# Katalyst Status View Enhancements

Katalyst is the interface into the K4 solution, and a key aspect is to provide service status as intuitive as if the client is “on the grid”, while being “off the grid”. To fulfill this, the Katalyst app shall support a link status icon in the mobile’s title bar, a ‘K4 link status icon’. Additionally, and again following industry norms, the devices lock-screen notifications shall display Katalyst alerts with easy click through to the Katalyst application.

The primary and very important goal is that when a vessel is off the grid, K4 is the service provider and thus a device on our K4 vessel or even remote on the cloud can reliably understand the performance of their internet as easily if they are moving around town using their MNO provider.

Importantly, to improve K4 brand awareness, a Katalyst “lite” application functionality is needed such that K4 Titlebar information to any device on a K4 vessel, however they do not get core Katalyst services. The light running Katalyst would allow the Owner, Guest, Kids, Mom, Dad or Crew to see the state of the link – just as if they are On the Grid on shore. It is critical that most mobiles on vessel are running the application, and the value of this well beyond to the customer/client, is that K4 can utilize the device/katalyst app to monitor WiFi/LAN/Application health and improve AM SDWAN capabilities - which is key for K4 success.

Finally, the Katalyst App is pulling information from the vessel’s K4 server, and this interface – call it K4 API, must be visible from all VLAN (public), resilient against high request rates (TIC) and allow for token/certificate access control – thus eventually the API could be licensed and sold.

## Katalyst Status Design Overview

The operator of the vessel and K4 system shall assess OGi performance using the Katalyst application running on their device (phone, tablet)[[1]](#footnote-1). Defined here are the katalyst application’s mobile title bar ‘K4 link status icon’, and lock-screen alert notifications. Importantly, the device with the Katalyst application should provide vessel status on/off the vessel, and as well at device startup a background helper application should run that provides the titlebar and notifications “by default”.

A screenshot of a cell phone

Description automatically generated

Figure Katalyst/Katalite Titlebar Vessel Link Status Titlebar Icon reference view.

Screen of a cell phone

Description automatically generated

Figure 2 Katalyst Service Status Titlebar icon. This should be supported w/ Katalite app as well.

It is important that the Katalyst Titlebar service status icon follows industry norms and are intuitive, particularly the status should be similar to MNO service availability icons since K4 is the Off-Grid Internet provider. Thus, the Katalyst Service Status icon should have bars representing link strength, technology indicator, Color Representing link quality (green good, red bad), K4 branding and current vessel uplink usage. Further, the application should pull this information from the K4 Box server onboard (or cloud if off-vessel), it is not device driven data. With this, the Katalyst App requires a secure API call to pull this information when the Application is awake and running. It is important that the App only queries the K4 Box or Cloud if the phone is awake and in-use.

Further, I do think it is critical that K4 develops a K4 “Katalyst” mode that provides only the “Link Status” icon functionality as shown in Figure 2, and in the form of a SDK as well. Thus, any passenger on the vessel can install this App and understand the performance of the vessel’s internet.

The various icon views and types are shown in Figure 3 with a brief description of each.

A screenshot of a cell phone

Description automatically generated

Figure K4 Katalyst Vessel Link Status titlebar icons.

The views should be similar to what is shown in Figure 3 and Figure 2. Defined here are the methods to calculate and show this information.

#### Link Status Icon

The Link status refers to the Background color of the Speedbar; the colors being red, yellow, green or white. The meaning of each is stated here:

There fuller definition is here:

1. White with bars – Stable Link Up, and Ping RTT is < Congestion/Stability threshold for WAN tech.
   1. Assuming K4 box pings every 5s, use last 3 ping running average.
   2. Ping RTT (last 3 ping ave for Wifi, Ethernet) < 100ms
   3. Ping RTT (last 3 ping ave for LTE) < 200ms
   4. Ping RTT (last 3 ping ave for SAT) < 750ms
   5. Note that these thresholds must be configurable, or another method is to self-determine each WAN link periodically via data quiet periods, measure “idle” latency and record this “daily” or similar - and use this value (Machine Learn).
2. Yellow with bars – Unstable Link Up, and Ping RTT is > Congestion/Stability threshold for WAN tech.
   1. Assuming K4 box pings every 5s, use last 3 running average.
   2. Ping RTT (last 3 ping ave for Wifi, Ethernet) >= 100ms
   3. Ping RTT (last 3 ping ave for LTE) >= 200ms
   4. Ping RTT (last 3 ping ave for SAT) >= 750ms
   5. Note that these thresholds must be configurable, or another method is to self-determine via quiet periods, measure “idle” latency and record this “daily” and use this value (Machine Learn). If the link is available.
3. Red no bars – Link Down, Pings are being lost.
   1. Assuming K4 box pings every 5s, use last 3 running average.
   2. No response for last 3 pings.
4. White no bars – Unknown, the Katalyst application cannot connect to the K4 Box or Cloud service to get information, or the data retrieved from the K4 Box is not populated.

For RTT calculation, the K4 Box is pinging the Cloud for link status every 5 seconds. With this, there are chances in where some pings are not responded to, and thus some examples are given here on latency calculation. If 2 pings are lost, and 1 is received – suggest using the ping timeout value for the average. (is this 3s now?)

1. Thus 2 lost pings, last ping was 100ms. Use 3000ms as the lost ping time.
   1. E.g. (3000+3000+100)/3 ~ 2000ms, thus yellow.
2. 1 lost ping…
   1. E.g. (3000+100+75)/3 ~ 1100ms, thus yellow.
3. If 3 lost pings, then red – all pings lost.

The reason for this is if 1/3 of the pings are being lost, then this will show up as a yellow indicator giving evidence of link congestion/stability issues. That is, one lost ping in a 15 second window will trigger a Yellow indicator.

For the BOX RTT idle calculation. Determine the quiet period by observing 1/ successful pings and 2/ no Internet usage for 30s period (<50Kbps of usage); and with the quiet periods measure the idle RTT and increase this measure by 1.3x as the threshold. The daily/frequent measure can be helpful – since as the vessel moves from LTE network to network, the idle/inherent latency of the network can change to the Internet – thus having a machine learned value would prove more reliable.

