Internet Parameters

And how to best use them

Contents

Introduction

For our dynamic load controller we need to monitor network statistics. To do this effectively, we must explore what statistics there are, and how we can best use these statistics; information which is currently unknown or vague.

Research questions

1. What different parameters can we appoint when measuring network?
2. How can we measure these different parameters?
3. What network parameters can we measure passively?
4. What parameters can we use to predict/recognize a dip in network speed?
5. Can we retrieve data from the network infrastructure?

What different parameters can we appoint when measuring network?

When measuring a networks performance, various terms are introduced to help put a name to various behavioural patterns inside of a network. Below we list 6 of the most applicable ones to our research.

Latency

Network latency is a term used to describe the time it takes one data-packet to go to its destination, and back. The time it takes this packet is described in milliseconds.

This information is not always useful but can negatively impact communication protocols which wait for an ACK-signal to progress in their control loop.

Jitter

Jitter refers to a variable latency. Ideally, in a network, the user would like either no latency; or a steady, low, latency.

Jitter happens when the latency fluctuates heavily. This is especially noticeable with livestreams/feeds or voice connections, as it will seem like the connection either slows down or significantly speeds up randomly.

Packet Loss

Previously we already mentioned packets, in those contexts; we assume that all packets make the target destination. Though in some cases, they do not. This is called packet loss.

While previously, those packets would always more or less arrive, though just with a delay; in this case they do not at all. Which means it has more drastic effects on the application in need of that data.

Throughput

Throughput refers to the amount of data flowing **through** a pipeline (both in and out!), in our case this pipeline is our network. This means you can measure throughput at many different parts of the network.

Bandwidth

Bandwidth is a very common term. It refers to the total amount of data a network can transmit in a set amount of time. Usually, the bigger the bandwidth, the more data can be pushed through the network.

How can we measure these different parameters in a network?

Knowing what a parameter says about a network can create a good understanding of what to look for. Next, knowing how to measure a parameter can provide more insight into how a network functions, and which measurements might be right for the project.

Latency

Latency cannot be measured over the entire network. We cannot say "The networks latency is x". This is because latency is always between two points, the sender and receiver.

We measure latency by sending a pre-determined amount of bytes to a certain destination. When we start sending the first packages, we start a timer. We stop this same timer when the receiver of the bytes lets us know they have received said bytes.

The time measured is called, latency.

Jitter

Measuring jitter is easy given we know how to measure latency. To measure jitter, we just need to save a few latency measurements; and calculate the difference between it.

Packet Loss

To measure packet loss, we send a predetermined amount of packets (bytes) to a receiver. On the receiver, we check how many packets we received. Since we know the start amount, we also know how many we should receive.

For example, we send 100 packets to receiver X. X measures how many packets it receives, in this particular instance; it receives 78. This means that the packet loss is 100 – 78 = 12%.

Throughput

Measuring throughput is simple, but often hidden in hardware registers. It relies on counting the RX and TX bytes of a network adapter/router/network pipeline etc. Usually this is done in seconds. Observe all the traffic coming and going, and that is the throughput.

For example, a pipeline sends 16mb of data and receives 25mb of data per second. The throughput of the TX is 16mb/s, the throughput of the RX is 25mb/s.

Bandwidth

Bandwidth cannot be measured, since it is a parameter which exists outside of the control of the users. Bandwidth is "given" to a network by either its internet provider or the hardware which the network is set up with.

What network parameters can we measure passively?

Many network parameters require an initial transfer to be able to measure anything. Download speed for example, measures the time it takes to download a certain file. In essence, loading the system.

To avoid this, we need to look for measuring techniques which do not involve transferring extra data onto the network; or techniques which can rely on already sent data.

Without a program built by the user, there is no way we can use data which exists on the network as a substitute for data which we otherwise would have artificially sent. For example, we cannot measure latency from bytes sent by our browser; since we do not have access to the code of the browser. This means we cannot start measuring the exact moment these bytes are sent, creating a massive delay in our measurement.

