CESNET Technical Report 25/2003

End-to-end Bandwidth Estimation Tools

Sven Ubik, ubik@cesnet.cz
Antonín Král, A.Kral@sh.cvut.cz

November 28, 2003

Abstract

Speed of data transfer over a wide-area network can be limited by any of the routers or links along a network path. If the bottleneck router or link is located beyond our administrative scope, we cannot find where it is and how small it is by checking link utilisation by connecting to routers and reading its byte statistics (either by login and CLI commands or using SNMP). In such a case we can use bandwisth measurement or estimation tools run between the end points to find the size and possibly the location of the bottleneck.

In this report we summarize our experience with *bandwidth estimation tools*, which can estimate installed or available bandwidth along a network path without access to any component along that network path and without stressing existing traffic with a large volume of testing traffic.

1 What are bandwidth estimation tools?

Installed bandwidth along a network path and even more importantly available bandwidth, that is a part of the installed bandwidth, which is currently not used by traffic, are very important characteristics of network potential performance for any new traffic submitted to the network.

Available bandwidth can be monitored by several techniques:

- Reading router interface byte counters read by CLI commands or SNMP (which allows us to compute current load, which is a complement to available bandwidth)
- Passive monitoring of traffic tapped with optical splitter or mirroring switch (dtto)
- Using a bandwidth measurement tool
- Using a bandwidth estimation tool

The advantage of the last two options is that they do not require any access to routers or network links along a path. The disadvantage is that they are generally less accurate than the first two methods.

Bandwidth measurement tools try to fill all or specified part of available bandwidth with testing traffic and count how much of the testing traffic passed through. The most commonly

Tool / Classification criteron	Each link / bottleneck	Inst. / avail. bw.	RTT / dispersion	sender / receiver
Clink [2]	bottleneck	installed bw.	RTT	sender
SProbe [3]	bottleneck	installed bw.	dispersion	sender and receiver
Pchar [4]	each link	installed bw.	RTT	sender
PathChar [5]	each link	installed bw.	RTT	sender
Pathrate [6]	bottleneck	installed bw.	dispersion	sender and receiver
Pathload [7]	bottleneck	available bw.	dispersion (relative one-way delay)	sender and receiver
ABwE [8]	bottleneck	available bw.	dispersion	sender and receiver

Table 1: Classification of bandwidth estimation tools

used tool of this type is iperf [1], which can use either TCP or UDP. However, its large volume of data sent in the network stresses existing elastic traffic and the achieved TCP throughput heavily depends on specified TCP buffer size, number of parallel streams, dynamics of existing traffic and TCP implementation and can be lower or higher than real network-layer available bandwidth.

Bandwidth estimation tools send just a few carefully scheduled packets and try to estimate the installed or available bandwidth from the analysis of packet departure and arrival times. The advantage of bandwidth estimation tools is that they send much less testing traffic into the network and thus do not stress existing traffic. This in principle should allow continuous use of these tool to monitor changes in available bandwidth. Their disadvantage is that they are generally less accurate than bandwidth measurement tools.

2 Classification of End-to-End bandwidth estimation tools

A number of bandwidth estimation tools have been developed. We can classify these tools based on number of properties including:

1. Determining bottleneck bandwidth vs. bandwidth of all links along the path
2. Determining installed bandwidth vs. available bandwidth
3. Based on measuring Round Trip Time (RTT) variance vs. packet delay dispersion
4. Sender only software vs. sender + receiver software

A selection of well-known tools representing different approaches is shown in Tab. 1. The tools that estimate available bandwidth usually try to produce network layer bandwidth except for ABwE, which normalizes its output for TCP throughput. Several comparative studies of bandwidth estimation tools have been conducted (e.g., [9, 10]). Most studies were limited to slow networks (up to 100 Mb/s) or simple network topologies (testbed of a few connected routers with generated traffic. We will describe properties of a few selected tools, which appear to be most useful, present results of measurements with one selected tool on a high-speed testbed and results of combined measurements with several tools over a real fast long-distance network. Adopting the notation of [11], we will refer to the link with the minimum installed bandwidth along a path as the narrow link and to the link with the minimum available bandwidth as the tight link.

