Why Your Application only Uses 10Mbps Even the Link is 1Gbps?

Contents

Introduction

This document describes the issue associated with high speed, high latency network. It derives a formula from BDP to calculate the real bandwidth usage in a given condition.

Background Information

As an increasing number of enterprise has or is in the process of building geographically dispersed datacenters and interconnect the datacenters via high speed link. The needs of better utilize the bandwidth is increasing.

The Bandwidth- Delay Product (BDP) has been published on the Internet for several years. However, there is no real-world example on what the issue looks like. The BDP formula focus on TCP windows size. It doesn't give us a way to calculate the possible bandwidth usage based on distance. This document briefly explains BDP and demonstrates the issue and resolution. This article also derives a formula to calculate bandwidth usage in a given condition.

Overview of the Issue

Your company has two datacenters. Your company backup business critical data from one datacenter to another datacenter. The backup admin reported that they cannot finish the backup within backup window due to network slowness. As the network admin, you are assigned to investigate the network slowness issue. You know these factors:

- These two datacenters are 1000KM apart.
- These datacenters are interconnected via 1Gbps link.

Upon investigation, you have noticed:

- There is enough available bandwidth.
- There are no network hardware or software issues.
- The backup application only utilizes around 10Mbps bandwidth even the rest of 990Mbps bandwidth is free.
- The backup application uses TCP to transfer data.

Bandwidth-Delay Product

To answer the question of the backup application only uses 10Mbps, it introduces the Bandwidth-Delay Product (BDP).

BDP simply states that:

BDP (bits) = total_available_bandwidth (bits/sec) x round_trip_time (sec)

or, since RWIN/BDP is usually in bytes, and latency is measured in milliseconds:

BDP (bytes) = total_available_bandwidth (KBytes/sec) x round_trip_time (ms)

This means that the TCP Window is a buffer that determines how much data can be transferred before the server stops and waits for acknowledgments of received packets. Throughput is in essence bound by the BDP. If the BDP (or RWIN) is lower than the product of the latency and available bandwidth, you can't fill the line since the client can't send acknowledgments back fast enough. A transmission can't exceed the (RWIN / latency) value, so the TCP Window (RWIN) needs to be large enough to fit the maximum_available_bandwidth x maximum_anticipaded_delay.

With above formula. The derived bandwidth calculation formula is:

Bandwidth usage (Kbps)=BDP(bytes)/RTT(ms) * 8

Note: This formula calculates max theoretical bandwidth usage. It doesn’t take OS’s packet transmission time into consideration because it has many factors involved e.g. available memory, NIC driver, local NIC speed, cache or sometimes even disk speed. As a result, when the TCP windows size is large, the calculated bandwidth would be greater than the actual bandwidth. When the TCP windows size is very large, the deviation can be large as well.

With the derived formula, you can answer the question on why the backup application can only use 10Mbps by doing below calculation.

- In general, the RTT for 1000KM is ~15. So RTT=15ms
- By default, Windows 2003 operating system Windows size is 17,520 bytes. So BDP=17,520 bytes
- Put these numbers into the formula:

Bandwidth usage (Kbps) =17520/15*8.

The result is 9344Kbps or 9.344Mbps. 9.344Mbps plus TCP and IP header. The end result is

~10Mbps.

