# show link stats all

## A Comprehensive Analysis

I

# Introduction

A key to understanding SkyPilot link and other elements of system performance can be gleaned from the statistics provided from the **`show link stats all`** command. There is a wealth of information in the output this command generates. This paper will look at the individual elements that comprise this output and provide some suggestions as to how to view the information presented and apply that to network performance.

To derive the full-benefit of the data and information obtainable by this command it is suggested that both the parent node and the child node, in question, be logged into simultaneously so as to compare the data. This will provide the greatest depth of information for some aspects of link and system performance. The reasoning for this will become more apparent definitions of the various components of this output are provided.

# `show link stats all` – A First Look

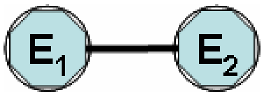
The full output of the command is displayed below. The sections following will present a detailed look at the pertinent statistics contained within this output.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  14   18     24     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  27   24     18     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        0
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191       32
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         0

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  0           6696        588         6           58822        0
00:0a:db:01:0a:4a data  20077       1664435     4438        64          20373152     0
00:0a:db:01:0b:18 data  5           20183       367         5           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ---------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

The output is divided into several statistical areas which include link overview, queue performance, receive and transmit information. As this paper progresses it is important to note that all statistics presented here are link statistics and represent information about the specific link only. They do not provide specific data about the node performance, although node performance information can be derived from the information presented.

# Transmission and Reception – An Overview

A good starting point for analyzing the output of **`show link stats all`** is a discussion of the primary transmission and reception functions of a parent node and its child node, as well as a basic discussion of the things that can go wrong, or interfere, with the transmission process. Understanding the process will go a long way toward accurately interpreting the information available from the **`show link stats all`** command.



In the perfect radio environment $E_1$ will schedule and transmit a frame to $E_2$. When $E_2$ receives that frame it will send an ACKnowledgement (ACK) to $E_1$ indicating the frame was received. Upon receipt of the ACK, $E_1$ will schedule and transmit the next radio frame to $E_2$ as soon as a timeslot is available. This process is the same for transmission to all children of node $E_1$.

In the reverse, if $E_2$ has data to transmit to $E_1$, $E_2$ will send a contention request to $E_1$ requesting permission to send data. $E_1$ will respond to that request and schedule $E_2$ for transmission. $E_2$ will transmit the radio frame to $E_1$. $E_1$ will respond to $E_2$ in kind with an ACK.

If $E_1$ has a large number of children all requiring timeslots to transmit and receive data, the time delta between scheduled slots will become larger. This increased time between actively scheduled timeslots for a child node will increase latency. As the total number of children requiring timeslots continues to increase, the resultant latency will increase. At some point, the increased burden on the scheduling engine will cause the transmit queues for a link to begin to fill. This can result in dropped data in the queues.

In the instance where RF interference becomes a factor near either $E_1$ or $E_2$ one of the transmissions - the original radio frame or the ACK - may be lost. In either instance, the transmitting node will perceive the response as requiring the original frame to be retransmitted. This occurs until either an ACK is received, or the transmission has been repeated five times, at which point it will be discarded.

Combining the two scenarios above – large numbers of children requiring timeslots combined with RF interference – will cause even greater latency on the network. This will cause more backup in the transmission queues and increase the chances of more data being dropped as a result. At a minimum the increased latency will impact throughput and performance for everyone downstream of the impacted links.

Let's start by looking at the various elements in the **`show link stats all`** output, providing definitions for them and relating them to link and bandwidth performance. As we're going through this analysis keep in mind the process of transmission ➔ ACK as the basis for much of the information presented. This transmission ➔ ACK sequence is fundamental to understanding the interrelationship of many of the statistics presented.

## *Link Summary Statistics*

The first group of statistics presented by the **`show link stats all`** command display link summary information. These can be seen below. These statistics summarize the basic elements of link performance by presenting RSSI values, modulation rates, transmitted and received frames and transmitted and received packets.

The output of this section is shown below. Immediately following is a definition of all the statistics contained within this section. The statistics of "most interest" are highlighted.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  14   18     24     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  27   24     18     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096
```

| Parameter | Definition |
|---|---|
| MAC Address | Specifies the MAC address for the current link of interest. All data transmitted (Tx) or received (Rx) is with respect to this MAC address. |
| LType | Specifies the Link (L) type (Type). This field will always display "data". |
| RSSI | Specifies the Receive Signal Strength Indicator (RSSI) for the current link. This is the same as LRSSI displayed by other CLI commands. |
| LTxMod | Specifies the Local (L) Transmit (Tx) Modulation (Mod) rate. This is the TRANSMIT modulation of the node you're logged into. |
| RTxMod | Specifies the Remote (R), Transmit (Tx), Modulation rate. This is the RECEIVE modulation at the node you're logged into. |
| TxPow | Specifies the Transmit (Tx) Power (Pow) for the node. This value can be ignored as it is fixed across the system. |
| RxFrames | Specifies the total number of SkyPilot radio frames (Frames) received (Rx) on this link. |
| TxFrames | Specifies the total number of SkyPilot radio frames (Frames) transmitted (Tx) on this link. |
| RxPackets | Specifies the total number of Ethernet data frames (Packets) received (Rx) on this link. |
| TxPackets | Specifies the total number of Ethernet data frames (Packets) transmitted (Tx) on this link. |

**RSSI** indicates the receive level of the incoming signal. This is a loudness indicator and not a quality indicator. This value should be 25 or higher. RSSI's lower than 25 indicate susceptibility to interference on the link. Above 25 and the links should operate stably.

**LTxMod** and **RTxMod** indicate link integrity, or quality. When logged into a node LTxMod represents the modulation used for transmission from that node to the far-end node. RTxMod represents the modulation used for transmission from the far-end node to the near-end. TxMod values represent link integrity and may be used to assist in determining the location of destructive interference within the network. Modulation rates will be lowest at receiver(s) experiencing the interference.

**RxFrames**, **TxFrames**, **RxPackets** and **TxPackets** represent the total number of Frames (radio) and Packets (Ethernet) transmitted and received across the link. These values are incremented whenever a new radio frame (TxFrames & RxFrames) or Ethernet Frame (TxPackets & RxPackets) are transmitted, or received across the link).

## *Receive Statistics*

The receive statistics log information about radio frames received on the link. When a counter increments in this area it represents RF interference at one end of the link or the other. The key to understanding which node is experiencing issues lies in understanding the key data in this output.

```
MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        0
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191       32
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         0
```

| Parameter | Definition |
|---|---|
| MAC Address | Specifies the MAC address for the current link of interest. All data transmitted (Tx) and received (Rx) is with respect to this MAC address. |
| LType | Specifies the Link (L) type. For active links this will always display “data”. |
| RxBytes | Specifies the total number of SkyPilot bytes (Bytes) received (Rx) on this link. This value includes SkyPilot overhead bytes. |
| TxBytes | Specifies the total number of SkyPilot bytes (Bytes) transmitted (Tx) on this link. This value includes SkyPilot overhead bytes. |
| RxMalformed | Specifies the total number of radio frames received that had malformed receive headers; results from corruption from external elements. In order for this counter to roll the header MUST be decodable. |
| RxDuplicate | Specifies the total number of duplicate radio frames received on this link. Radio frames have sequence numbers used to accumulate this data. Means that far-end did not receive the ACK response. |
| RxMissed | Specifies the total number of radio frames that were missed during transmission. Sequence numbers are used to accumulate this counter. |
| Queue Drops | Specifies the total number of queue drops due to the queue overflowing. This occurs AFTER a highly aggressive RED algorithm could not keep up. |

**RxBytes** and **TxBytes** are incremented every time a radio frame is transmitted or received by the node. The values in this counter represent the total number of data bytes plus radio overhead received or transmitted at the node, on the link in question. This value is not an accurate reflection of the total bytes transmitted to, or received from, a given user on the network even when these values are read directly from the SkyConnector.

**RxDuplicates** are logged when a frame is received more than once. Retransmissions will occur whenever the transmitting node fails to receive an acknowledgement for a transmission. All radio transmissions have sequence numbers embedded in the radio frame – this is how both the RxDuplicate and RxMissed statistics are derived. If there is a skip in sequence numbers an **RxMissed** will be logged.

**Queue Drops** occur whenever the inbound data rates exceed the link queue capacity. SkyPilot nodes employ RED to help alleviate queue overflow. When RED fails to control the queue ingress rates, Queue Drops will result.

## *Transmit Statistics*

The Transmit Statistics log information for radio frames transmitted on the link of interest. When most of the counters increment, in this area, it represents RF interference at one end of the link. The key to understanding what is being impacted by the interference lies in understanding what the various components of this data mean.