2.1 Pathchar

Pathchar [5] is based on measuring RTT variance for packets of different sizes. The tool can estimate installed bandwidth of each link along a network path. However, the results tend to be very inacurate on multihop fast networks (>100 Mb/s) with a lot of existing traffic. For example, we obtained the following result for a network path from Cesnet (Czech Republic) to Uninett (Norway):

```
lab2:/usr/local/src# ./pathchar tcp4-ge.uninett.no
pathchar to tcp4-ge.uninett.no (158.38.0.194)
 can’t find path mtu - using 1500 bytes.
 doing 32 probes at each of 45 sizes (64 to 1500 by 32)
 0 localhost
 |    60 Mb/s,   48 us (296 us)
 1 195.113.147.1 (195.113.147.1)
 |   361 Mb/s,   14 us (358 us)
 2 r21-pos0-0-stm16.cesnet.cz (195.113.156.114)
 |   102 Mb/s,   60 us (595 us)
 3 cesnet.cz1.cz.geant.net (62.40.103.29)
 |   1243 Mb/s,   4.15 ms (8.91 ms)
 4 cz.de1.de.geant.net (62.40.96.38)
 |   608 Mb/s,   7 us (8.94 ms)
 5 de1-1.de2.de.geant.net (62.40.96.130)
 |   939 Mb/s,   10.9 ms (30.7 ms)
 6 de.se1.se.geant.net (62.40.96.66)
 |   ?? b/s,   -12 us (30.6 ms)
 7 nordunet-gw.se1.se.geant.net (62.40.103.118)
 |   1067 Mb/s,   3.68 ms (38.0 ms)
 8 no-gw.nordu.net (193.10.68.30)
 |   1152 Mb/s,   9 us (38.0 ms)
 9 oslo-gw1.uninett.no (193.10.68.50)
 |   423 Mb/s,   3.39 ms (44.8 ms)
10 trd-gw.uninett.no (128.39.0.250)
 |   102 Mb/s,   31 us (45.0 ms),  13% dropped
11 tcp4-ge.uninett.no (158.38.0.194)
11 hops, rtt 44.5 ms (45.0 ms), bottleneck  60 Mb/s, pipe 339546 bytes
```

The actual installed bandwidth of all links was either 1 Gb/s (Gigabit Ethernet) or 2.5 Gb/s (OC-48). We can see that pathchar underestimated installed bandwidth of most links including the first link from the sending PC to the first router.

2.2 Pathrate

Pathrate [6] estimates installed bandwidth of the narrow link. It is based on measuring dispersion of packet trains. We will present some results of pathrate measurements in section 4.

2.3 Pathload

Pathload [7] estimates available bandwidth. It is based on the use of a sequence of so called Self-Loading Periodic Streams. The basic idea is that the one-way delay of a periodic packet stream shows increasing trend when the stream rate is larger than the available bandwidth along the path.

Measurement starts with sending the first periodic packet stream of UDP packets from the sender to the receiver. The sender timestamps each packet. The receiver compares this timestamp with the arrival time and computes relative one-way delay. If time is not synchronized between the sender and the receiver, it is not a real one-way delay, but for the purposes of this measurement we just need to know how measured relative one-way delay changes for subsequent packets. When the transmission rate of the packet stream is larger than available bandwidth, a short term overload should cause an increasing trend in measured relative one-way delay. A series of packet streams of different rates is generated until pathload iteratively finds an approximate available bandwidth.

2.4 ABwE

ABwE [8] estimates available bandwidth. It is based on measuring packet pair dispersion and designed for measuring available bandwidth on high-speed links (up to 1 Gb/s).

