Introduction

The most popular packet capture solution in the world is WireShark. It's no co-incidence that it's free, but it is also quite capable. Even the more expensive analysers with extended problem databases and issue recognition features all support the WireShark file formats, which have become a "defacto" file standard these days. However whether the software is free or paid for, it can only be as good as the information that gets passed up to it. This is where we need to be careful ……

Hardware and NIC cards

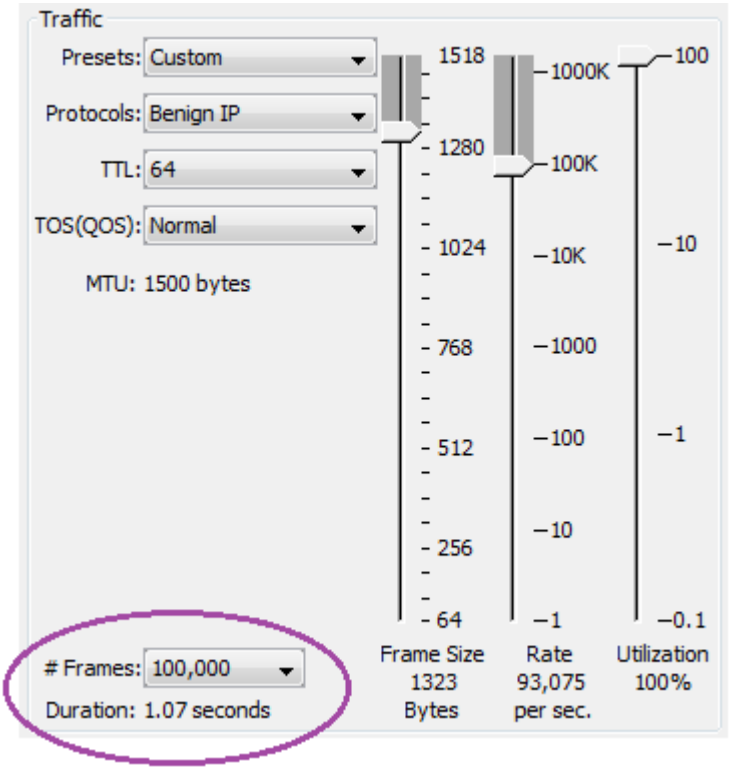
Fundamentally the NIC card receives the packets from the network into a receive buffer, then copies the information out and sends it up the stack. The speed at which the buffer fills and empties is the key here. A 16 bit BUS copies data out of the buffer at a rate of 16 bits with each processor cycle allocated to the job (maybe 750 cycles for one frame). A 32 bit BUS copies twice the amount of data with each cycle therefore needs half the cycles to copy the same amount of data. A 64 bit architecture is copying twice as much again, hence requires even less cycles to complete the job.

The challenge begins when the network is forwarding packets faster than the BUS architecture can copy it out. If the receive buffer is full and a packet arrives the dropped packet counter on the NIC card is meant to recognise this and record the drops.

Experiment

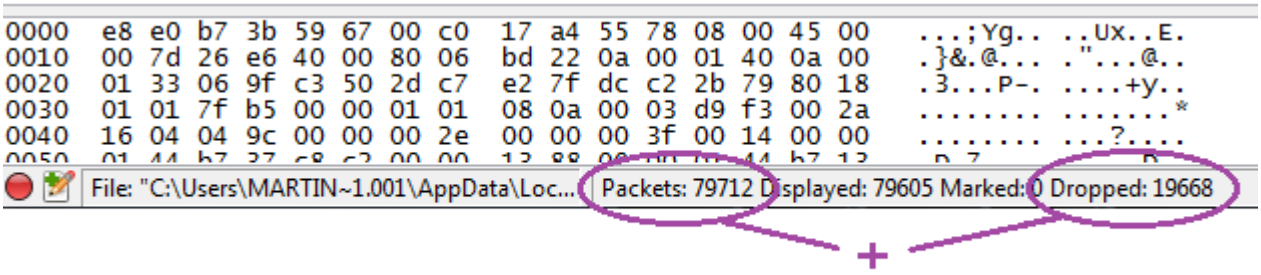
We set up a simple test, pointing the traffic generator on a Fluke Networks XG box to send 100,000 packets to a laptop running Wireshark. The laptop is a standard Toshiba with a P5 processor and 8Gig of memory.