Verify

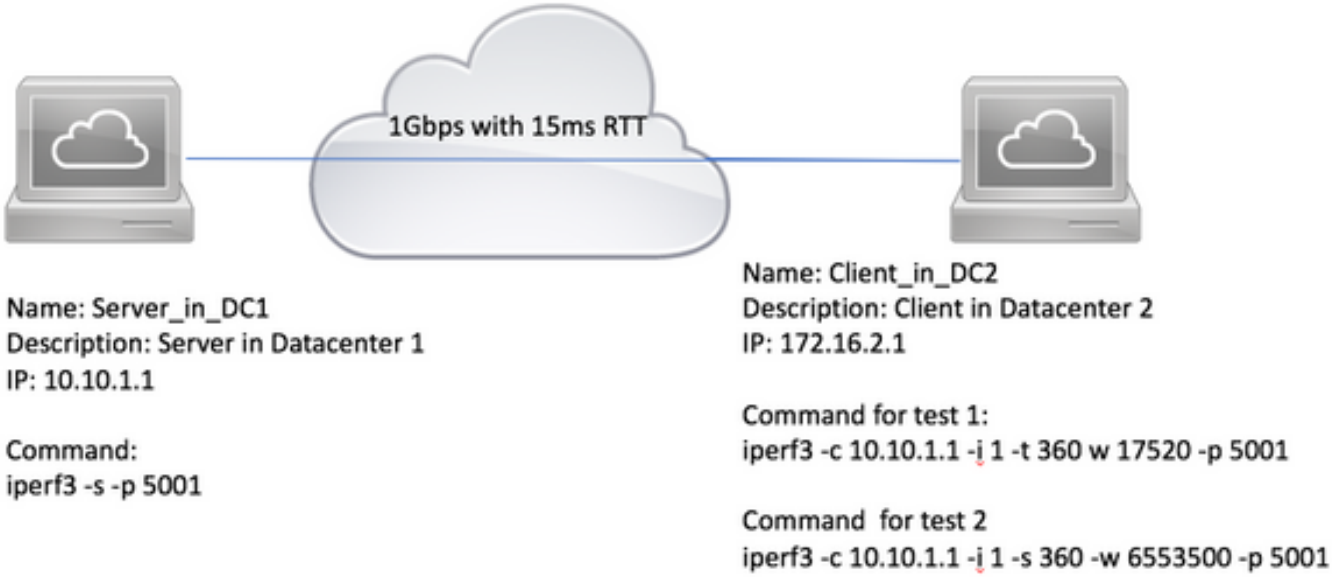
As a network admin, you have theoretically answered the question. Now you need to confirm the theory in real world.

You can use any network performance testing tool to confirm the theory. You have decided to run **iperf** to demonstrate the issue and resolution.

This is the lab setup:

1. A server in datacenter 1 with IP address 10.10.1.1.
2. A client in datacenter 2 with IP address 172.16.2.1.

The topology is as shown in the image:



Please follow these steps to verify:

1. Run **iperf3 -s -p 5001** on 10.10.1.1 to make it a server and listen on TCP port 5001.

2. To test with default TCP window size 17,520 bytes. Run **iperf3 -c 10.10.1.1 -**i **1 -t 360 -w 17520 -p 5001** on 172.16.2.1 to make it a client. This command tells iperf to connect to the server on port 5001, runs for 360 seconds and reports bandwidth usage every 1 second with TCP windows size 17,520 bytes.

3. To test with customized TCP window size e.g. 6,553,500 bytes, Run **iperf3 -c 10.10.1.1 -i 1 -t 360 -w 6553500 -p 5001**

This is the lab test result with default TCP Window size 17,520 bytes. You can see the bandwidth usage is ~10Mbps.

```
C:\Tools>iperf3.exe  -c 10.10.1.1  -t 360  -p 5001 -i 1 -w 17520
```

```
Connecting to host 10.10.1.1, port 5001
[  4] local 172.16.2.1 port 49650 connected to 10.10.1.1 port 5001
[ ID] Interval           Transfer     Bandwidth
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  1.30 MBytes  10.9 Mbits/sec
[  4]   1.00-2.02   sec   919 KBytes  7.41 Mbits/sec
[  4]   2.02-3.02   sec  1.28 MBytes  10.7 Mbits/sec
[  4]   3.02-4.02   sec  1.14 MBytes  9.59 Mbits/sec
[  4]   4.02-5.01   sec  1.24 MBytes  10.4 Mbits/sec
[  4]   5.01-6.01   sec  1.33 MBytes  11.3 Mbits/sec
[  4]   6.01-7.01   sec  1.15 MBytes  9.65 Mbits/sec
[  4]   7.01-8.01   sec  1.12 MBytes  9.36 Mbits/sec
[  4]   8.01-9.01   sec  1.22 MBytes  10.3 Mbits/sec
[  4]   9.01-10.01  sec  1.13 MBytes  9.49 Mbits/sec
[  4]  10.01-11.01  sec  1.30 MBytes  10.8 Mbits/sec
[  4]  11.01-12.01  sec  1.17 MBytes  9.84 Mbits/sec
[  4]  12.01-13.01  sec  1.13 MBytes  9.48 Mbits/sec
[  4]  13.01-14.01  sec  1.28 MBytes  10.7 Mbits/sec
[  4]  14.01-15.01  sec  1.40 MBytes  11.8 Mbits/sec
[  4]  15.01-16.01  sec  1.24 MBytes  10.4 Mbits/sec
[  4]  16.01-17.01  sec  1.30 MBytes  10.9 Mbits/sec
[  4]  17.01-18.01  sec  1.17 MBytes  9.78 Mbits/sec
```