ABwE uses packet pairs with a fixed size and initial delay between packets. Several of these packets pairs (typically 20) are sent into the network. A pair of probing packets can be separated by cross traffic packets, which will lead to time delay between probing packets.

Conversion from time delay to available bandwidth is based on empirically taken lengths of the cross traffic packets which could be expected on the links. This value is normalized to obtain the bandwidth which is achievable for TCP traffic.

3 TestBed Configuration

Configuration of our testing environment for available bandwidth measurements is shown in Fig. 3. Cross traffic stream was routed to a non-existing IP address on the same network segment where the receiver was connected. MAC address for this IP address was statically added to the router's ARP table to forward packets to the right port. Cross traffic met testing traffic in the outgoing interface of the router and could thereby emulate specified volume of available bandwidth. We achieved the maximum cross traffic rate of 80811 packets per second, which equalled to 951.6 Mb/s (in 1500-byte packets).

The configuration included the following components:

Router Cisco 12008 GSR router with 3-port Gigabit Ethernet adapter

Computers Dell 2650 with Intel PRO/1000 adapter

Packet generator UDPgen packet generator [12].

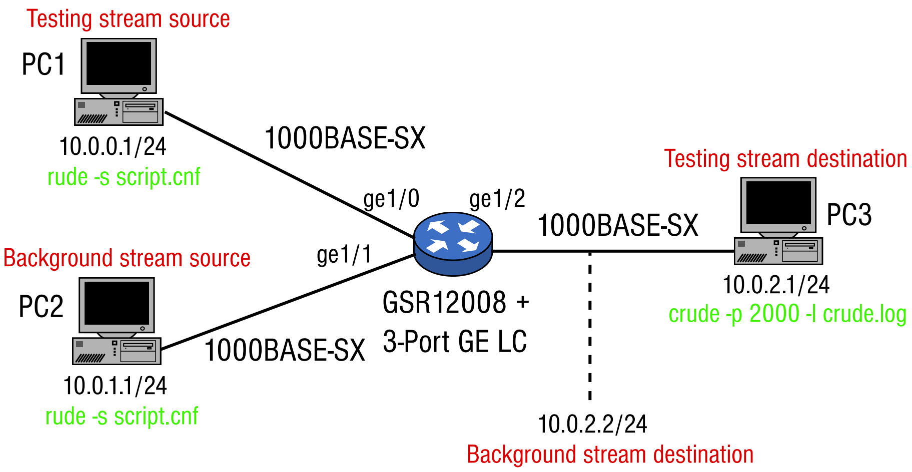

Figure 1: Testbed configuration

3.1 Evaluation of Pathload

Pathload consists of two parts - pathload_snd and pathload_rcv. The receiver (pathload_rcv), which actually performs measurements, needs two parameters - IP address of the sender (pathload_snd) and requested bandwidth resolution.

Default maximum available bandwidth measurable by pathload_rcv limited by internal constants is 120 Mb/s. To enable operation at higher speeds, we tweaked these internal constants in pathload_rcv. After some experimentation we found that we could achieve operation at the highest possible speed when `MIN_TIME_INTERVAL` constant was changed from 100 to 5 and multiplication factor in `min_time_interval` was changed from 5 to 1.5. This enabled pathload_rcv to to measure available bandwidth up to 800 Mb/s. The optimal values of these constant and achievable speed may differ on different PCs used to run pathload_rcv.

Pathload generally exits with one of the following messages:

- `Exiting due to user specified resolution (U)` - The resolution requested by user could be achieved.
- `Exiting with a wider range due to a time interval error (T)` - During at least one fleet, the sender was unable to send packets with the inter-packet spacing that the receiver requested.
- `Exiting with a wider range due to MAX_RATE constraint (M)` - This constraint limits the maximum rate that pathload can generate. If that rate is reached at a certain fleet, the tool does not proceed any further.
- `Exiting with a wider range due to MIN_RATE constraint (I)` - This constraint limits the minimum rate that pathload can generate. If that rate is reached at a certain fleet, the tool does not proceed any further.
- `Exiting due to grey region (G)` - The resolution requested by user could not be achieved due to fluctuations in measurements. The available bandwidth is approximately equal to the range of the grey region.

