyte
- 10 Gbit/s * 1 ms * 2 = 2.5 Mbyte
- 10 Gbit/s * 50 ms * 2 = 125.0 Mbyte

traditional buffer size for TCP: 64 KByte

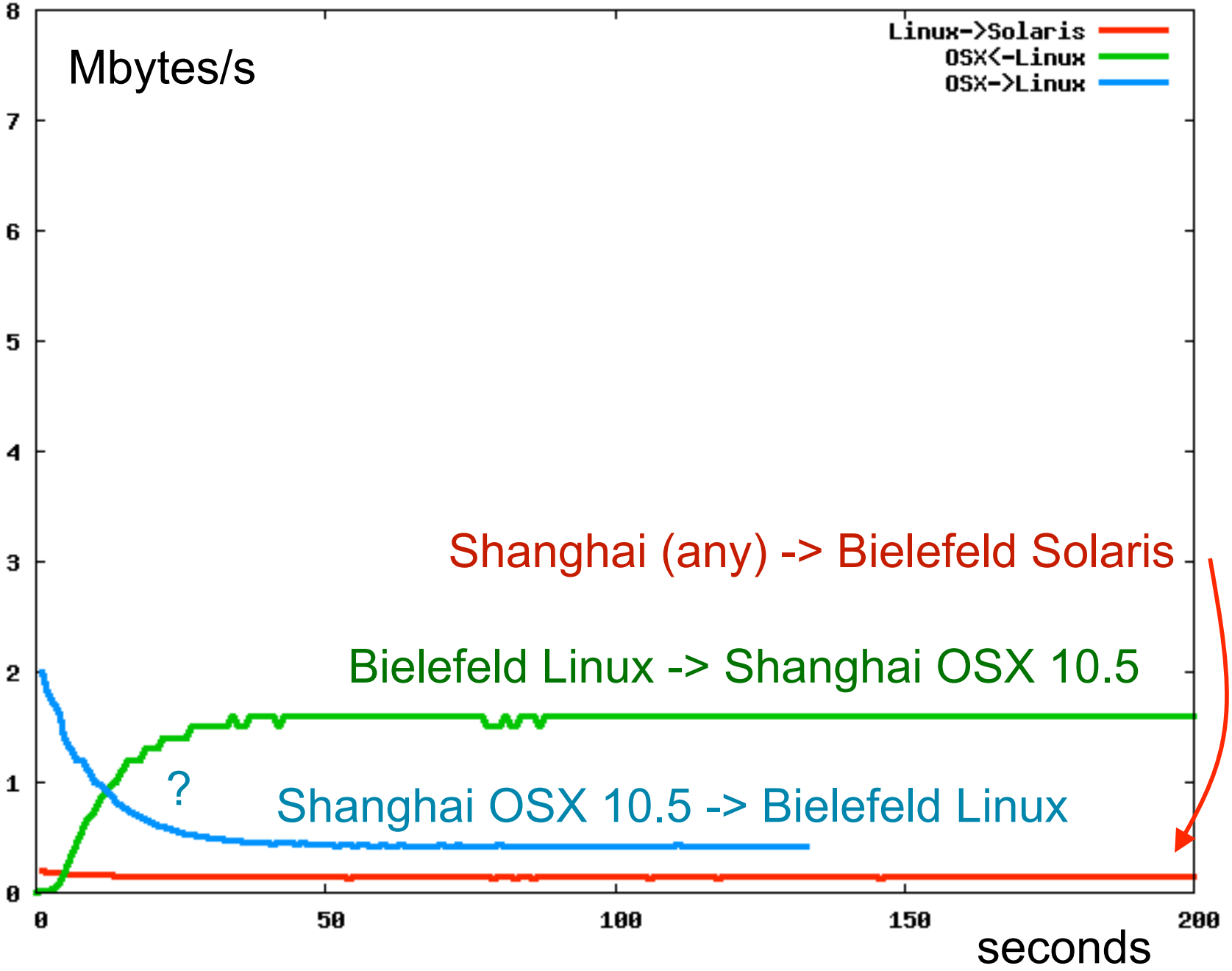
TCP/IP congestion avoidance

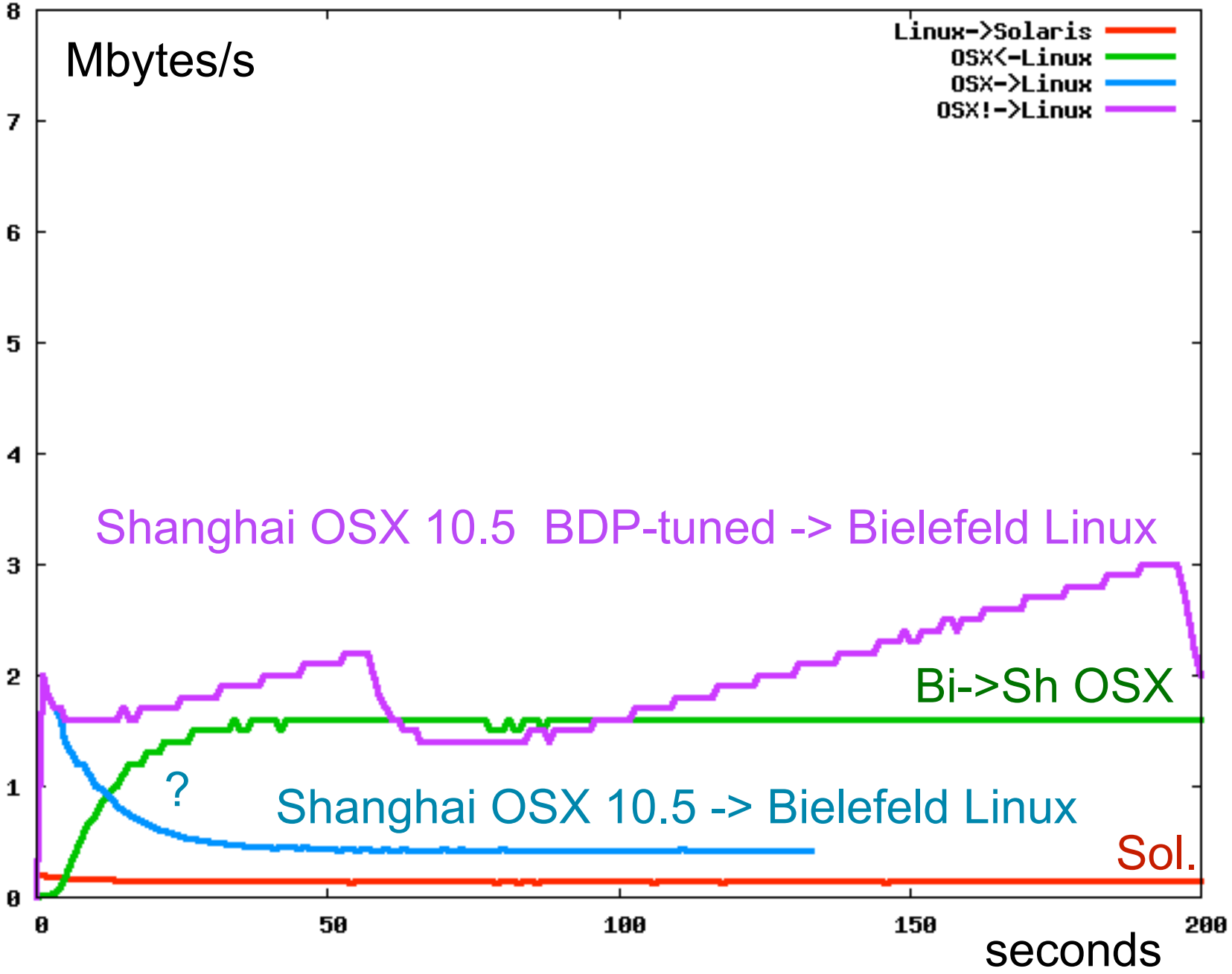
Active Queue Management (AQM)