#### Link Performance Bars

A primary challenge with performance is that the uplink network is not operated by K4, and generally speaking Radio RF performance data is not available in real time. Additionally, the radio performance is not always a good assessment of WAN link performance since the rates observed are driven by the service plan MIR and WAN link channel occupancy and usage by other remotes in the radio network; thus using radio RF data (SNR) alone or at all is not accurate. Another note here, with K4 SDWAN being adaptive and multilink, the performance bars will need to be a sum of all utilized WAN links. Therefore, here we are going to set performance bars as the measure of the WAN link or links summed performance / data rate in Mbps.

The WAN performance measurements are made from the K4 BOX, and this data is continuously captured and stored on the Katalyst API for use[[2]](#footnote-2). Another important note is that performance is a measure of the vessel’s D/L Performance (estimated peak download rate) and not a measure of “congestion” or a measure for that specific client w/ the Katalyst app; the Katalyst app is the display of “bars” from the vessel-level test and measure.

A quick note here on Box N+1 D/L test. The Box should perform a D/L test every 5 minutes, and the file size should be 1 MB in size, with some augmentation on Rate calculation based on the WAN RTT. Another method to reduce usage, would be if the WAN D/L data volume for the last 5 minutes < 5 MB, and there are no Katalyst Application polls from on-vessel or cloud, push out test another 5 minutes. Further, beyond periodic measurements from Box, if a client with Katalyst application is active, this should trigger D/L test if no test was run in the last minute to provide a more accurate PB value, and is re-run every 1 minute if the mobile is unlocked/open. Finally, the test needs to be SSL based, thus to avoid any localized or in flight caching of content; and I am open to other methods to measure link performance – there are many UDP Packet train methods to measure performance that are superior – and K4 can consider this in the future.

The N+1 D/L rate calculation suggested here is the following:

N+1 D/L Rate = (FileSize\*8)/(D/L Time - 4\*RTT);

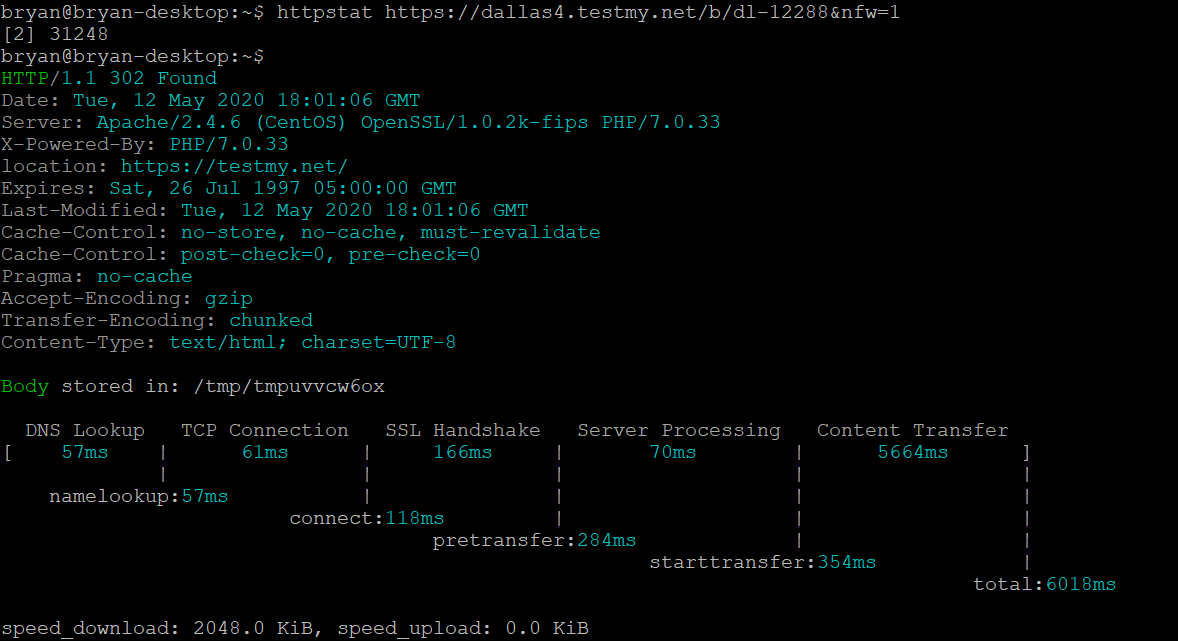
asserting the DNS time is not included in the D/L test, if DNS IS included in the D/L time, the calculation should be:

N+1 D/L Rate = (FileSize\*8)/(D/L Time - 6\*RTT).

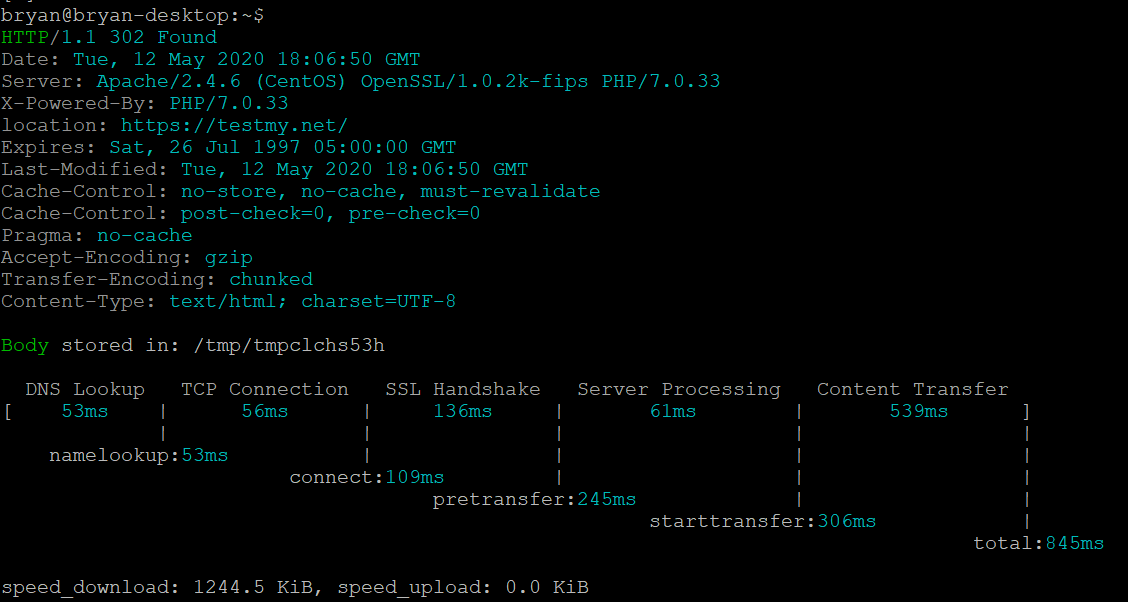
With the design… Middle of the night, 12 MB/hour would be the maximum downloaded and could be reduced if WAN D/L idleness is there.