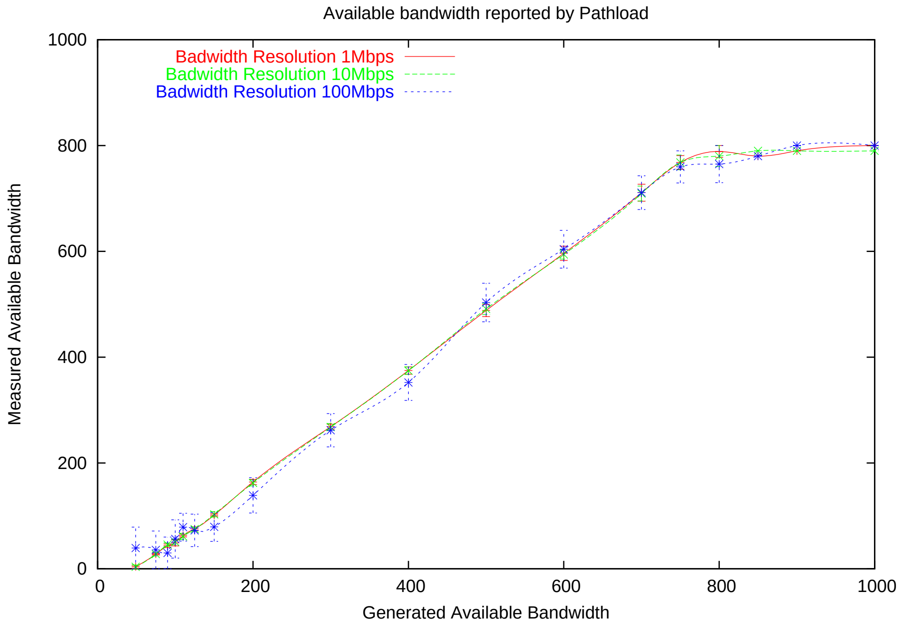

Figure 2: Available bandwidth reported by pathload

Measured values are listed in Tab. 2 and depicted in Fig. 2. Pathload reports available bandwidth as interval. Its lower bound is show as Min value and its upper bound as Max value in the table. We have done all tests for resolution set to 1, 10 and 100 Mb/s. For every volume of cross traffic we ran pathload five times and calculated averages. Throughout measurements we did not notice significant packet losses. In several iterations pathload reported packet loss in one stream in the fleet. The exit status is listed in Tab. 3, referring to the letter in parenthesis in the list of exit options above.

We can see from Tab. 3 that pathload was able to estimate available bandwidth up to the limit of 800 Mb/s when the requested precision was 100 Mb/s, which is rather coarse estimate. When the requested precision was 10 Mb/s or 1 Mb/s, pathload was generally unable to find available bandwidth with such precision. We can see from Fig. 2 and Tab. 2 that finer required precision did not have almost any effect on the produced estimate. As we will show in the next chapter, reliability of measurements on a real network was much worse. At the time when we did testbed measurements, ABwE was not yet available. Because of limited evaluating value of testbed measurements we decided that it would not worth the effort to repeat the measurement with ABwE and proceeded to real network tests.