```
MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  0           6696        588         6           58822        0
00:0a:db:01:0a:4a data  20077       1664435     4438        64          20373152     0
00:0a:db:01:0b:18 data  5           20183       367         5           941392       0
```

| Parameter | Definition |
|---|---|
| MAC Address | Specifies the MAC address for the current link of interest. All data transmitted (Tx) and received (Rx) is with respect to this MAC address. |
| LType | Specifies the Link (L) type. For active links this will always display “data”. |
| RED Drops | Specifies total number of Random Early Discards occurring in the transmission queue for this link. RED Drops occur in anticipation of Queue Drops. |
| TxRetries | Specifies the total number of Transmission (Tx) Retries (Retries) that occurred on the link. TxRetries increment when no ACK is received for the transmission. |
| TxRLimit | Specifies the total number of transmission (Tx) retries (R) that exceeded the retransmission limit (Limit). Counter increments after five unsuccessful transmissions. |
| TxAborts | Specifies the total number of aborted (Aborts) transmission (Tx) attempts due to modulation rate changes. Occurs when modulation rate drops for the link. |
| TxDataFrames | Specifies the total number of frames (Frames) transmitted (Tx) from parent node to a child node indicating that there is data (Data) to follow. On a child node this parameter specifies the total number of responses to the parent node that have been transmitted. |
| TxAcks | Specifies the total number of transmission (Tx) acknowledgements (Acks) sent to the parent node. This statistic will only be calculated on one of the many potential links. |

**RED Drops** are a holdover from the Queue Drops statistic in the previous group. Link queues are maintained at adequate levels by a RED (Random Early Discard) algorithm. Under normal operating conditions RED Drops should be few and far between. When congestion increases RED Drops will begin to increment and the RED Algorithm will become more aggressive. Whenever the queue ingress rate significantly exceeds the queue egress rate Queue Drops will occur.

**TxRetries** occur whenever an ACK is not received for the last radio transmission. Every radio frame transmitted requires a corresponding ACK, acknowledging receipt of the radio frame. If no ACK is received, then the node will retry until either the transmission is acknowledged or it has reached five consecutive attempts.

**TxRLimit** occur after five consecutive attempts to send the same radio frame. This can mean that interference is at the far-end node (receiving the transmission) or the near-end node (receiving the ACK’s).

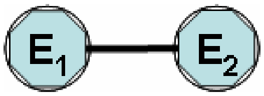
**TxAborts** occur whenever a radio frame is scheduled for transmission and a dynamic modulation rate reduction occurs on the link. The resulting radio frame, from the lower modulation rate, may be smaller than the queued radio frame. When this occurs, a TxAbort is registered for the link.

# Tying It All Together

Now that some basic definitions and conditions have been presented for the statistical elements, let's take a look at them in some detail and relate them to issues that can occur on a wireless network. As we progress through this section we're going to expand on the definitions presented above and show the specific circumstances through which the counters within the system will increment.

## *First a Review*

Let's do a quick review of the basic structure of wireless radio transmission within the SkyPilot Network. We'll be using the same diagram as in the start of this paper. Let's start with a transmission from $E_1$ to $E_2$. When the data is received by $E_2$, $E_2$ will transmit an ACKnowledgement (ACK) back to $E_1$ indicating a successful transmission.



If interference occurred at $E_2$ during the reception of the data, $E_2$ would not send the ACK to $E_1$. If interference occurred around $E_1$, $E_2$ would transmit the ACK but it would not be successfully decoded. Either condition results in $E_1$ repeating the transmission of the original radio frame during the next scheduled transmission timeslot for $E_2$. This "failure" can occur up to five times. After the fifth failure to transmit the data, $E_1$ would drop the transmission frame, increment the TxRLimit counter and attempt to send the next frame in the queue. The same sequence of events occurs when $E_2$ transmits to $E_1$.

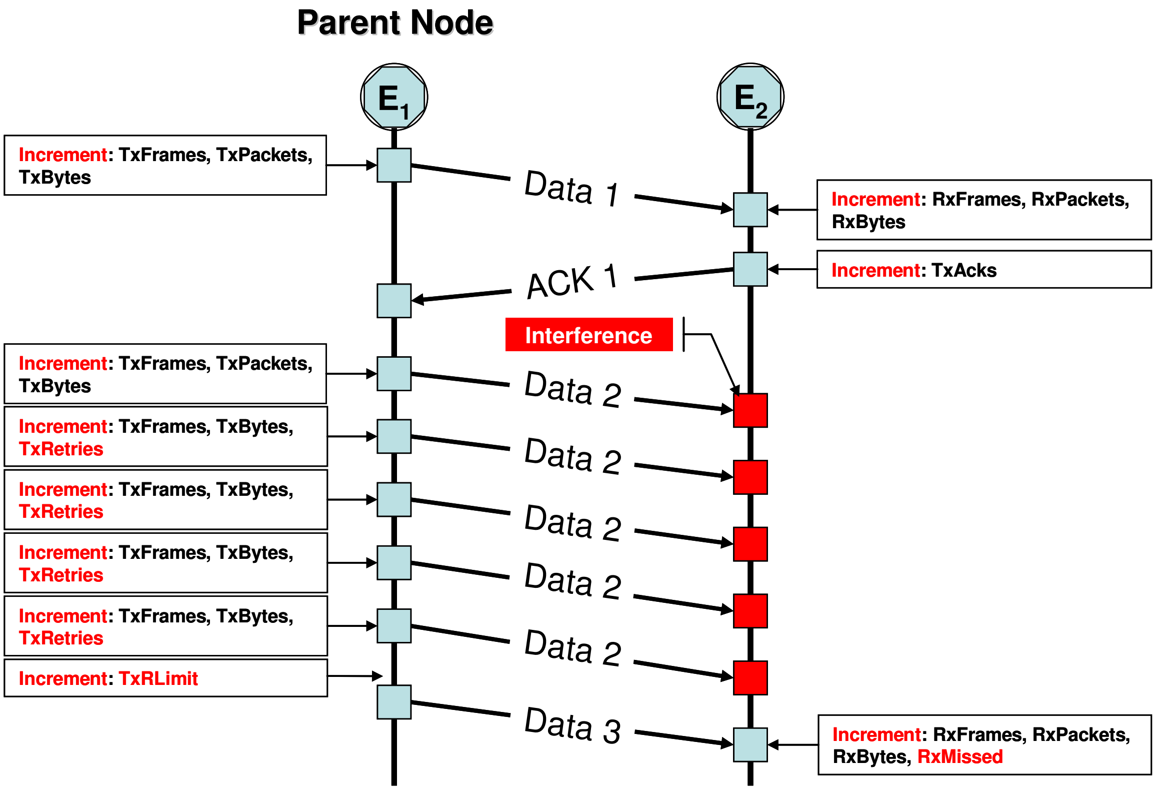
## *Things to Keep in Mind*

There is more that happens, during this retransmission phase, than what is considered at first glance. We will start by taking a closer look at nodes $E_1$ and $E_2$ through the use of a timing diagram. We will also look at how the statistics evolve, on each side of the link, depending on where the interference occurs on that link. We will then progress to a discussing the impact of retransmissions on a busy link, where queues are beginning to populate. Finally, we will provide a procedure that can be used to evaluate both link and bandwidth manager performance on the network. This will be one approach that can be used. It is hoped that the information presented herein will provide the reader with ideas for other approaches to mesh analysis.

It is always important to remember what side of the link each node is on. This point of reference, when viewing the statistical data presented, is of utmost importance for the proper interpretation and analysis of the information and localization of any issue. We cannot stress the importance of this enough. Once this point of view is taken during a few opportunities for analysis it will become a force of habit.

# Node Behavior - Interference at the Far-End

The first scenario we'll take a look at is analyzing the `show link stats all` output for interference at the far end of the link. In this case, the parent node will be the unit that you're logged into. It will also represent what we call the "near-end" of the network – the one closest to you, in terms of where you're logged into and performing the primary analysis from.


Parent Node
$E_1$
$E_2$
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Interference
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: TxRLimit
Data 3
Increment: RxFrames, RxPackets, RxBytes, RxMissed

The diagram above comprises what is commonly called a bounce diagram. The vertical scale represents time and the vertical lines below nodes $E_1$ and $E_2$ display events, with the most recent at the bottom and the oldest at the top. The arrows, left-to-right, represent the direction of data travel and their downward slope indicates time has elapsed since the initial event. Spaces between the events (shown as either blue or red boxes) indicate internal processing time on the node where the event occurs (e.g. time interval between the reception of Data 1 and the transmission of ACK 1), or the delta time between events (i.e. where $E_2$ is receiving Data 2 repeatedly).

Viewing the events that occur as a result of interference at $E_2$, or the far-end, a lot can be determined about the link performance in the RF spectrum. The following explanation assumes that you are logged into the node $E_1$, or the near-end. The first two events on the bounce diagram (Data 1 and ACK 1) indicate a successful transmission from the parent node $E_1$ to the child node $E_2$.

When the parent transmits the radio frame it will increment the link counters for **TxFrames** (SkyPilot radio frames), **TxPackets** (Ethernet frames) by the total number of Ethernet frames concatenated in the transmission frame and **TxBytes** which reflects the total number of transmitted bytes in the entire transmission frame. When the frame arrives at the far-end ($E_2$), $E_2$ should receive the frame, successfully decode it and then log statistics for **RxFrames** (SkyPilot radio frames), **RxPackets** (Ethernet frames), **RxBytes** (total bytes received in the transmission) and when the acknowledgement for the data is sent **TxAcks** will be incremented. This process is outlined by the process of Data 1 and ACK 1 in the above diagram.