The traffic generation was set to 1Gig (100%) and at a rate of almost 100,000 frames per second at 1500bytes.



01677 428700
info@fullcontrolnetworks.co.uk
www.fullcontrolnetworks.co.uk

The corresponding Wireshark trace can be seen below, so how many packets were recorded? Roughly 20% of the packets were not analysed due to the performance of the card not keeping up, marked as Dropped. However adding the Packets and Dropped numbers together show that around 700 packets (there are some broadcasts in here to be filtered out) are MIA - missing in action!

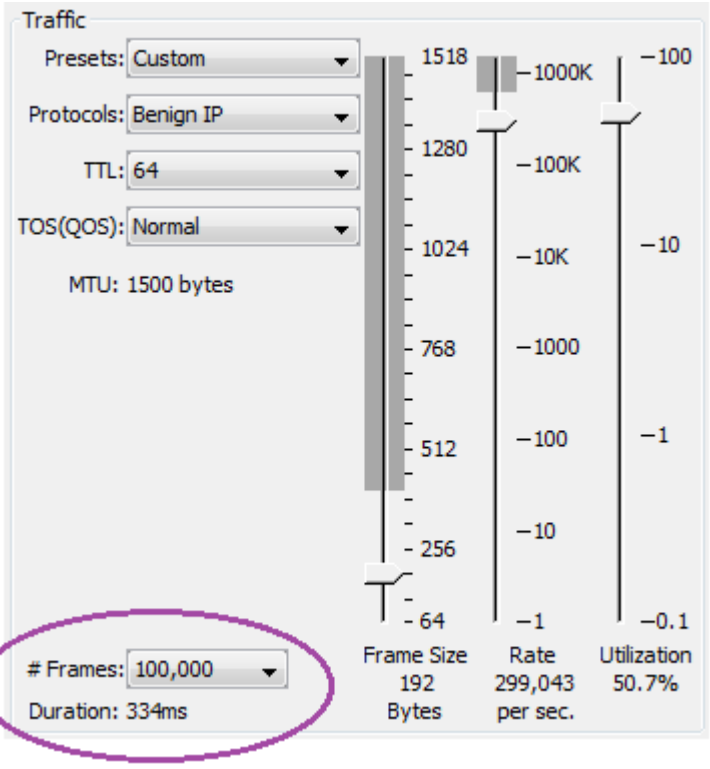


The experiment shows that you only have a partial story to start with and there is an element of “not recorded anywhere packets”; which you wouldn’t even know that you’ve missed! When you look at the captures the software is going to spot this and report frames missing (based around the IPID and sequence numbers). Wireshark does have the ability to see that TCP isn’t complaining about the missing packets, therefore it must be the capture data that’s wrong rather than conversation itself and reports the gap in the data as “segment not captured”.

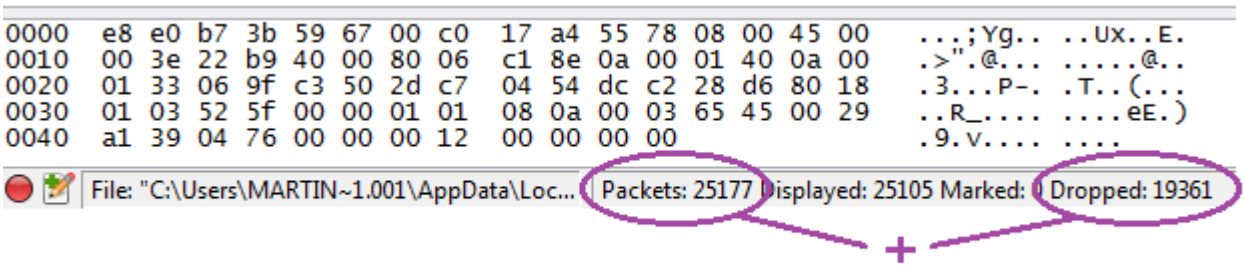
The real question is can you go and pick a fight with the network/server/application/software supplier team based on 20% dropped data and 1% MIA data?