Another consideration is to utilize the application “httpstat” which can provide D/L and U/L speed of the transfer by removing the dns, tcp, ssl and server processing time and thus only utilize the file transfer time – httpstat is a curl wrapper application. The application works for unsecure and secure sights, and can test against our loadtest server or a fast public server.

Notice here when using testmynet with a single file download (12MB file):



The rate is 16 Mbps using the httpstat rate, and when I use the 5.66s for 12MB I get 17.1 Mbps. Note that my home internet (which is nominally 40 Mbps).



When testing w/ a 1 Mbps file, the rate is closer to 10 Mbps, when I use 539ms for a 1MB file I calculate 14.8 Mbps. I would suggest that K4 takes the “content transfer time, and calculates the rate based on content transfer minus 2x RTT to remove slow start from the equation.

N+1 D/L Rate (httpstat) = (FileSize\*8)/(D/L Time - 2\*RTT).

A better method to consider would be a UDP Iperf test that would send small 100KB bursts at prescribed rates to determine minimal N+1 client tput. Please read the appendix on a method to develop this capability – this should be utilized to best measure link performance w/ minimal data usage.

Thus suggested here is to measure WAN Performance with three methods; first monitor each active WAN link’s D/L data rate in 5s windows (using tcpstat), second ‘periodically’ run a N+1 D/L test from BOX to estimate the active WAN link performance, and third monitor vessel client connections on each WAN link that are between 2-5 MB in size and estimate the client D/L rate. The WAN link performance is an estimate on vessel D/L speeds, but as well a measure of D/L quality to the client. The obvious challenge is that the K4 Box cannot continuously D/L files to test rate, and consume vessel data, thus the mechanism measures, samples and estimates loading to provide the “performance bars”. Suggested draft logic is shown here.

A close up of a map

Description automatically generated

Figure K4 Box/Katalyst Performance Bars logic block diagram.

Key elements of Figure 3.

1. All Performance Bar estimates are made from the Box software.
2. The PB estimate starts with measuring each utilized WAN link’s D/L Tput via tcpstat, every 5 seconds. Sum the WAN links’ total D/L Tput.
   1. Can utilize 5s measurement w/ 3 window running average (configurable).
   2. Run through PB estimate every 5s.
3. This value will fall into one of 4 Bandwidth buckets; 5, 4, 3 and 2 Bar buckets.
   1. The rates suggested here are: 5 Bar > 8 Mbps, 4 bar = 4-8 Mbps, 3 bar = 1-4 Mbps, 2 bar < 1 Mbps.
4. With the link rate measured:
   1. Assess WAN Link Occupancy with RTT measurement (Stability metric above).
      1. This is done via Ping data.
   2. If RTT is unstable: > 30% rise of idle-RTT – use current Tput-base Bar estimate. (point 3)
      1. RTT rise is due to congestion, i.e., thus link is fully used and use rate.
   3. If RTT is stable: No RTT rise, next.
   4. RTT rise needs to be configurable – could be set lower/higher.
5. If RTT is stable, no signs of congestion, was a D/L N+1 run in the last 1 minute?
   1. No – keep the same PB from the prior 5s window
   2. Yes – The D/L N+1 rate will drive the PB settings suggested here.
      1. 5 bars, > 1500 Kbps
      2. 4 bars, 1000-1500 Kbps
      3. 3 bars, 600-1000 Kbps
      4. 2 bars < 600 Kbps
6. Move back to 2, wait 5s.

In summary, the PB metric is based on measured WAN link D/L data rates and RTT metrics, both are measured every 5s minimally. Next, K4 Box is measuring N+1 D/L rate every minute the Katalyst application is open and mobile is unlocked (or every 5 minutes if the app is closed/phone locked), and as well measures client D/L rates for larger file transfers whenever available; this metric will be available about every 1 minute asserting Katalyst application use. Concurrent Multi-channel use w/ the K4 adaptive SDWAN will tweak this a bit, but we should be thinking of this from the start.

Table 1 Summary of Katalyst Menubar Box-collected KPI.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Metric** | **Frequency** | **WAN Links** | **Calculation** | **Usage** | **Notes** |
| Ping | 5s | All Active | Running Ave of 3 measures. | Congestion Measure | RTT rise is used to suggest congestion or link stability issues are occurring.  RTT utilized for WAN Link stability color coding. |
| WAN D/L Rate | 5s | All Active | Total WAN D/L throughput, can set Perf Bar starting point. | Measure of Link performance. | Utilize Total WAN D/L tput as a starting point toward Perf Bar sets.  Have similar measure for U/L as well. |
| WAN Client D/L Rate | 5-300s | All Active | Monitor all client D/L activity at a socket level. Estimate D/L rate for mid-sized sockets (1-5 Mbyte) and utilize similar RTT adjusted calculation[[3]](#footnote-3). | N+1 Client D/L performance. | Used to set Perf Bars if WAN link Tput is not causing congestion (RTT rise).  Monitor client data to set Performance Bars; similar to Box Periodic D/L test.  Have similar measure for U/L as well. |
| BOX D/L 1 MB file | 60-300s | All Active | D/L Rate with RTT adjustment.  Use D/L Rate to estimate UX performance and Perf Bar score. | N+1 Client D/L performance. | Used to set Perf Bars if WAN link Tput is not causing congestion (RTT rise).  Periodic test is every 5 minutes, unless Katalyst App is requesting data on API – then every 1 minute.  Have similar measure for U/L as well – assure the tests do not collide. |

#### Data Rates

The other item shown in the device Menubar would be the U/L and D/L performance, i.e., the numbers to the right of the Performance Bars. The numbers are the total Vessel WAN U/L and D/L rates in Mb likely makes the most sense. With the U/L being the top number and D/L being the bottom number.

#### Cloud Usage/Data

The Menubar Performance Bar KPI here can be ship/shore synced to the cloud such that remote Katalyst apps and the Cloud dash will show the Menubar data just the same, with a few minute delay however having the 5s granularity as provided on-vessel Katalyst usage.