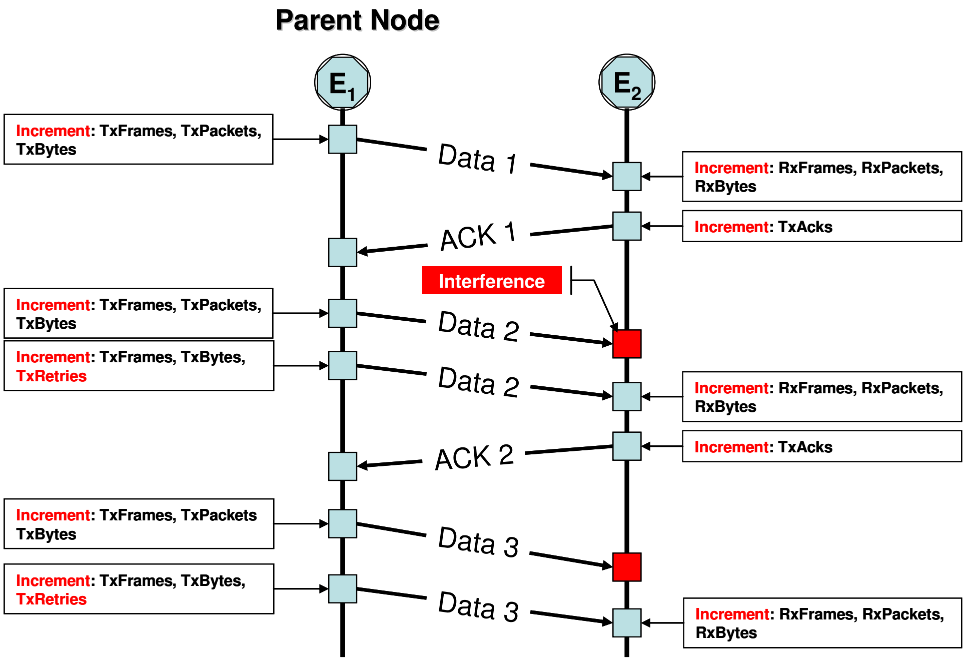
When RF interference occurs at the far-end ($E_2$) statistic counter behaviors will change. The RxFrames, RxPackets and RxBytes counters will not roll if a radio frame cannot be successfully decoded during reception (i.e. it is errored by interference during reception). Remember that interference does not impact the transmitter, or the transmission, it only impacts the receiver.

At the start of the Data 2 transmission, the parent node $E_1$ will transmit the radio frame. When this occurs $E_1$ will increment the TxFrames, TxPackets and TxBytes counters to reflect the current radio frame transmission. The packet will arrive at $E_2$ but interference will prevent it from being decoded by $E_2$. $E_2$ will not log any statistical data for the reception of the frame due to its failure to properly decode the transmission. As a result it will not send an ACK back to $E_1$ for this transmission.

The next time $E_1$ schedules transmission to $E_2$ it will resend the original radio frame. This time, it will only increment TxFrames (radio frames), TxBytes (total bytes in transmission) and **TxRetries.** The TxRetries counter only occurs on the 2nd, 3rd, 4th and 5th transmission of the same radio frame. TxRetries indicate that no ACK was received from the far-end of the link. In this example no ACK was transmitted due to inability to receive, and decode the transmitted radio frame as a result of interference local to the far-end of the link ($E_2$).

After the fifth attempt to transmit the radio frame with no ACK being received, the near-end node (in this case $E_1$) will cease attempting to transmit the frame. During this fifth attempt the TxFrames, TxBytes and TxRetries counters will be incremented. After the specified timeout period the near-end node will increment the **TxRLimit** counter indicating that five failed attempts to transmit the frame have been reached.

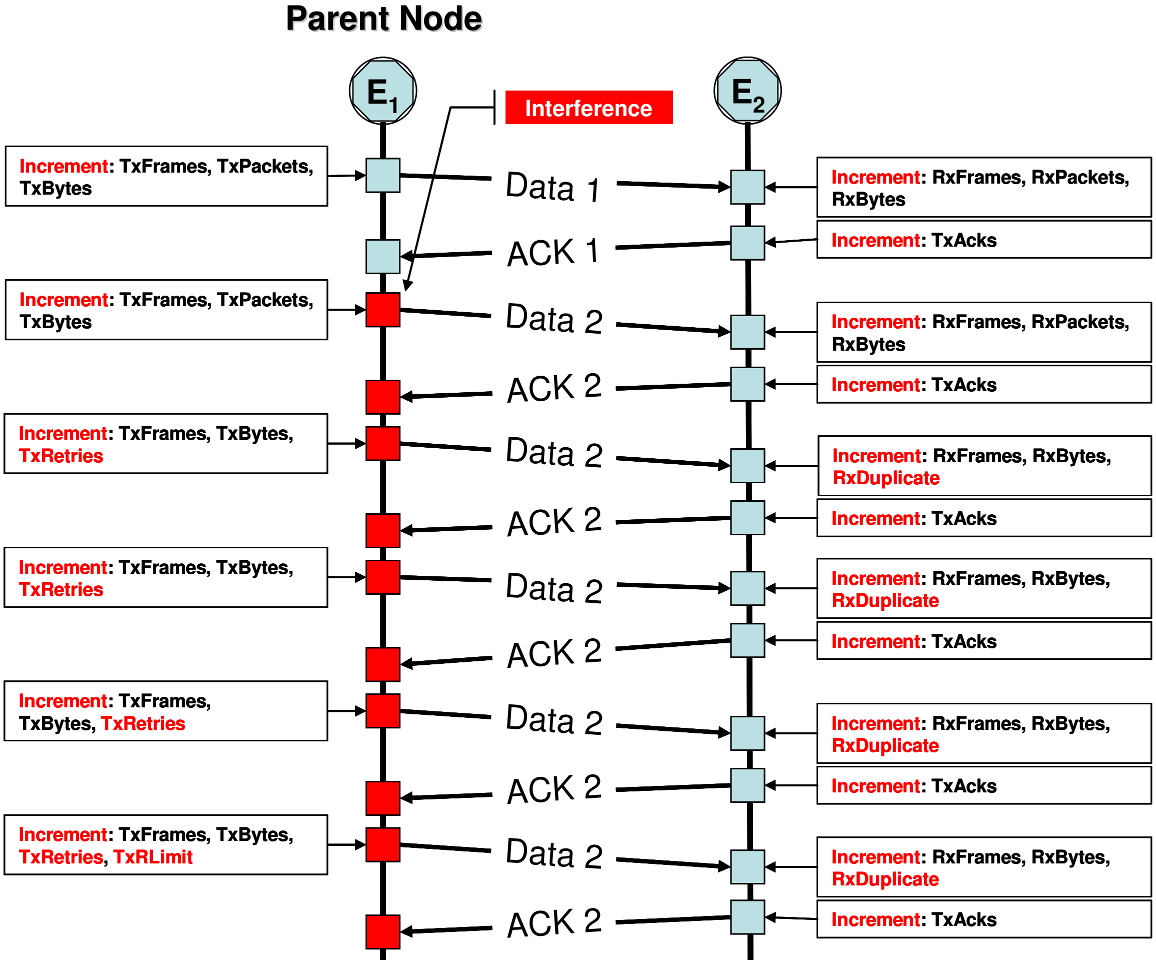
The diagram, to the left displays a similar scenario to the one above, but with the interference only being temporary, in each instance. In this example the TxRLimit is never reached, and the TxRetries counter increments only for each retry attempted. Because the retries are mostly successful the TxRetries counter will not increment at a high rate of speed.


Parent Node
$E_1$
$E_2$
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Interference
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxPackets, RxBytes
ACK 2
Increment: TxAcks
Increment: TxFrames, TxPackets TxBytes
Data 3
Increment: TxFrames, TxBytes, TxRetries
Data 3
Increment: RxFrames, RxPackets, RxBytes

In this example, as the one above, the receiving node $E_2$ will only log Rx statistics when it can properly decode the received radio frame. It will not log statistics for frames it cannot decode.

# Node Behavior - Interference at the Near-End

The next scenario to view is the state where interference is occurring at the near-end, rather than the far end. While the end-result is the same, in terms of the near-end counters incrementing, the far-end counters are behaving much differently than what they were in the previous example. As in the prior example we will start out by assuming the interference occurs for the duration of the five transmission attempts. While this may not always be the case in the real-world, it helps to start here and then work out other scenarios as they come up.


Parent Node
E1
E2
Interference
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: RxFrames, RxPackets, RxBytes
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries, TxRLimit
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks

As can be seen from the diagram above the interference is now at the near-end node, or the parent node. Again, the counters for a successful transmission (Data 1 and ACK 1) are presented. However, the differences in the scenario are readily encountered in the transmission of the Data 2 radio frame. As you can see the interference is already present during the transmission of the frame. As noted above, interference impacts only the receiver and not the transmitter.

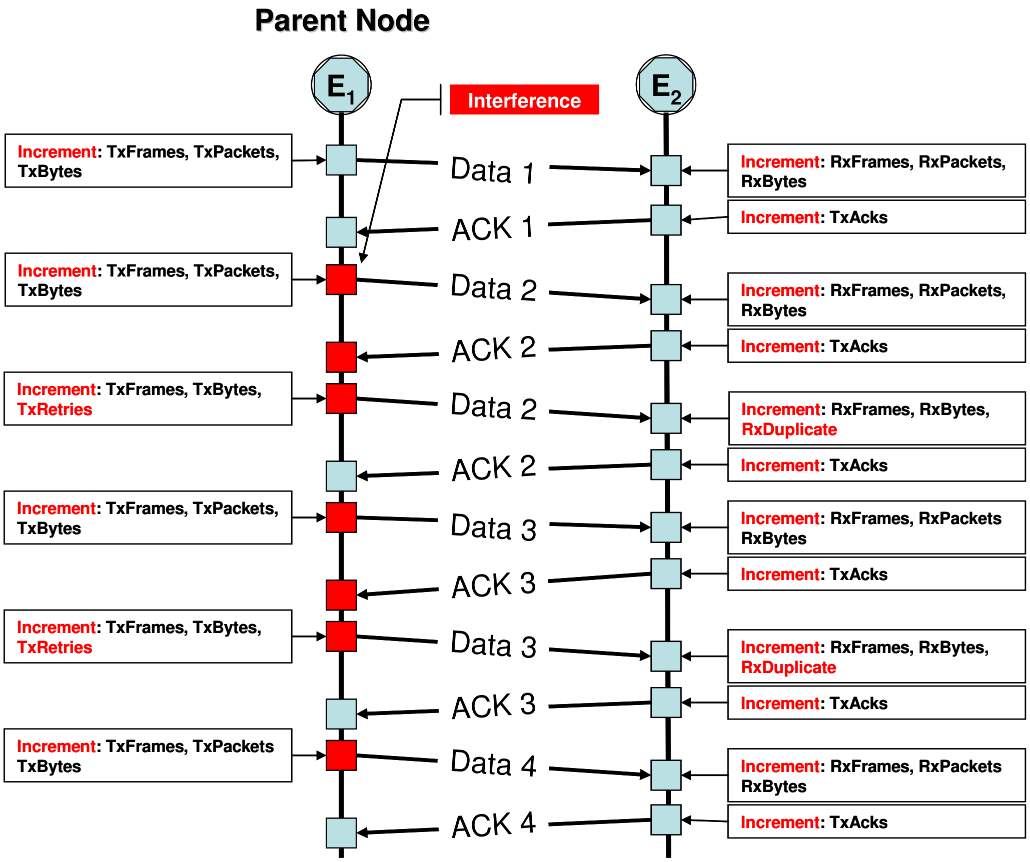
When the far-end node $E_2$ receives the radio frame it will decode it and increment the RxFrames (radio frames), RxPackets (Ethernet packets contained in the radio frame) and RxBytes (total transmitted bytes). The far-end node $E_2$ will then transmit an ACK to its parent node $E_1$. The ACK transmission encounters interference while $E_1$ is receiving the radio frame. $E_1$ cannot decode the radio frame due to the interference and assumes that no ACK was sent.

At the next scheduled transmission timeslot for $E_2$ the near-end node $E_1$ will retransmit the original radio frame. At this juncture it will increment the **TxFrames** (radio frames), **TxBytes** (total bytes transmitted) and **TxRetries** (total number of retransmissions attempted). The far-end node $E_2$ will receive this data. It will log, the second time around, **RxFrames** (radio frames), **RxBytes** (total bytes received) and **RxDuplicate** (total number of radio frames received with the same sequence number). It will generate and transmit an ACK to its parent node and increment the **TxAcks** counter. This process is repeated until either the fifth transmission attempt is reached at which point the near-end node ($E_1$) will increment the **TxRLimit** counter, discard the Data 2 frame and queue up the next radio frame for transmission to the far-end node. At the reception of the Data 3 radio frame the far-node will increment the **RxFrames**, **RxPackets**, **RxBytes** and **RxMissed** counters.

The next example shows the same diagram, but in a situation where the interference is more sporadic. As can be seen, from this example the RxMissed statistic and the TxRLimit statistic does not increase.

The TxRetries and RxDuplicate values are increasing as a result of the sporadic nature of the interference on the link.

From these two examples you can see how interference at either the near-end of the link or the far-end of the link will cause the same symptom on the near-end node – where TxRetries and/or TxRLimit will increase each time the ACK from the far-end node is not received. The lack of ACK response can be a result of interference at either end of the link and it's important to view both ends to determine where the issue lies.


Parent Node
E1
E2
Interference
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: RxFrames, RxPackets, RxBytes
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks
Increment: TxFrames, TxPackets, TxBytes
Data 3
Increment: RxFrames, RxPackets RxBytes
ACK 3
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 3
Increment: RxFrames, RxBytes, RxDuplicate
ACK 3
Increment: TxAcks
Increment: TxFrames, TxPackets TxBytes
Data 4
Increment: RxFrames, RxPackets RxBytes
ACK 4
Increment: TxAcks

## Link Queues – The Basics

Queues are designed into network devices to handle surges in frame arrival rates on a link. Queues are utilized whenever the frame arrival rate exceeds the frame forwarding rate. Frames that are unable to be forwarded are stored in the queue and forwarded as transmission resources become available.

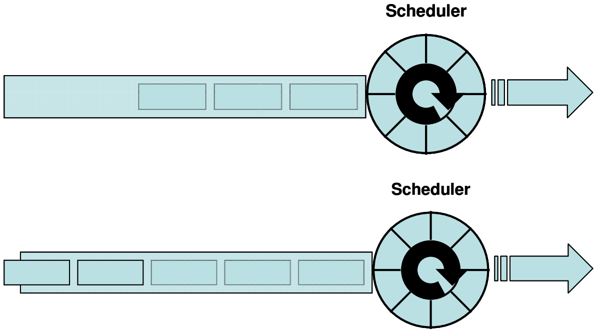
The SkyGateway, for example, has a potential frame arrival rate from the data network of 100Mbps. With perfect links to the SkyGateway, the highest modulation rate achievable is 54Mbps and of the 54Mbps approximately one-half that rate is available for user data throughput. That resource must be alternated between data transmission and data reception reducing the effective unidirectional throughput, or goodput, on any link.

Because frame arrival rates can exceed, by greater than double, the SkyGateway's ability to forward those frames to the mesh, the SkyGateway has transmission queues configured to handle the influx of data frames from the network.