This is the lab test result with TCP window size 6,553,500 bytes. You can see the bandwidth usage is ~200Mbps.

```
C:\Tools>iperf3.exe  -c 10.10.1.1  -t 360  -p 5001 -i 1 -w 6553500
Connecting to host 10.10.1.1, port 5001
[  4] local 172.16.2.1 port 61492 connected to 10.10.1.1 port 5001
[ ID] Interval           Transfer     Bandwidth
[  4]   0.00-1.00   sec  29.1 MBytes   244 Mbits/sec
[  4]   1.00-2.00   sec  25.4 MBytes   213 Mbits/sec
[  4]   2.00-3.00   sec  26.9 MBytes   226 Mbits/sec
[  4]   3.00-4.00   sec  18.2 MBytes   152 Mbits/sec
```

```
[  4]   4.00-5.00   sec  25.8 MBytes   217 Mbits/sec

[  4]   5.00-6.00   sec  28.8 MBytes   241 Mbits/sec

[  4]   6.00-7.00   sec  26.1 MBytes   219 Mbits/sec

[  4]   7.00-8.00   sec  21.1 MBytes   177 Mbits/sec

[  4]   8.00-9.00   sec  22.5 MBytes   189 Mbits/sec

[  4]   9.00-9.42   sec  9.54 MBytes   190 Mbits/sec
```

Solution

From software development perspective, multi-threading to run multiple concurrent TCP secessions can improve bandwidth usage. However, it is not practical for network or system admin to modify the source code. What you can do is fine tune the OS.

RFC1323 defines multiple TCP extensions for high performance TCP. These includes Window Scale Option and selective ACK. They are implemented by the main operating systems. However, by default, some OS disable them even the TCP/IP stack are written to support them.

- These OS disable RFC1323 by default: Windows 2000, Windows 2003, Windows XP and Linux with kernel earlier than 2.6.8.

If you face the issue on Microsoft Windows system, please follow this link to fine tune TCP. https://support.microsoft.com/en-au/kb/224829.

For other OS, please see vendor's documentation on how to configure them.

- These OS enable RFC1323 by default: Windows 2008 and later, Windows Vista and later, Linux with kernel 2.6.8 and later. You may need apply patches to improve these functions. In some circumstances, it is desired to disable them. Please see vendor's documentation on how to disable them.

- Some appliances are built on top of Microsoft Windows 2000,Windows 2003 or embedded operating system. e.g. NAS, Health care hardware. Please check vendor's documentation to verify whether RFC1323 is enabled or not.

How to Tell Round Trip Time (RTT) between Two Locations?

In general, RTT is associated with distance. Below table lists the distance and its relevant RTTs. You can also use ping test to get some idea on the RTT in normal network conditions.

Distance (KM)	RTT(ms)
1,000	15
4,000	50
8,000	120

Note: Above is a guide only, the real RTT time can be vary. Also, the latency is impacted by the technology used. For example, 3G latency can be 100ms frequently regardless the distance. It is true for satellite as well.