#### SDWAN Enhancements

The Menubar Performance Bar KPI can also be utilized to enhance the K4 SDWAN, particularly for K4’s adaptive multilink SDWAN solution. With this data, K4 SDWAN can utilize multiple WAN links simultaneously, cost-assign, weight-assign clients & client-application flows/sockets to a WAN path to intelligent distribute the applied client load. Thus w/ Performance Bars, comes AM-SDWAN. This is critical for K4 success. QoS is another important aspect, with the previous data – SDWAN QOS can possibly support prioritization and pacing of data to allow for improved QoS for high priority/timely data. A separate whitepaper shall cover SDWAN

#### Katalyst WiFi/LAN Monitor

If there is large scale adoption of Katalyst Application, since every client will want OGI K4 Service Bars, K4 can utilize the Katalyst Application to monitor WiFi, LAN and even WAN health. With this, the vessel network topology can be more fully monitored and characterized. As well it is possible to get application usage data, and thus allow for improved AM SDWAN Performance, and Application dashboards for the vessel operator. A separate whitepaper shall cover Katalyst LAN monitor.

## Lock Screen Notifications

Suggested as well to improve service reliability and visibility, the Katalyst application should have a helper application that when the phone is locked or even by default if the phone is started – the helper Katalyst helper application is monitoring the K4 Vessel. Thus the phone always is receiving alerts on or off vessel. The view of the alerts are show in Figure 5. An important aspect is one-click Katalyst application open, and action buttons.

A screenshot of a cell phone

Description automatically generated

Figure K4 Katalyst Alerts shown within Lockscreen Notifications.

## Appendix – Measuring Link Performance using UDP iperf

The challenge in general is how to get a reliable directional throughput measurement using minimal amounts of data. A traditional and simple method is to use HTTP/TCP and perform D/L and U/L tests; while this is relatively “easy” to do, to actually get a reliable test this substantial amounts of data and data streams to measure the throughput reliably (due to TCP slow start, TCP window-stalling, etc. behaviors); as much as 8-12 TCP flows passing a total of 10-50MB of data. Thus, if the idea is to measure the tput frequently (every 30-60s) such methods are a non-starter. So where from here?

The goal with K4’s throughput measurement is to take a measurement passing less than 200KB of data and with an accuracy that is withing 20% of actual. The measurement method is a N+1 performance measurement, that is, there is an applied N-client loading to the system and the katalyst server performs +1 user additional tput test – and ideally measures the idle/available capacity with this measure.

Suggested here is that we we setup a K4 iPerf-Like server/client in the cloud and on the vessel katalyst server. From this, the basic test methodology is:

1. The on-vessel katalyst server iperf client-mode will run a udp iperf test to our public cloud iperf server (Katalyst-Client to Cloud-Server udp data push).
2. This test would provide two data points for the Cloud-Server… First a U/L data transfer test, and second and importantly, the Katalyst client has punched a hole in the NAT chain from Vessel to the Cloud.
3. Thus, the Cloud Server should note SIP/Sport and DIP/Dport, and perform a UDP Cloud-Client to Katalyst-Server udp data push immediately – since there is limited time in which the NAT hole will be “open”.

The basics are shown here.

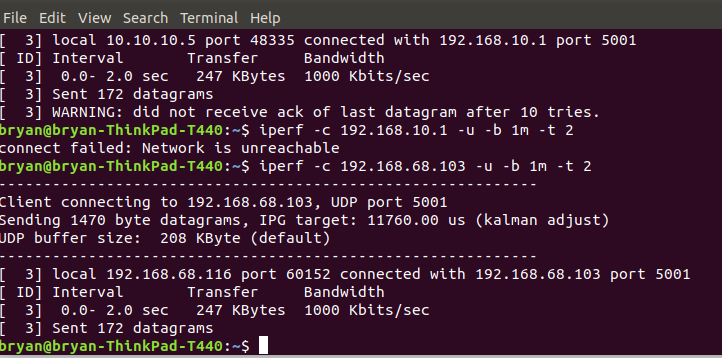
A picture containing text, map

Description automatically generated

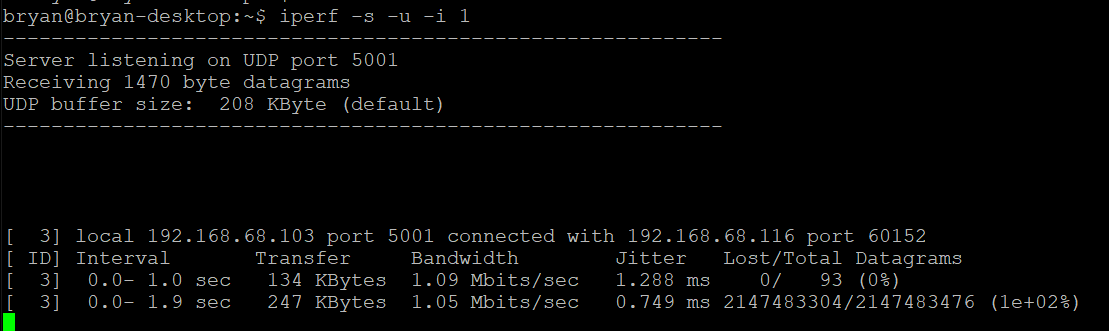
Figure 6 K4 UDP D/L and U/L Performance testing methods.

iPerf likely could be utilized, with some scripting built around it, and one could measure directional N+1 throughput, and have this for WAN link performance. The functionality requires:

1. K4 Cloud system on the Internet and allows UDP/Iperf data from acceptable public IP.
2. K4 Katalyst server on the vessel sends U/L UDP/Iperf test to well known K4 Cloud server IP on pre-determined UDP Dport.
   1. Note – Kat server as a “iperf client” performs a 2s test, send 250Kbyte of data, 1470 IP payload size @ 1Mbps.



1. K4 Cloud system measures U/L UDP test rate, knows what vessel it came from (can put vessel name in payload?).
   1. Note – Cloud server as a “iperf server” provides 1s metrics on data rate and loss, confirms 1 Mbps receive rate and no loss.