SkyExtenders and SkyConnectors also have transmission queues integrated into their operational structure so that arriving data frames may be stored until they are granted the ability to transmit by their respective parents. In some instances, especially during periods of network congestion, it may be a requirement to store as many as 100 data frames in a given queue that are ready for transmission.

At first glance, the concept of the queue seems to solve a lot of problems – the foremost being loss of data. If there were no queues available and the inbound frame rate was 100 packets per second and the outbound frame rate were 1 packet per second, every one second interval where this high frame rate occurred would represent 99% data loss. With an appropriately sized queue, data loss could be eliminated entirely. But, as in everything, there is a downside to the concept of a queue.

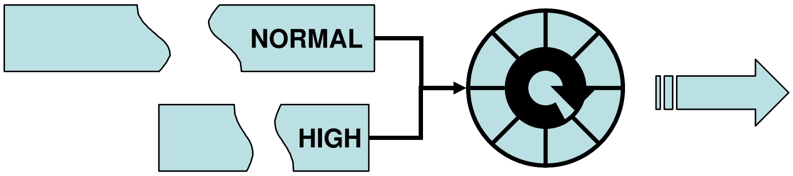
We can liken queues to a road system. If you're traveling on a road during non-busy hours your car (similar to a data frame) moves onto the road system with no resistance and no additional delay as you travel toward your destination. The time it takes for you to travel from your start point to the end-point of the trip is referred to as latency. This one-way trip translates into one-half of what is called the Network Round Trip Time (NRTT). The time it takes to return to the trip origin would be equivalent to NRTT **EVEN IF** you took a different route.



If the highway becomes congested conditions begin to change. On-ramps turn into queues. The more cars on the ramp the greater the "queue depth" becomes. If you're near the back of the line, your "latency" becomes greater as it takes longer to reach the freeway. There's a large difference in the time it takes to gain access to the medium (highway or network) when there are three cars on the ramp versus a hundred cars. The NRTT becomes increasingly great the more frames (cars) that populate the queues.

This traffic analogy illustrates well the conditions that occur on a data or wireless network. The analogy breaks down at the "freeway". In a network system, once you've gained access to the medium the transmission rate remains constant. A congested 100Mbps network will transmit a frame, once it's on the media, at 100Mbps. A congested freeway will suffer a decline in speed once you're on it.

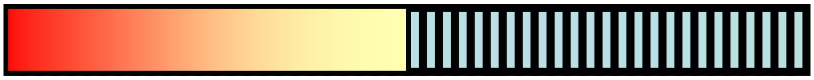
As mentioned above, the SkyPilot nodes have transmission queues, for each link, built in. These queues are designed to handle high frame ingress rates during congestion. When a queue begins to reach a certain capacity it becomes necessary to enact certain controls on that queue in order to slow the arrival rate of the



frames. For TCP transactions which comprise the bulk of internet traffic, the best way to enact queue controls is through a mechanism called Random Early Discard, or RED. While the deeper operational aspects of RED are much too complicated to address in this paper, there are some basic points that will assist in your understanding its implementation within SkyPilot devices.

RED algorithms do not target frames from a specific node, or a specific device, on the network. RED algorithms discard random frames from the transmission queues when the queue depth (number of frames in the queue) gets to a certain value. For many queues, network operators choose to set this depth at 50% capacity. SkyPilot nodes are no different.

Below the 50% mark, the queue operates and functions much like the freeway on-ramp. Inbound frames are received into the queue and they are stored there until they are scheduled for transmission. New frames are stored at the “back” of the queue and the oldest frames are transmitted first. This is also called First In First Out, or FIFO.



Once the queue exceeds 50% capacity the RED algorithm begins to function. Initially, it will discard a very small number of random frames, as shown in the illustration above where yellow indicates very little activity and red indicates a high level of activity. The end-result of RED is to impact the TCP Slow Start curve for the data stream(s) where discarded frames have occurred. When a random frame in the stream is lost, the TCP slow-start curve (transmission rate between the client and the server) will close by approximately one-half the rate. In many instances, where network congestion is just starting, the reduction of the transmission rates of a few users on the network is enough to allow the network devices to “catch up”. As network resources clear, the TCP Slow Start curve on the impacted machines will automatically readjust to the greater resource availability.

If the RED algorithm fails to adequately control the frame ingress rate to the queues, they will continue to fill beyond the 50% mark. As the queues reach 60%, 65%, 70% and onwards the RED algorithm becomes increasingly aggressive toward its discarding of frames entering the queues. A greater number of data streams are impacted by this discarding and a greater number of TCP Slow Start Curves will close by one-half, as a result. If multiple data frames from a single stream are discarded, it may slow by one-half and then one-half again.

If RED’s more aggressive approach still fails in its attempt to control the ingress frame rate, the queues will eventually fill. At the point that the queue reaches capacity all incoming data frames will be dropped and the queue becomes what is known as a tail-drop queue. All frames that exceed the capacity of a tail-drop queue are dropped.

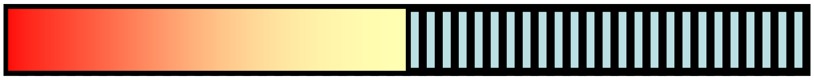
Each SkyPilot node logs and reports statistics representing the total number of frames dropped in the link queues as a result of RED Drops and as a result of the queues reaching capacity, despite the RED Drops, Queue Drops. These statistics are important as we continue our discussion of the **`show link stats all`** command.

## *Link Queue Performance*

Like link transmit and receive statistics, the link queue statistics also provide valuable insight into the performance of a given link, or groups of links. To refresh your memory as to where these are located in the **`show link stats all`** output, we have reprinted it below.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  14   18     24     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  27   24     18     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        0
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191       32
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         0

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  0           6696        588         6           58822        0
00:0a:db:01:0a:4a data  20077       1664435     4438        64          20373152     0
00:0a:db:01:0b:18 data  5           20183       367         5           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ---------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

Recalling to mind the basic queue theory covered in the previous section we can begin to assess link queues using these statistics. In the event that a data queue begins to fill past the 50% mark, **<u>RED Drops</u>** will begin to increment. As the queue fills to a greater capacity the frequency and number of RED Drops will increase.



Using the data, above, if the command **`show link stats all 00:0a:db:01:0a:4a`** were performed multiple times in a row and RED Drops were incrementing at some high rate of change for each query we can assume that data frame transmissions, to the 00:0a:db:01:0a:4a node are occurring at a rate much less than the inbound frame rate for that node.

If the command **`show link stats all 00:0a:db:01:0a:4a`** were run multiple times in a row and the RED Drops only incremented a single time, we could safely assume that the inbound frame rate for the link is not consistently exceeding the frame transmission rate – at least not at a level that would raise concern. An occasional RED Drop will not impact users on the wireless network; continual RED Drops will impact users as they indicate that queues are not only filling, but remaining in that state.