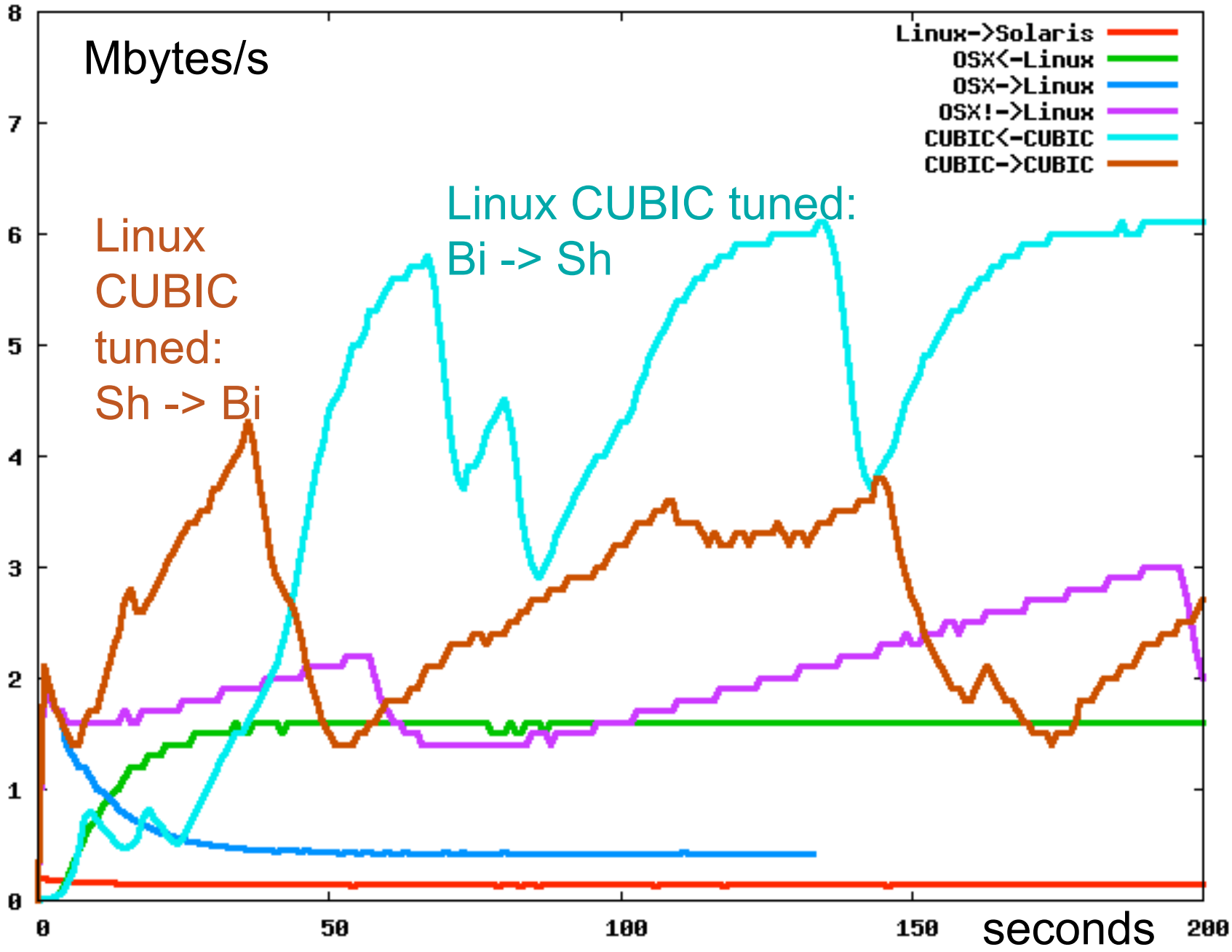
- Random early detection (RED)
- Flowbased-RED (QoS)
- IP ECN bit, Explicit Congestion Notification
- Cisco AQM: Dynamic buffer limiting (DBL)
- TCP Window Shaping -- state-of-the art: CUBIC TCP

CUBIC Test Shanghai <-> Bi

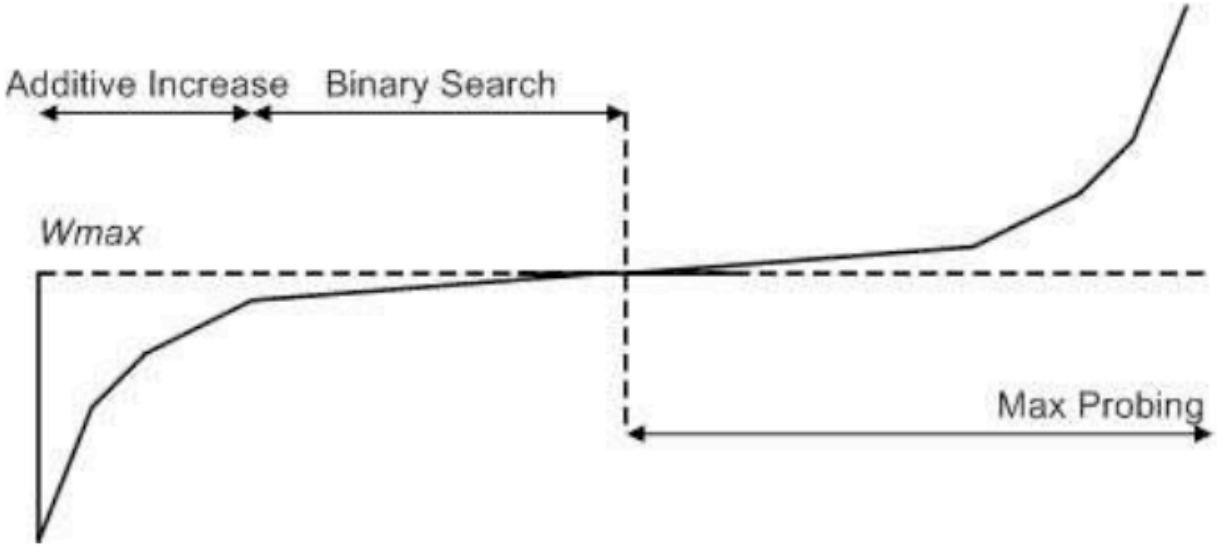
```
$ uname  -s -r
Linux 2.6.28-11-generic


$ sysctl -a
…
net.ipv4.tcp_wmem = 4096        16384 3969024
net.ipv4.tcp_rmem =  4096       87380 3969024
…
net.ipv4.tcp_congestion_control = cubic
net.ipv4.tcp_available_congestion_control = cubic reno
…
```