1. K4 Cloud server would echo’s back D/L UDP test to same ports, and perform a similar set of tests.
   1. Could be a higher D/L UDP push stream – such as 5 Mbps for 2 seconds. This requires more discussion.
2. Noted here that this is a two second duration sending 1 Mbps of UDP data. Now if the loss in both directions is near zero, we can say at least that at that time there was “at least” 1 Mbps of idle capacity.
3. The Performance test provides two critical data values.
   1. FIRST: provides the N+1 “5 perf bar” rate, which here I am stating is if the rate is > 1Mbps, the it is 5 bars. This requires a very light usage, however gives the “good/bad” assessment.
   2. SECOND: provide N+1 Channel rate, here the UDP Iperf must perform a larger Tput test (e.g. 10m -t 5), and from this add the Iperf usage w/ the usage of client data to get a total link rate and performance of the link. Possibly this is performed every 15-60 minutes, while the small test is performed every 30-60s.
4. Other options to consider on iperf udp send tests…
   1. Can vary the test depending on WAN state.
      1. If there is only one WAN link, only perform the light good/bad test.
      2. If there are multiple links, utilize a different test depending on the link type (LTE send at larger rates, VSAT at lower rates – based on CIR).
   2. Monitor the receive behavior on the server side of inter-arrival time/variance.
      1. Note that the iperf server output has jitter and loss is proved every 0.5s provided, and if we monitor each packets interarrival time and monitor packet loss on the ½ second timescale, one can reliable determine “link performance” wrt Tput but also how close the system is to congestion – IF close, loss rates will be more visible in each ½ second window, also packet inter-arrival will be more spread.

Conclusions. Strongly suggest we test link performance in the following way.

1. Setup UDP Iperf functionality as stated here (katalyst and cloud server).
2. Multiple active links available.
   1. Run U/L and then D/L on each link, randomize ideally – not running a the same time.
   2. Run test every 30 seconds on each link.
   3. iPerf Metric collection every 500msec.
   4. Test for 0.5 Mbps tput for U/L for 2s (this is ~125KB/test)
   5. Test for 1 Mbps tput for D/L for 2s (this is ~250KB/test)
   6. Have the results on tput/loss drive 1-5 bars.
      1. If rate is the same as target, no jitter, and no loss (5 bars)
      2. If rate is lower, jitter higher and with loss (bars reduced – can discuss).
   7. Run a larger test every 15 minutes 1 Mbps tput U/L 5s (600KB), and 5 Mbps tput D/L 5s (3MB).
3. Single active link available
   1. Run test every 60 seconds on each link.
   2. iPerf Metric collection every 500msec.
   3. Test for 0.5 Mbps tput for U/L for 2s (this is ~125KB/test)
   4. Test for 1 Mbps tput for D/L for 2s (this is ~250KB/test)
   5. Have the results on tput/loss drive 1-5 bars.
      1. If rate is the same as target, no jitter, and no loss (5 bars)
      2. If rate is lower, jitter higher and with loss (bars reduced – can discuss).

With minimal data usage, iperf can provide accurate link performance and congestion characteristics that cannot be determined from a TCP test – it is needed to assess. Further, and down the road, I would suggest we develop a K4 Iperf that is purpose built for this type of testing and we improve this yet further.

## Test case using iperf, and a client/server running ubuntu and inter-connected via WiFi.

The test here discusses the previous ideas to determine actual reliable link rate. The test utilized two ubuntu systems – both connected via wifi. The initial test ran a TCP download test from the server to laptop, and the peak rates observed were 4.5-6 Mbps.

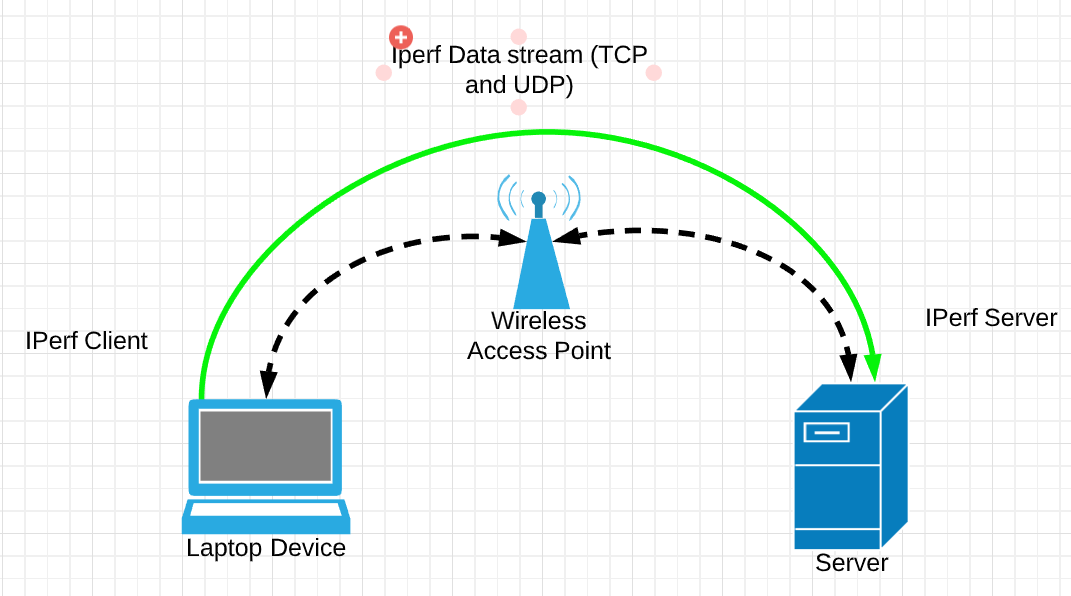


Figure 7 Test System utilized for iperf validation.

