Virtual Access

Cisco LNS Optimisation for MLPPP

The importance of PPP timeout multilink lost-fragment

Issue 1.0
Date 7 April 2010

How “lossy” networks can affect MLPPP

Multilink PPP (MLPPP) offers great benefits to service providers as well as end-users. However, service providers have to ensure that their MLPPP virtual template is optimised for this technology for the end-user to see the full capabilities of this product.

In general MLPPP is used to bond two or more DSL lines together. Traffic is distributed over the links by one end and then reassembled by the other end. In more technical terms the MLPPP transmitter assigns sequence numbers to packets before they are transmitted on the broadband links so that if they arrive out of order they can be re-ordered by the MLPPP receiver. This works fine until there are lost packets in the streams. Lost packets are typically caused by noisy DSL lines or by congestion in the access network.

When two physical links are used there are algorithms that can be used to detect MLPPP packet loss quite quickly. However, when L2TP is used MLPPP is bonding links that are extended over an IP network. IP networks do not guarantee packet order integrity and so it becomes more difficult to detect packet loss, and the loss is typically detected using a timer.

When a packet is lost the receiver that is re-ordering the received packets waits for a specified period of time to see if the lost packet is going to arrive. While it is waiting, it buffers up the other packets that it received. Eventually it decides the lost packet is not going to arrive and sends the buffered packets up the stack after the timeout.

In the case of a typical broadband network scenario the receiver is the LNS. By default, a Cisco LNS waits for 975ms before deciding a packet is lost. This 1 second delay is a considerable amount of time, especially if an end-user is running VoIP over their MLPPP connection. The maximum latency threshold for VoIP is around 250ms; obviously a delay of 975ms is unacceptable. For typical data traffic this delay will cause TCP stacks to retransmit, maybe more than once, and the throughput becomes extremely poor. Following the 975ms delay it will take the TCP stacks some time to recover to a steady flow again.

Cisco IOS includes the parameter `ppp multilink slippage` that attempts to use the receive buffer size in order to optimize the timeout, but with the widely varying sizes of IP packets it is not at all effective.

However, Cisco has recently introduced a new command in their IOS software version 12.4(15)T12 and all later versions. This command allows the service provider to reduce the amount of time the LNS waits to decide whether a packet is lost.

This command should be added to any virtual template that supports MLPPP:

```
ppp timeout multilink lost-fragment <seconds> <milliseconds>
```

Virtual Access recommends setting this timeout to 100ms and so the setting in the MLPPP virtual template should be:

```
ppp timeout multilink lost-fragment 0 100
```

This setting of 100ms is usually enough time to make sure the packet is definitely lost, and yet is fast enough to get packets through before TCP retransmits. This alteration to the virtual template will ensure that regardless of what type of traffic the end-user is running over their MLPPP connection, the MLPPP protocol will not add to the problem.

Practical throughput measurement

The following test gives some perspective as to how much of an improvement this configuration command can make. For this example we have introduced a heavy 1% packet loss in the upstream data path.