As the queues reach and exceed their capacity the most recent frames destined for the queue will be dropped. When a queue has reached its capacity it takes on the attributes of a tail drop queue and the **<u>Queue Drops</u>** statistic will begin to increment. Queue Drops should alert the operator that the link is severely congested (in the instance where the bulk of the traffic is TCP) or that there is a rise in the amount of UDP traffic on the link. UDP does not reduce its transmission rate as a result of RED Drops due to the fact that it has no transmission control mechanism. In either case, both the legitimate traffic and the unwanted traffic are being impacted. Queue drops should occur rarely, if at all.

In the example, presented above, it can be seen that Queue Drops occurred 32 times on the 00:0a:db:01:0a:4a node. If the **`show link stats all 00:0a:db:01:0a:4a`** were continually refreshed and this value stayed static over time, there would be no cause for worry. However, if a constant refresh of **`show link stats all 00:0a:db:01:0a:4a`** were to show the value incrementing for each refresh it indicates a queue that is not only filled to capacity but still has ingress traffic that continues to exceed that capacity.

# Linking Queue Stats Together

The next step in the evaluation process is to determine whether a particular behavior in this area is related to link or node performance. When the **`show link stats all`** command is executed without a MAC address to narrow the scope of the output, the data for all links will be displayed. This output is what will be needed to determine if the issue is link related, or node related.

## *Identifying Link-Centric Issues*

The output below clearly shows that the queue statistics are worse for the link to 00:0a:db:01:0a:4a than they are for other links. How has this been determined? The first item of interest is the fact that there are 32 Queue Drops on this link, while no other link reports queue drops. RED Drops for this same link are 627 times higher than the next closest link, which is another indicator that the issue presented is link-centric, rather than node-centric.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  14   18     24     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  27   24     18     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        0
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191        32
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         0

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  0           6696        588         6           58822        0
00:0a:db:01:0a:4a data  20077       1664435     4438        64          20373152     0
00:0a:db:01:0b:18 data  5           20183       367         5           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ----------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

When looking for link-centric issues, repeating the **`show link stats all`** command may yield increases on nearly every poll for RED Drops or Queue Drops for just a single link to the node (SkyGateway or SkyExtender). If consistent incremental changes are seen, with each subsequent poll, the issue is link-centric. Links experiencing the most difficulty will have greater increases in either RED Drops and/or Queue Drops. It is normal to see some RED Drops, on a link, from time-to-time. They should not be increasing for every query of a link.

## *Identifying Node-Centric Issues*

The **`show link stats all`** output can also assist in determining node-centric issues when viewing the Queue Drops and RED Drops. When conditions persist to produce the symptoms included in the following discussion, it is usually due to a lack of system capacity. System capacity is a function of available resources in the scheduler (i.e. total time slots) and modulation rates to the first-hop nodes on the network. Lower modulation rates translate into lowered system capacity. Higher modulation rates translate into greater system capacity. SkyConnectors with active links to the SkyGateway, even with the newer releases of firmware, represent system scheduling inefficiencies which will eventually reduce system capacity for those located behind SkyExtenders. It is always advisable to make sure that SkyConnectors have active links to SkyExtenders rather than the SkyGateway whenever and wherever system logistics make this possible. We do understand that it cannot be avoided in all cases.

From the data below it is apparent that the issues are impacting all links on this node. Both Queue Drops and RED Drops are high for each link displayed. This represents a resource issue for the node. Note that RED Drops and Queue Drops correspond to data held in queues used for transmission of data.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  14   18     24     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  27   24     18     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        4271
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191       3257
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         1259

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  50215          6696        588         6           58822        0
00:0a:db:01:0a:4a data  99477          1664435     4438        64          20373152     0
00:0a:db:01:0b:18 data  87546          20183       367         5           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ---------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

What can be extracted from this output? While the RED Drops for the 00:0a:db:01:0a:4a link are higher than the other links, the Queue Drop value is lower than the link above it. This means that the queues did not overflow for as long, or as often as they did for the link 00:0a:db:01:09: cf. This, many times, implies a slightly lower data rate to that link, than to the other links.

Because the RED Drops and the Queue Drops are high for all three links it would behoove us to refresh this data by repeating the command several times more to determine if the values are incrementing, or not. If the RED Drop and Queue Drop values increment for each subsequent refresh then the resource issues that created this condition still exist and are continuing to impact subscribers on the network.

If repeated refreshes fail to yield any further increase in either of these numbers it indicates that there was a resource issue for this node that has, at least for now, subsided and the links have returned to normal.

In either case there is further analysis to be done. By looking at the RED Drop and the Queue Drop values it can be determined that buffer capacities have been reached and, in the case of Queue Drops, exceeded. It can also be determined that the node has run out of capacity to forward data frames to the next hop – whether that next hop is a parent or a child link – at the same rate that the data frames are being received for that link. The conditions that cause this lack of resources can result from several factors. The following sections will show a method by which to converge ALL of these together to determine the root cause.

# Identifying the Queue Impact of RF Issues

Now that the basic concepts of link interference and queue performance have been discussed they can be combined to provide a unified look at the performance of a node, or a link. As with any process, there needs to be a starting point. The starting point that we have chosen may be different from the one you will eventually use. As long as the end-result is efficient and effective identification of the issue(s) at hand, the preferential starting point choice is of no consequence.

## *Start by Viewing RF Performance*

If a network has been through no recent or major changes and user complaints about performance are beginning to escalate, there is a good chance that the performance issues are related to RF interference. Thus, it is common to start from the RF foundation and work progressively through the system statistics to verify the cause of the issue.

There are two RF link performance indicators that are accessed via the **`show link`** command. These indicators are RSSI and TxMod. This RF performance section will begin by discussing the significance of these two statistics.