4 Real Network Tests

We tried several bandwidth measurement and estimations tools together along two paths over Géant network, making one round of measurements and estimations per hour for the period of one month. Each path crossed more than 10 routers, all links were either Gigabit Ethernet or OC-48. We used the following tools:

- TCP iperf with socket buffers set to 4, 8 and 16 MB (capacity of the empty pipe was 4.8 MB and 5.5 MB for the first and second testing path, respectively)

Real available bandwidth [Mb/s]	Background traffic [pps]	Estimated available bandwidth for different requested precision					
		1 Mb/s		10 Mb/s		100 Mb/s	
		avg. Min	avg. Max	avg. Min	avg. Max	avg. Min	avg. Max
49	80800	1.170	6.776	0.000	7.440	0.000	78.780
75	78549	26.246	29.488	24.142	31.426	0.000	71.510
90	77275	40.014	46.904	42.158	49.080	0.000	60.092
100	76426	43.406	54.730	45.538	55.788	20.136	92.664
110	75577	60.180	65.594	55.184	67.024	52.470	104.94
125	74303	71.792	79.252	71.404	78.698	41.930	103.44
150	72180	98.738	104.002	97.716	109.980	51.922	106.416
200	67934	159.272	169.048	156.726	168.024	105.466	172.160
300	59442	246.116	273.300	261.348	274.500	230.576	293.266
400	50951	367.802	381.764	367.462	381.636	318.260	386.230
500	42459	476.684	499.764	480.494	501.820	466.872	539.822
600	33967	582.924	609.698	585.536	603.200	568.404	639.736
700	25475	694.754	726.934	695.262	723.402	679.102	742.790
750	21229	755.268	780.854	754.266	782.152	729.364	790.000
800	16983	777.670	800.000	760.196	800.000	729.800	800.000
850	12737	780.000	780.000	790.000	790.000	780.000	780.000
900	8491	790.000	790.000	790.000	790.000	800.000	800.000
1000	0	800.000	800.000	790.000	790.000	800.000	800.000

Table 2: Available bandwidth reported by pathload

Real available bandwidth [Mb/s]	Exit status for different requested precision														
	1 Mb/s					10 Mb/s					100 Mb/s				
49	G	G	G	G	G	U	U	U	U	U	U	U	U	U	U
75	G	U	U	G	G	U	U	G	U	U	U	U	U	U	U
90	G	G	G	G	G	U	U	U	U	U	U	U	U	U	U
100	G	G	G	G	G	U	U	G	G	U	U	U	U	U	U
110	G	G	G	G	G	U	U	G	U	G	U	U	U	U	U
125	G	G	G	G	G	U	U	U	U	U	U	U	U	U	U
150	G	T	T	–	–	U	–	U	U	G	U	U	U	U	U
200	G	G	G	G	G	U	G	G	U	G	U	U	U	U	U
300	G	G	G	G	G	U	G	G	G	G	U	U	U	U	U
400	G	G	G	G	G	G	G	G	G	G	U	U	U	U	U
500	G	G	G	G	G	G	G	G	G	G	U	U	U	U	U
600	G	G	G	G	G	G	G	G	U	G	U	U	U	U	U
700	G	G	G	G	G	G	G	G	G	G	U	U	U	U	U
750	G	M	G	G	G	G	G	G	G	G	U	M	U	U	U
800	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
850	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
900	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
1000	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M

Table 3: Exit status reported by pathload

- parallel TCP iperf sending 5 streams, each with 4 MB socket buffer
- UDP iperf sending up to 900 Mb/s
- pathload
- ABwE
- TCP iperf with socket buffer size moderated by results from ABwE

Although the test over just two paths was very limited, it provided a view on some relations of results indicated by diferent tools and their dynamics. A sample from one 4-day period is shown in Fig. 3.

We can observe that results produced by different tools generally did not match and it was not easy to conclude what was the real available bandwidth. More detailed discussion of our observations can be found in [14].