Before

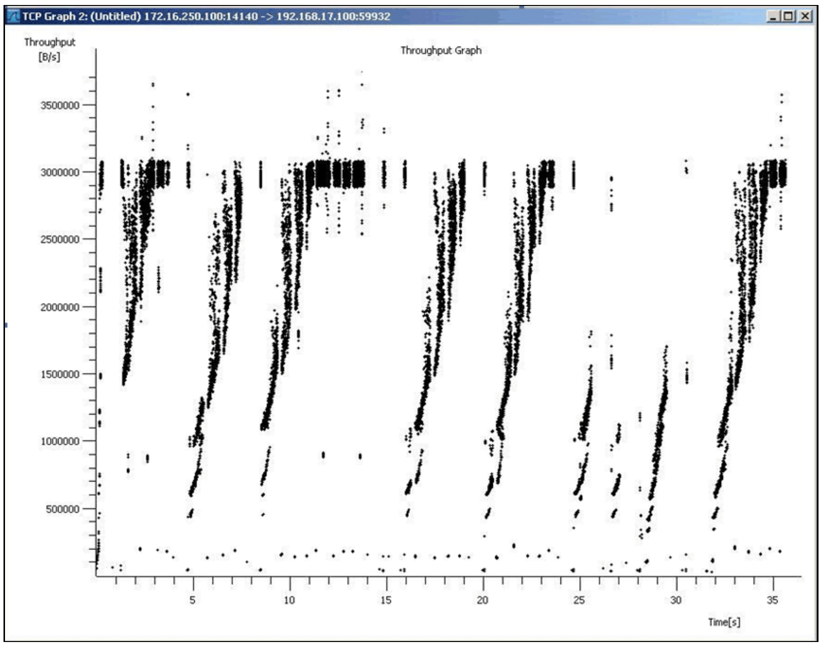

Figure 1: TCP throughput without packet loss timer (1% upstream packet loss)

After

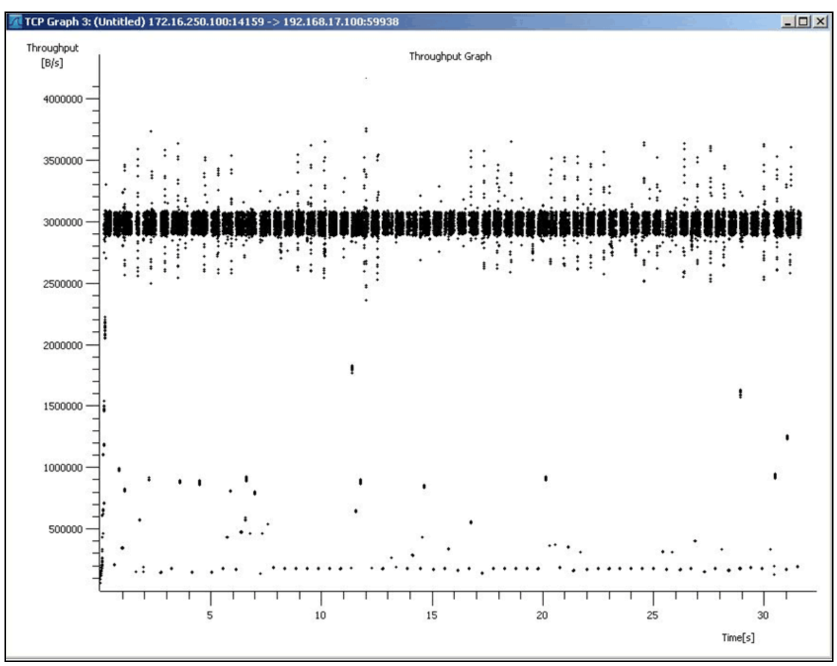

Figure 2: TCP throughput with 100ms packet loss timer (1% upstream packet loss)

Before
Figure 1 is a graph of a FTP TCP downstream throughput, produced by Wireshark. This graph demonstrates very severe breakages in the throughput. These breaks last at least 1 second and it can be seen that the TCP stream takes some time to recover to its steady state of 3M bytes/sec. This is clearly unacceptable and would have disastrous effects in relation to end-user experiences.

After
Figure 2 shows the throughput in an identical test environment, again with the very heavy 1% upstream packet loss but with the lost-packet timer set to 100ms. We can see a far more consistent and smooth throughput flow and any breaks in transmission are quickly recovered from in around 100-150ms. This means that the end-user experiences a far superior connection quality.

Conclusion

The introduction of this new timer in Cisco's MLPPP implementation dramatically improves MLPPP performance and end-user experience. Any service provider using Cisco as their LNS to terminate MLPPP should upgrade to a version of IOS that supports the timer and add the timer to their MLPPP virtual templates as soon as possible.