Mbytes/s
Linux->Solaris
OSX<-Linux
OSX->Linux
8
7
6
5
4
3
2
1
0
0
50
100
150
200
seconds
Shanghai (any) -> Bielefeld Solaris
Bielefeld Linux -> Shanghai OSX 10.5
?
Shanghai OSX 10.5 -> Bielefeld Linux

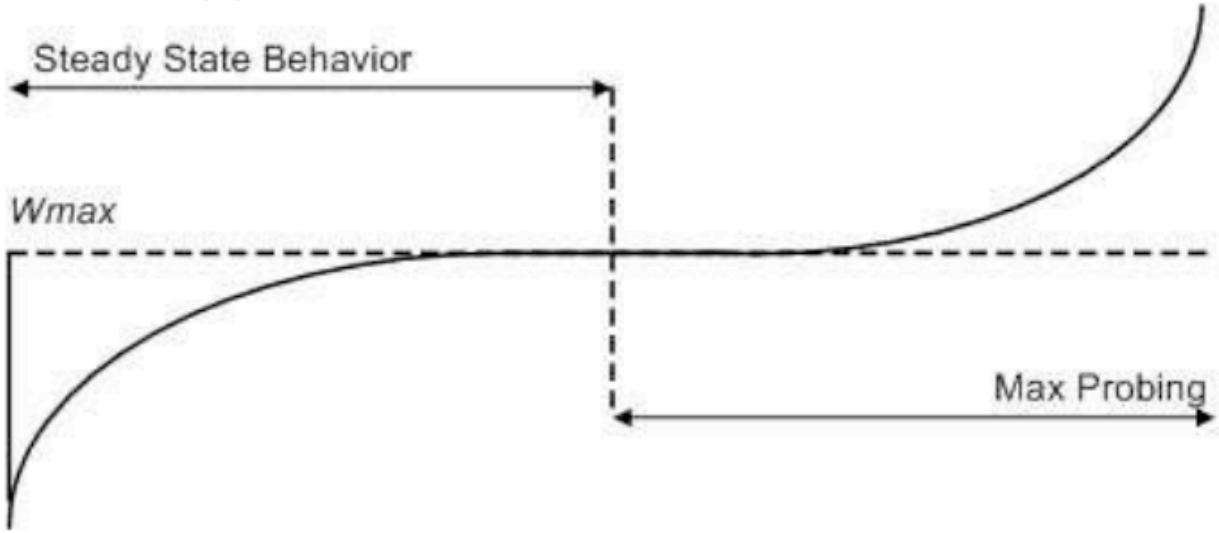

Mbytes/s
Linux->Solaris
OSX<-Linux
OSX->Linux
OSX!->Linux
Shanghai OSX 10.5 BDP-tuned -> Bielefeld Linux
Bi->Sh OSX
?
Shanghai OSX 10.5 -> Bielefeld Linux
Sol.
seconds
0
1
2
3
4
5
6
7
8
0
50
100
150
200


Mbytes/s
Linux->Solaris
OSX<-Linux
OSX->Linux
OSX!->Linux
CUBIC<-CUBIC
CUBIC->CUBIC
Linux CUBIC tuned: Bi -> Sh
Linux CUBIC tuned: Sh -> Bi
0
1
2
3
4
5
6
7
8
0
50
100
150
200
seconds

BIC/ CUBIC Window growth



(a) BIC-TCP window growth function.



(b) CUBIC window growth function.

Summary (pragmatic)

- Speedup for scp between Bielefeld <-> Shanghai: from **0.3** MByte/s to **3-5** Mbyte/s at no cost.
- How-to:
  - ☞ Check the **BDP** (bandwidth delay product) on your path.
  - ☞ Allow for **buffers sizes** at least twice the BDP.
  - ☞ Let state-of-the-art **CUBIC TCP** do all the dirty work.
- TCP algorithms are (still) under active **development;** testbed implementations run on Linux…

Acknowledgements & Refs:

- DFN
- MPI f. Mathematatik I.d. Naturwissenschaften, Leipzig

- Brian L. Tierney, *http://fasterdata.es.net/TCP-tuning/TCP-Tuning-Tutorial.pdf*
- Sangtae Ha, Injong Rhee, Lisong Xu: *draft-rhee-tcpm-cubic-02.txt*