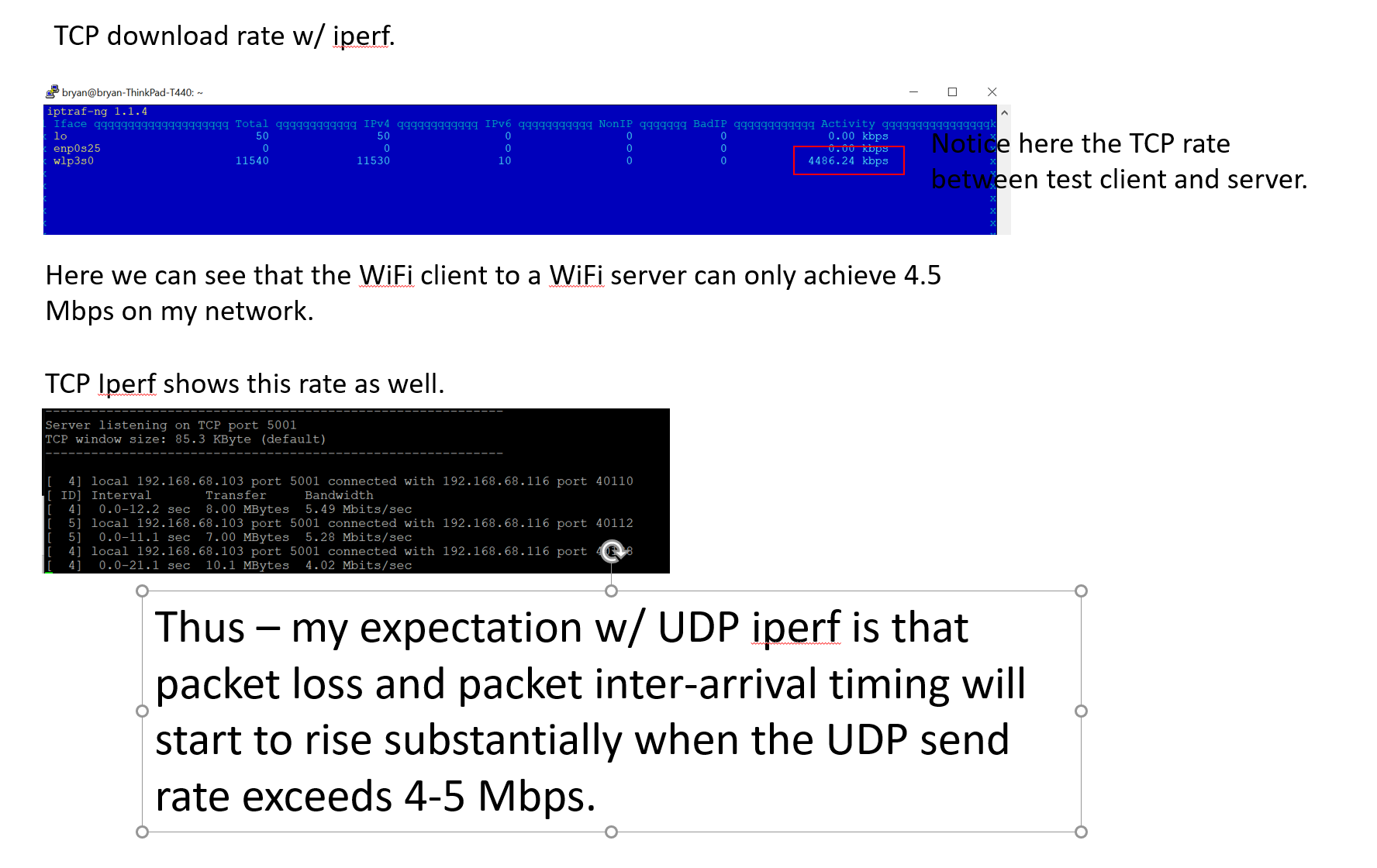
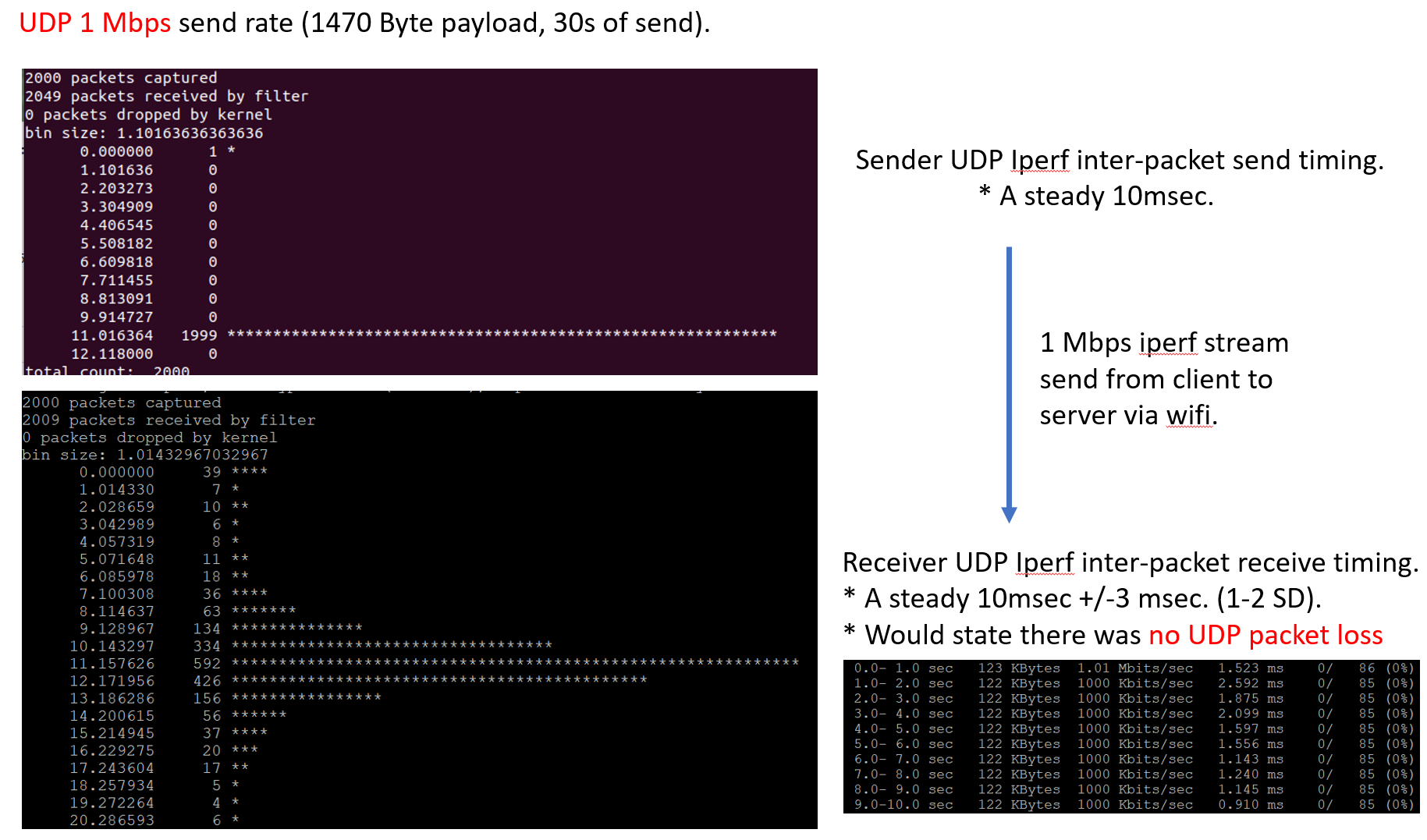