However, if we need to sent data from a program we programmed ourselves; it becomes a whole different story. Since we can precisely predict and check if data has been sent, this means we can also precisely predict and check when we need to start measuring. This way, we can repurpose data to include more than its basal functionality.

However, there is a metric which we will always be able to measure regardless of its source. Which is throughput.

Throughput can be viewed like a person standing beside the highway, counting the cars that come by. It doesn't matter which cars they are, who they're from or whats in them; they're going by. That's all that counts. Throughput counts network bytes in a similar fashion, regardless of their source.

We can use this to always generate a pretty clear picture of throughput in a system, while not being bound by specifics. It is the only true parameter which we can always measure passively.

What parameters can we use to predict/recognize a dip in network speed?

Predicting network usage is near impossible, since the load induced on the network is tied directly to the amount of devices using it. “knowing” what each device is going to do, is not feasible.

However, we can continuously probe the network for its metrics; to determine if our added load puts the network at risk.

If we know just a few parameters, we can determine if our internet speed is sufficient to comfortably carry the load induced by our program. We can do this the following way.

If we take the size of a single frame, the frames per second and the current internet speed, we can very easily predict if our network is up to the task.

For example we can derive this formula from this ideology:

*LoadonNetwork=(frameSize*FramesperSecond)/NetworkSpeed*

This equation will give us the amount of times we can run this program on our current bandwidth. The current networkspeed already takes into account the processes currently using bandwidth; but does not account for processes which have yet to induce a load on the system. For example, a tertiary user opens a web browser for video streaming. Therefore, we would like to have a comfortable margin where we can safely assume our process will run unaffected.

Though like previously mentioned, this solution will not prove valuable. Since to get network speed, we have to load the network artificially. We can bypass this, by using the previously mentioned throughput metric.

If we take the total throughput of our device and check it against a threshold. We can determine how much of that threshold has been reached. Then we can use this metric, to see if we want to induce more load onto the network; or if that load might put us over the threshold.

For example:

CurrThroughput = 1000 bytes.

AddedLoad = 500 bytes.

Threshold = 1700 bytes.

(1000 + 500) !> 1700 bytes.

In above example, we could safely add more load to the system without reaching our threshold.

Can we retrieve data from the network infrastructure?

Another question we early on explored, is if we could poll or ping certain network infrastructures the same way you can ping HTML pages for example.

This way, we could get information about the rates (speed) of certain processes, or maybe even the internet plan with its bandwidth specifications.

It turns out that, against expectations, it is actually possible to ping routers. This is doable via a plethora of ways; but does require some back-end setup.

It all makes use of the functionality which all routers have, which is a LAN login page. When the user knows the ssid and password to this login page, we can make a pre defined statement to execute once were logged in. Many examples shared online were simple queries, like changing the password or changing the ssid name.

The query and return values of these statements vary from router to router however, which makes this method very non generic and thus not suitable for our use case. Not only these return values make it unpractical, the verification needed to even access this data is a bigger obstacle. Something the user will never have, unless its their own network; at which point accessing data this way is extremely complicated compared to checking it manually.

Conclusion

In conclusion, we can determine that there are certainly many different ways to measure a networks performance. However, to us; not putting a load on the network is very important. Which keeps us coming back for the “Throughput” metric. One that proves itself best every time by inducing no new load, while still saying a lot about what is happening to the network in its current state. We explored some techniques and information about this metric which we can certainly use in our product.

Sources

https://obkio.com/blog/how-to-measure-network-performance-metrics/#network-metric-6-bandwidth

https://cplusplus.com/forum/general/122819/

https://en.wikipedia.org/wiki/Network_throughput

https://www.techwalla.com/articles/how-to-calculate-network-throughput

https://en.wikipedia.org/wiki/Router_(computing)

https://www.howtogeek.com/794161/what-wi-fi-router-specs-are-the-most-important/

https://www.cnet.com/home/internet/yes-your-router-collects-data-on-you-heres-how-to-protect-your-privacy/

https://www.techtarget.com/searchnetworking/answer/What-metrics-and-tools-can-accurately-measure-network-performance

https://www.geeksforgeeks.org/performance-of-a-network/