5 Conclusion

We conclude that current bandwidth estimation tools are not reliable and precise enough for short-term estimates to assist congestion control in choosing the right socket buffer sizes (and thus TCP window sizes). They can be used for long-term estimates to assist network planning. However, network planning is usually done by users who have SNMP access to much more

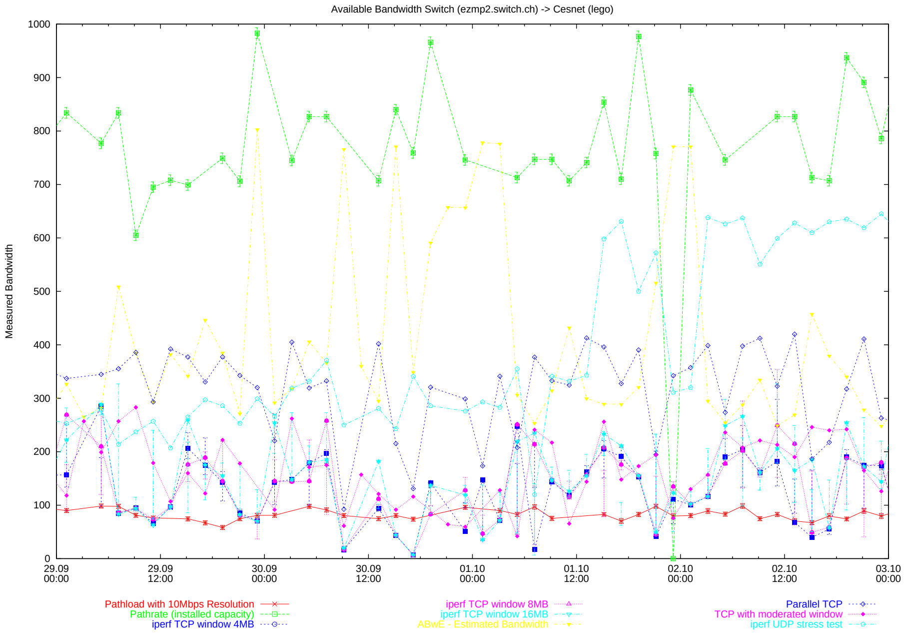

Figure 3: Combined bandwidth measurements and estimations

reliable and precise router byte counters. For short-term available bandwidth, we either need to get access to router byte counters or further research to improve bandwidth estimation tools is necessary.

References

[1] Iperf - TCP and UDP active bandwidth measurement tool, `http://dast.nlanr.net/Projects/Iperf`.

[2] *Clink*, `http://rocky.wellesley.edu/downey/clink`.

[3] Stefan Saroiu, P. Krishna Gummadi, Steven D. Gribble. *SProbe: A Fast Technique for Measuring Bottleneck Bandwidth in Uncooperative Environments*, INFOCOM 2002, New York, 2002.

[4] Bruce A. Mah. *Pchar*, `http://www.employees.org/ bmah/Software/pchar`.

[5] *Pathchar*, `http://www.caida.org/tools/utilities/others/pathchar`.

[6] Constantinos Dovrolis. *Pathrate*, `http://www.pathrate.org`.

[7] Constantinos Dovrolis. *Pathload*, `http://www.pathrate.org`.

[8] Jiri Navratil. *ABwE: A Practical Approach to Available Bandwidth Estimation*, PAM 2003, La Jolla, April 2003.

[9] Jacob Straus, Dina Katabi, Frans Kaashoek. *A Measurement Study of Available Bandwidth Estimation Tools*, IMC2003, Florida, 2003.

[10] *Network Measurement Tools Test, Part II*, Poznan Supercomputing and Networking Center, `http://qos.man.poznan.pl/files/measurement2.pdf`

[11] M. Jain, C. Dovrolis. *End-to-end Available Bandwidth: Measurement Methodology, Dynamics and Relation with TCP Throughput*, Proceedings of ACM SIGCOMM, August 2002.

[12] *UDPgen*, Kernel module for generating UDP streams `http://www.fokus.fhg.de/usr/sebastian.zander/private/udpgen`.

[13] Sven Ubik, Pavel Cimbál. *Achieving Reliable High Performance in LFNs*, TNC 2003, Zagreb, May 2003.

[14] Sven Ubik, Antonín Král. *End-to-end Bandwidth Estimation - Progress Report*, End-to-end performance project, CESNET, 2003.