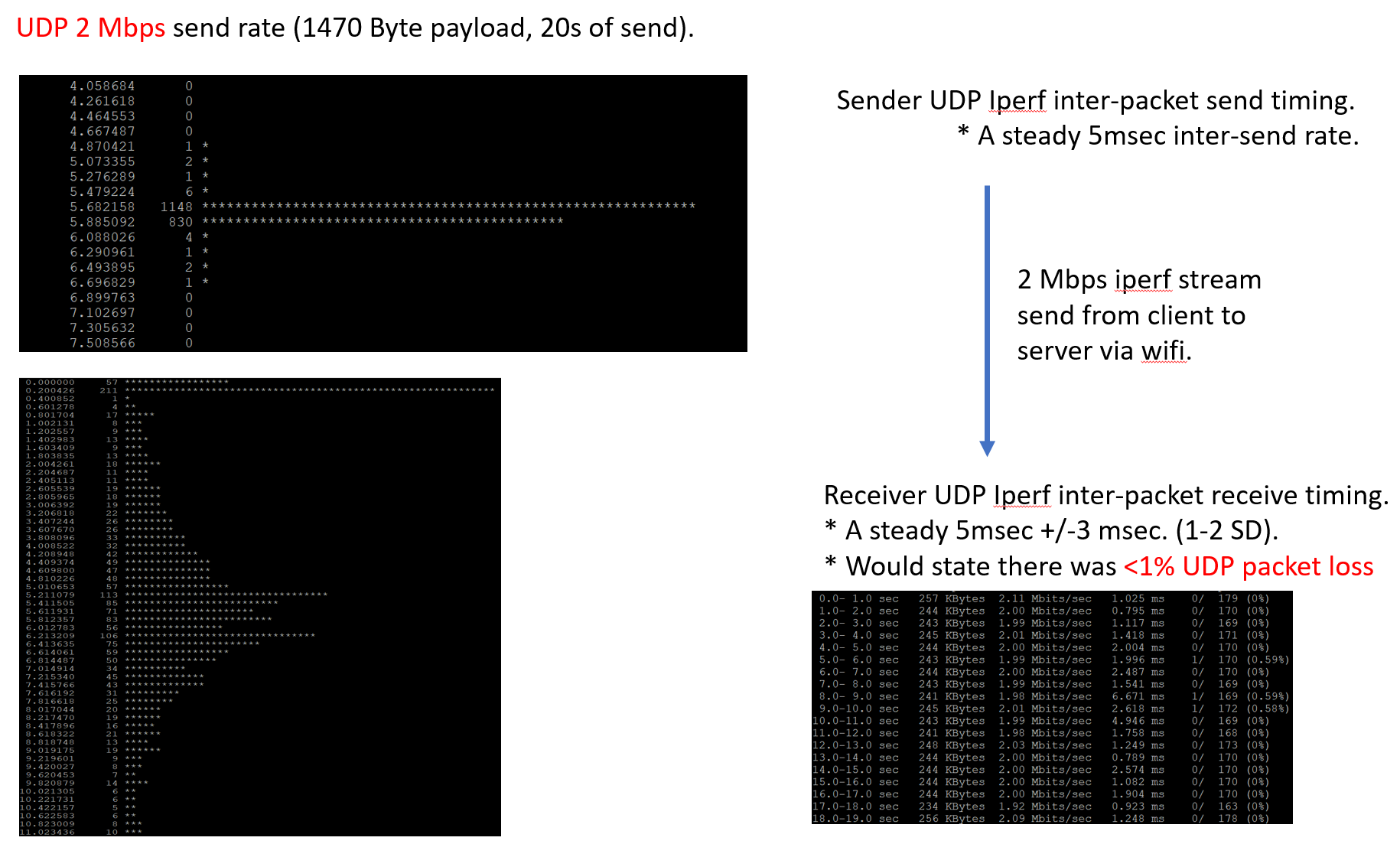
Figure 8 TCP IPerf summary

Now on the surface, I would not expect much more then 5 Mbps of reliable UDP tput for this link, and is this what we observe? Yes

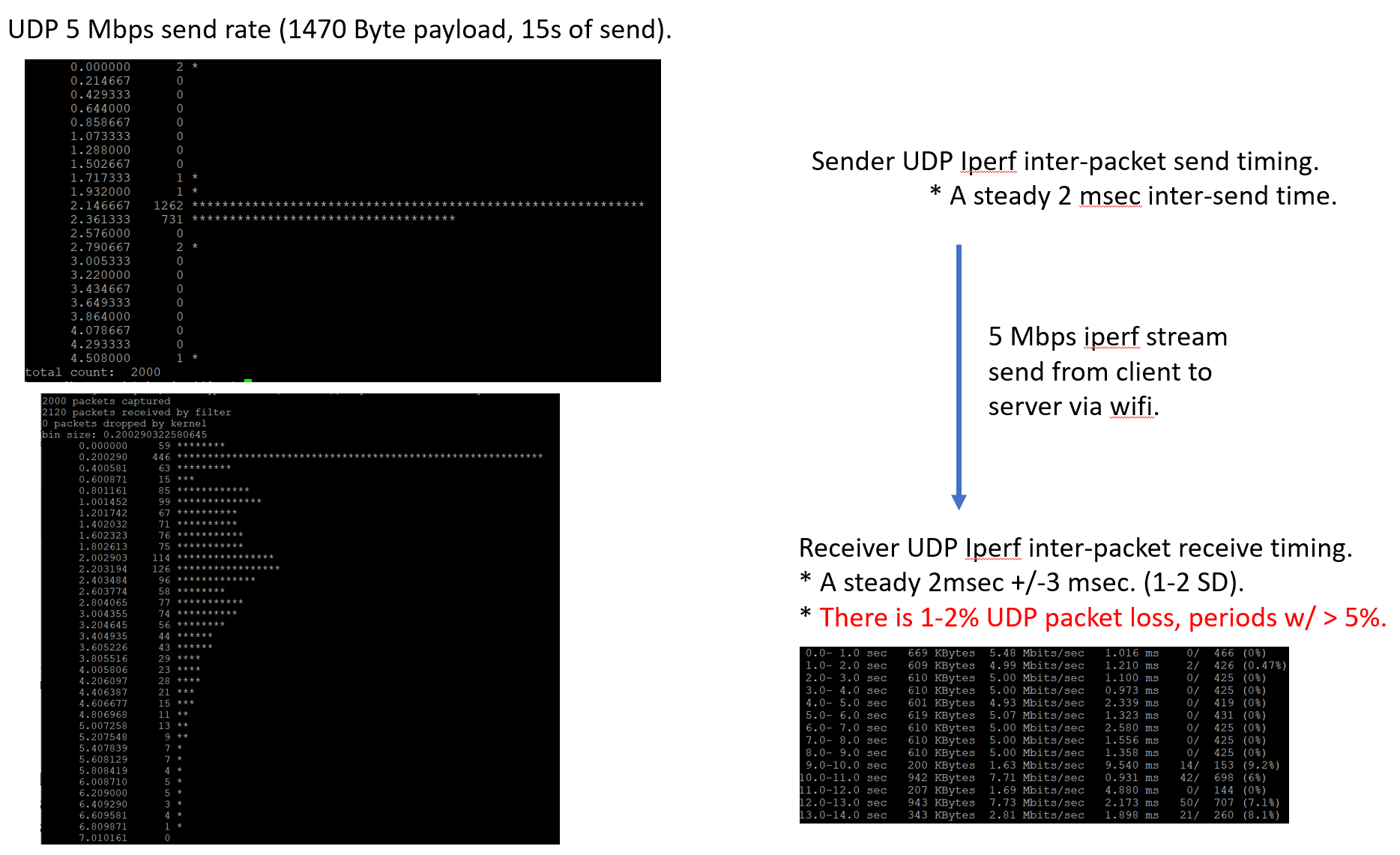
### 1 Mbps UDP Iperf test (from laptop to server)