We can actually make this situation worse …….

In this second example, still 100,000 frames but the packet rate has been upped to nearly 300,000 packets per second at a 192 byte size which gives us a burst of traffic for 334ms.



01677 428700
info@fullcontrolnetworks.co.uk
www.fullcontrolnetworks.co.uk



The total number of packets seen is 25,177 + 19,361 = 44,538

The Wireshark capture (above) shows only 25% of the packets recorded and 20% of the packets dropped; which means a massive 55% of the capture is MIA. In this situation the quality of the data in the packet capture engine is next to useless.

What the experiment shows is that fast bursts of traffic, such as 300,000 packets per second, are more dangerous than the steady large file transfers. If the kit we are using is struggling to read these bursts will a user or server device also struggle? In this experiment we lost more than half the packets in 330ms – clearly it's the bursting that's the issue.

In real life is this relevant?

Good question, well sort of. How many Gigabit networks actually get this busy, not that many really but these are the networks or parts of the networks that you don't look at much. A better question might be are you more likely to use your packet analyser in a quiet or busy network, then the reality of the testing above makes more sense?

Talking to Server and Application people the issues caused by small but intense bursts of traffic (micro bursting) is very real and this is the type of traffic which will over run a receive buffer for short periods and drop frames. The issue highlighted here is that you are looking because you don't know and you can't learn what you can't see, so whether it is counted as a dropped packet or not, you have issues.

Does this mean you should never use your laptop to do this work? Probably, but we will all still do it as it's just too convenient. What you need to be aware of is the limitations of what you are doing and the knowledge that the dropped packet counter is not an accurate guide of whether the capture is accurate or not.

How can you improve the quality of the capture?

In a laptop/PC there are a few things that can influence the number of packets that make it to the capture.

- Quality of the NIC card
  - We advise a proper RJ45 interface and not a stubby little convertor cable style connector
- Bus speed
  - 64 bit cards are going to copy data in less cycles than 32 bit etc , so using the old PC in the corner as a capture store really doesn't make sense
- Number of processor cycles allocated to the software - there are a few areas you can influence here:-
  - Don't have other applications open on your laptop

01677 428700
info@fullcontrolnetworks.co.uk
www.fullcontrolnetworks.co.uk

- Don't use pre-capture filters, these can be a very intensive user of resources
- Faster processors do help

- Increasing the Receive buffer size will increase the size of the burst of traffic that can be absorbed before this becomes an issue

What other options are there?

There are quite a few dedicated capture options out there, but most of them revolve around a custom NIC card with its own resources on board. This means that the card is not fighting for resource from the PC and can keep up with the network speeds. With most of these technologies the card also "timestamps" the packets at this point so the slower process of writing the data to disk does not influence what the Packet Analyser has to say about your capture.

At Full Control Networks we offer a few different ways to get into this technology:-

InveaTech dedicated NICs – build your own packet store using a custom line rate card, although the RAID array speed needs to be watched.

Fluke Networks XG – Portable 1 and 10Gig capture at line rate (to memory only to keep the speed up).

Fluke Networks NTM/TruView – Custom NIC cards and RAID solutions for capturing and storing data at line rate.

Bottom Line

Whichever analysis engine you use, it's probably OK as long as it can see all the data. The question is how much data is it seeing, how much does it not see and how much is completely missed? Can you trust the results ……..

01677 428700
info@fullcontrolnetworks.co.uk
www.fullcontrolnetworks.co.uk