```
> show link
MAC Address       LType NType State      LRSSI  RRSSI  LTxMod RTxMod LAnt RAnt
----------------- ----- ----- ---------- ------ ------ ------ ------ ---- ----
00:0a:db:01:09:cf data  ext   act mgmt   15     16     18     18     0    0
00:0a:db:01:0a:4a data  ext   act mgmt   27     29     12     6      5    2
00:0a:db:01:0b:18 data  ext   act mgmt   18     17     24     24     0    0
```

**RSSI = Susceptibility to Interference**

A primary indicator of link performance is the RSSI (Receive Signal Strength Indicator) value. A good rule of thumb is that values below 25, for either the LRSSI (Local RSSI) or RRSSI (Remote or far-end RSSI) indicate an increased susceptibility to RF interference on the link. Every attempt should be made to ensure that first-hop links, specifically to the SkyGateway, can achieve a minimum RSSI of 25 for rural installations and 30, or greater, for municipal Wi-Fi installations.

The input signal level to the receiver can be approximated by using the following formula:

**RSSI – 95 ≈ Receive Level in dBm**

As calculated from the formula, an RSSI of 25 roughly equates to an input signal power of -70dBm. If the channel noise floor, characterized by co-channel interference, is -85dBm a 36Mbps modulation rate may be achieved. A 48Mbps modulation rate would require the link RSSI to be maintained at 29. To obtain an RSSI of 29 for a link will necessitate the need for First Fresnel Zone requirements to be met and link distances to be short enough to eliminate any fade that might drop the RSSI below this value.

The table to the right summarizes the C/I ratios and Rx Sensitivity values required to maintain each modulation rate the SkyPilot system supports. The Rx Sensitivity numbers are minimum input signal levels required to achieve a desired modulation rate. This value MUST be upheld in addition to achieving the required C/I ratio.

| Modulation Rate | Required C/I | Rx Sens. |
|---|---|---|
| 6Mbps | 3dB | -92dBm |
| 9Mbps | 4dB | -91dBm |
| 12Mbps | 5dB | -90dBm |
| 18Mbps | 6dB | -89dBm |
| 24Mbps | 10dB | -85dBm |
| 36Mbps | 13dB | -82dBm |
| 48Mbps | 19dB | -76dBm |
| 54Mbps | 22dB | -73dBm |

For example, if a 36Mbps modulation rate were desired, the input signal level could be no lower than -82dBm. The signal strength may be higher than that but it cannot be lower. The -82dBm requirement translates to an RSSI value of 13 (dBm + 95 ≈ RSSI). Additionally, at least a 13dB C/I ratio is required which means that co-channel interference levels can be no greater than -95dBm (-82dBm – 13dB = -95dBm).

Taking it a bit further, if a 54Mbps modulation rate were desired the input signal level could be no lower than -73dBm, or a minimum RSSI of 22. A 22dB C/I ratio must be maintained at all times. This would require the co-channel interference numbers to be at an extremely low -95dBm. This is not achievable in the real-world.

If the channel noise floor were at a more real-world level of -85dBm, as a starting point, achieving a 36Mbps modulation rate will be a bit harder. In this example in order to achieve a 36Mbps modulation rate will require adding the 13dB SNR requirement to -85dBm bringing the input signal level requirement to -72dBm. Add 95 to the -72dBm value to obtain an RSSI value of 23. This assumes that no link fade will occur. In the real world a link can fade up to 5dB based on conditions so a more realistic RSSI value might be 28 in order to maintain a constant 36Mbps modulation rate.

While the math may make link budgets look a bit confusing, what becomes apparent and clear is that the lower the RSSI value the more susceptible the link is to interference in the channel. Lower RSSI values translate into lower C/I ratios which translate into lower modulation rates. The greater the amount of interference in the channel, the higher the RSSI must be in order to maintain the same modulation rate.

### RSSI – Identifying Multipath Issues

As the gap between LRSSI and RRSSI for a link increases it is a strong indicator that the First Fresnel Zone requirements are not being met. While this paper does not have the scope to cover all topics surrounding the First Fresnel Zone requirements, the take-away should be that the First Fresnel Zone is designed to mitigate and control the impact of multipath on the RF link. Nodes placed both too high, or too low, violate First Fresnel requirements and may experience a number of performance issues due to increased amounts of multipath.