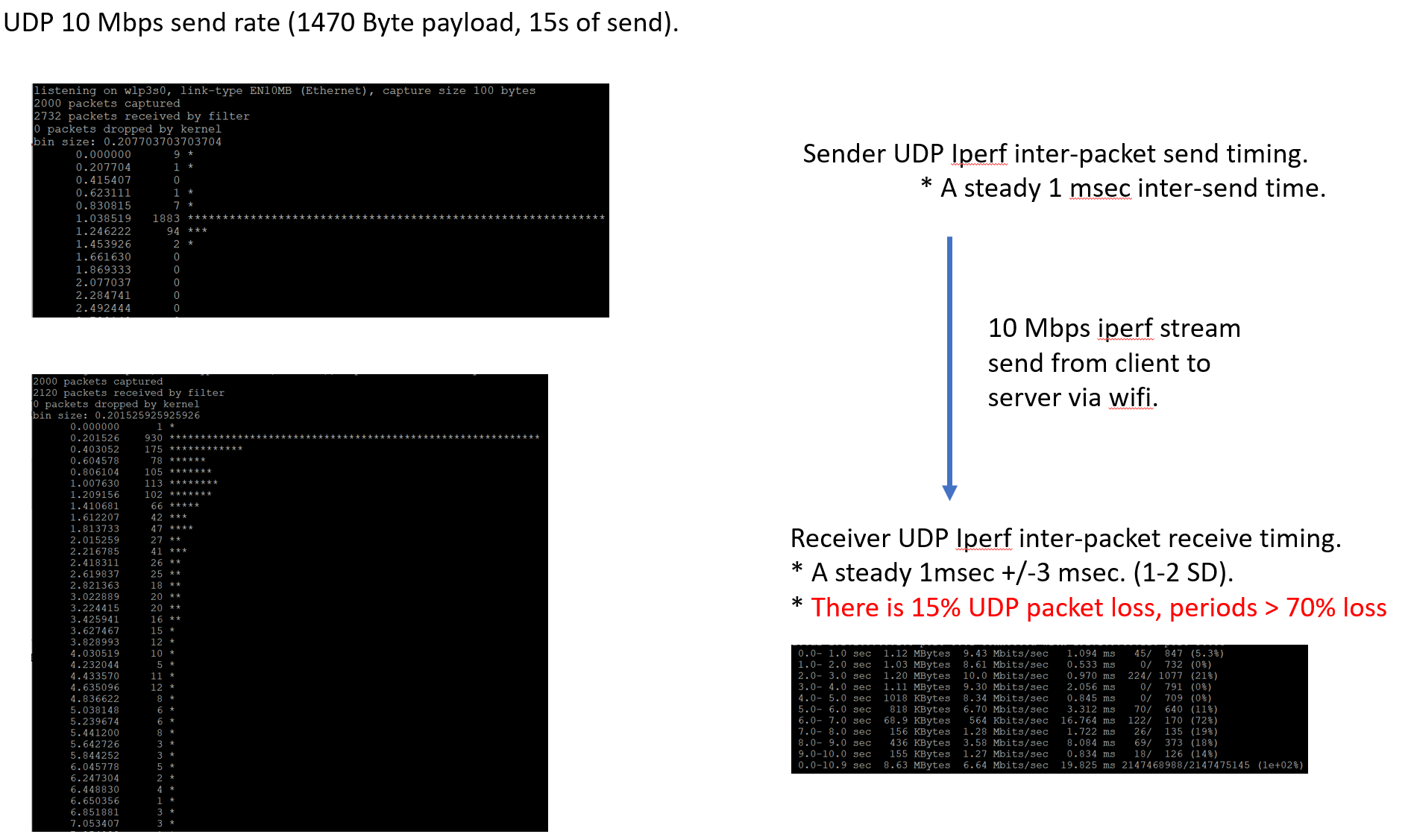
### 2 Mbps UDP Iperf test (from laptop to server)



### 5 Mbps UDP Iperf test (from laptop to server)



### 10 Mbps UDP Iperf test (laptop to server)



1. This should be true if on/off vessel with some limitations. [↑](#footnote-ref-1)
2. It is critically important that the Katalyst API are available for a “Katalite” application or 3rd party applications via on-vessel and internet via licensing key. Thus, third party apps could license to use the service API. [↑](#footnote-ref-2)
3. There are many methods to measure client D/L rate, could use Per-Client accounting and estimate client D/L rate for floating 5 second windows, with some minimum received volume per time to allow the measurement. This has several weaknesses, but could be a simpler method. [↑](#footnote-ref-3)