```
> show link
MAC Address       LType NType State     LRSSI  RRSSI  LTxMod RTxMod LAnt RAnt
----------------- ----- ----- --------- ------ ------ ------ ------ ---- ----
00:0a:db:01:09:cf data  ext   act mgmt  35     15     18     48     0    0
00:0a:db:01:0a:4a data  ext   act mgmt  27     29     12     6      5    2
00:0a:db:01:0b:18 data  ext   act mgmt  28     27     24     24     0    0
```

In the above example, the link 00:0a:db:01:09:cf displays a variation in RSSI values of 20. While this may not be true in every case, this variation normally indicates the far-end node 00:0a:db:01:09:cf is in an area shrouded either by trees, in a small valley, or surrounded by reflective surfaces (i.e. buildings). It is probably lower in height than the parent node used to query these statistics. While this example is a gross exaggeration of the issue, large variations in RSSI values indicate the presence of destructive multipath on the link. This destructive interference lowers link performance (TxMod) and, in some cases, RSSI and/or modulation rates.

If links variations exceed a difference of 5, or greater, between LRSSI and RRSSI it may be beneficial to adjust the elevation of one, or both, of the nodes on the link to ensure that the First Fresnel requirements are met. This is especially important if there are performance issues with the links in question.

### TxMod Indicates Link Quality

| Modulation Rate | Required C/I | Rx Sens. |
|---|---|---|
| 6Mbps | 3dB | -92dBm |
| 9Mbps | 4dB | -91dBm |
| 12Mbps | 5dB | -90dBm |
| 18Mbps | 6dB | -89dBm |
| 24Mbps | 10dB | -85dBm |
| 36Mbps | 13dB | -82dBm |
| 48Mbps | 19dB | -76dBm |
| 54Mbps | 22dB | -73dBm |

Once RSSI values have been reviewed and any changes have been made, the TxMod data is the next area of interest. TxMod, short for transmission modulation, values represent link quality. A link may have excellent RSSI values and exceptionally low TxMod values. This condition indicates interference in the channel, or co-channel interference.

Because the SkyGateway and SkyExtender have individual, directional, antennas one link may be compromised by interference while all other links on other antennas may not experience any issues. This trend tends to be more apparent in residential broadband than in the more densely populated municipal Wi-Fi meshes.

An approximation of the C/I value for the link can be made using a combination of reported RSSI and modulation rate and comparing it to the modulation rate versus C/I table reprinted here. If the RSSI values for the link are 35 and the modulation rate is 6Mbps, the channel C/I ratio for the link can be calculated at 3dB. This means that the co-channel interference power level is -60dBm. How was this arrived at?

- RSSI – 95 ≈ Signal Level in dBm
- Signal Level in dBm – 3dB = Interference in dBm

In this example, the link RSSI values are high (35) and the modulation rates are low (6Mbps). This indicates the presence of strong co-channel interference values. Co-channel interference can be the result of carriers from other vendor's products or self-interference – as a result of topology within the SkyPilot network. The interference degrades overall link performance and limits the link capacity by lowering the modulation rate. If this were a link to the SkyGateway the impact in performance is not only to the link but to the entire system in the domain of that SkyGateway.

### How TxMod Affects the Scheduler

Without going into a lot of detail, here are some basics of the scheduler in the SkyGateway. The scheduler has 1250 time slots to manage and schedule every second. These timeslots are divided between all first-hop links to the SkyGateway and then further divided by nodes beyond those links. The SkyGateway controls the performance of the entire network.

The 6, 9 and 12Mbps modulation rates are not only slower than their higher value peers, but they also REQUIRE two timeslots per scheduled node, rather than one. If all first-hop links to a SkyGateway consisted of 6, 9 or 12Mbps links the 1250 scheduled timeslots per second would be reduced to 625 timeslots. This further reduces capacity.

As can be seen from this example it is extremely important to ensure that link modulation rates to the SkyGateway are maintained at high values - we recommend 24Mbps minimum and prefer 36Mbps. Lower modulation rates on these critical links reduce system capacity by reducing the amount of subscriber data that can be packed into each radio frame transmission.

The modulation rates, like the RSSI values, can be used to target the location of problems on the network. The node with the lowest receive modulation rate will be the node that has the interferer nearest to it.

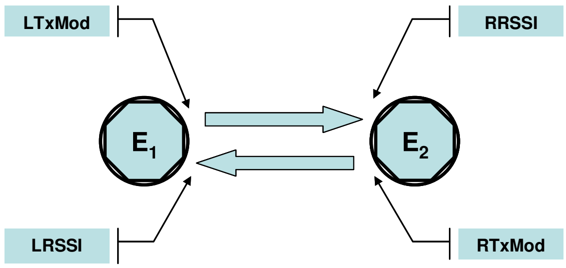
### Some TxMod Examples

The next example shows high values for LRSSI and the RRSSI on the highlighted link indicating that a high modulation rate should be achievable. The modulation rates are very unbalanced with the LTxMod of 18Mbps and the RTxMod of 48Mbps. As per the above explanations about the relationship between RSSI and interference in the channel, it can be assumed that there is something "wrong" with this link. It can also be safely assumed that this is impacting performance of the link. If this were the SkyGateway it would not only be impacting the performance of the link, but of the entire system.

```
> show link
MAC Address       LType NType State     LRSSI  RRSSI  LTxMod RTxMod LAnt RAnt
----------------- ----- ----- --------- ------ ------ ------ ------ ---- ----
00:0a:db:01:09:cf data  ext   act mgmt  29     29     18     48     0    0
00:0a:db:01:0a:4a data  ext   act mgmt  27     29     12     6      5    2
00:0a:db:01:0b:18 data  ext   act mgmt  28     27     24     24     0    0
```

The key to understanding how the LRSSI, RRSSI, LTxMod and RTxMod translate into the network is in the diagram to the right. The values presented assume that the user is logged into the $E_1$ node.

The LTxMod value translates into the Receive Modulation at the far-end. The LTxMod in this example is low and that clearly shows that the interference is local to the $E_2$ node. If the RTxMod were low the interference would be local to the $E_1$ node.



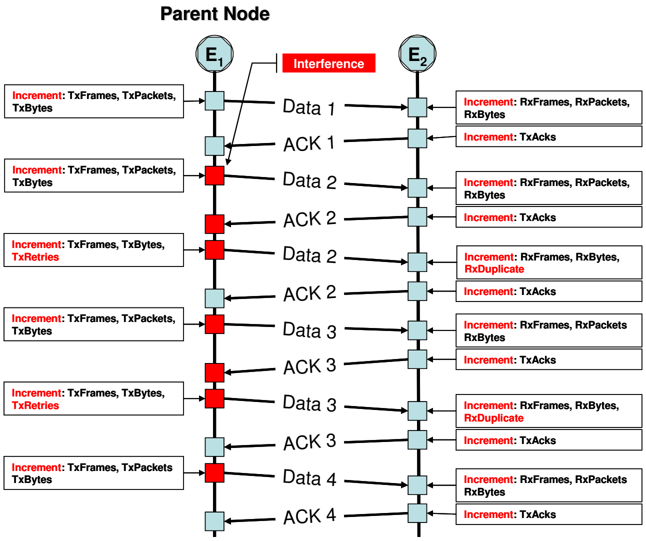
The final take away is that RSSI values ARE NOT impacted by the presence of interference, only the modulation rates. RSSI indicates susceptibility to interference. The TxMod values indicate the quality of the link, or how bad the interference is. Based on whether the LTxMod or the RTxMod value is lowest determines which node the interference is local to. The nodeInfo script will assist in determining if the interference is a caused by network topology. It will also identify the offending nodes. A site survey may be required if little, or no, self-interference is generated by the topology.

# Separating the Scheduler from the RF - Details

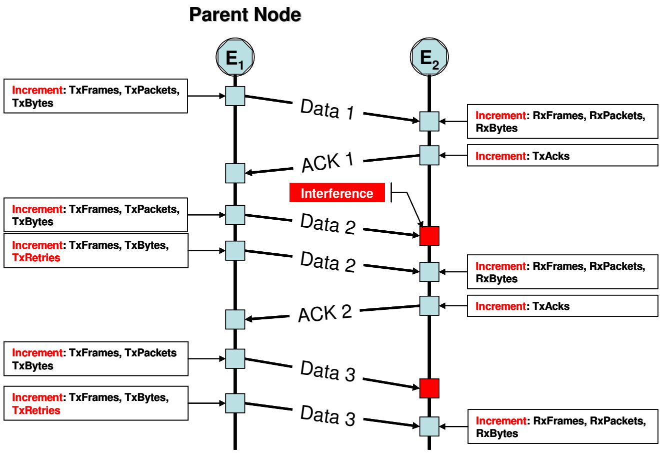
The previous discussions looked at RF performance parameters in-depth; starting from the basics of communications between devices, discussion of queuing theory and methods to a deeper look at RSSI and TxMod relationships. This next section will link the RF issues of low RSSI and/or TxMod values to link throughput, latency and performance.

## *First – A Review of the Basics*

In the introductory sections of this paper two bounce diagrams were presented to demonstrate the impact of RF interference (or low C/I ratio) on transmissions between two nodes. A detailed analysis of the statistics that increment under these conditions was offered. The take away from this section was that when an ACK is not received from the destination node, the transmitting node must attempt to retransmit the radio frame. This may occur up to five times. The time delay between retransmissions is a function of the amount of active links scheduled by the parent node. This process delays all packets in the queues behind the frame held for retransmission.


Parent Node
E1
Interference
E2
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: RxFrames, RxPackets, RxBytes
ACK 2
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxBytes, RxDuplicate
ACK 2
Increment: TxAcks
Increment: TxFrames, TxPackets, TxBytes
Data 3
Increment: RxFrames, RxPackets RxBytes
ACK 3
Increment: TxAcks
Increment: TxFrames, TxBytes, TxRetries
Data 3
Increment: RxFrames, RxBytes, RxDuplicate
ACK 3
Increment: TxAcks
Increment: TxFrames, TxPackets TxBytes
Data 4
Increment: RxFrames, RxPackets RxBytes
ACK 4
Increment: TxAcks

The next section covered queues which are used to hold incoming data frames that cannot be transmitted at the same rate on the link. When the queues reach 50% capacity RED begins to randomly discard frames, recording RED Drops for each frame discarded. The RED algorithm becomes increasingly aggressive with each additional data frame added to the queue. RED serves to actively control TCP stream rates but will not change UDP stream rates.


Parent Node
E1
E2
Increment: TxFrames, TxPackets, TxBytes
Data 1
Increment: RxFrames, RxPackets, RxBytes
ACK 1
Increment: TxAcks
Interference
Increment: TxFrames, TxPackets, TxBytes
Data 2
Increment: TxFrames, TxBytes, TxRetries
Data 2
Increment: RxFrames, RxPackets, RxBytes
ACK 2
Increment: TxAcks
Increment: TxFrames, TxPackets TxBytes
Data 3
Increment: TxFrames, TxBytes, TxRetries
Data 3
Increment: RxFrames, RxPackets, RxBytes

When the queues eventually fill, RED Drops will cease, the queue behaves like a tail drop queue, and Queue Drops begin to increment. Queue Drop counts will continue to increase until the queue drops below 100% capacity.

TxRLimit, RxMissed, RED Drops and Queue Drops are alerts that user data is being dropped on the link. The severity, and impact, of those drops will be determined by the number of customer data frames discarded by these functions. Because data frames are almost always concatenated in transmission from and to the SkyGateway, these drops indicate a far worse hit to network performance than what they would on a link between a SkyConnector and a SkyExtender. The amount of concatenation per radio frame decreases the further into the network the data travels. Bear these points in mind as you read through the final steps of the evaluation process.

## `show link stats all` – The Total View

This section will link RF evaluation and Scheduler evaluation into a unified whole. With any link evaluation process we like to start by looking at the link RSSI and TxMod rates. This can be done using the **`show link`** command. Once you're satisfied the link is performing as best it can given your unique operating conditions, the **`show link stats all`** command can be executed to gather further information.

### *Resource Starvation Causing Data Loss*

The example following shows the performance of three links in terms of RSSI, TxMod, RxDuplicate, RxMissed, Queue Drops, RED Drops, TxRetries, TxRLimit and TxAborts. What is immediately evident in this data is the fact that the RSSI and modulation rates are high and indicate good channel integrity on all links. Confirming this impression is the fact that TxRetries, TxRLimit, TxAborts, RxDuplicate and RxMissed are all very low compared to the total number of radio frames transmitted and received on these links.

The concerns are highlighted in red. Queue Drops and RED Drops are comparatively high numbers. The RF statistics confirm there is little latency resulting from failed transmissions. What do these numbers indicate?

It's clear that the queues are: A) receiving data frames at a rate higher than can be transmitted and B) reaching their capacity frequently. If this were a SkyGateway we could assume that there are far too many active subscribers. If all the traffic at the SkyGateway were necessary traffic the only solution would be to add an additional SkyGateway and split the network. A review of those data streams may reveal unwanted traffic. Those streams could then be filtered or throttled to free up scheduler resources. The changes could then be viewed again, later, here.

If this condition were occurring on a SkyExtender the first inclination would be that this SkyExtender is a bottleneck on the network. A SkyExtender exhibiting these symptoms is aggregating traffic of the ENTIRE network – all links in the mesh funnel through it and it's the only first-hop node with an active link to the SkyGateway. In any instance, stats that look, or are now incrementing, like this indicate scheduler resource issues.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  34   36     36     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  32   36     36     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  33   36     36     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           0           4271
00:0a:db:01:0a:4a data  2607020918  1665748152  0           0           0           3257
00:0a:db:01:0b:18 data  188672452   408529570   0           0           0           1259

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  50215          50         1           0           58822        0
00:0a:db:01:0a:4a data  99477          31         4           0           20373152     0
00:0a:db:01:0b:18 data  87546          34         3           0           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ---------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

## RF Problems Causing Resource Issues and Data Loss

The next example shows how the statistics would play out if the issues were RF related. In many cases RF issues are isolated to a single link as they are in this particular case. The areas of concern are highlighted in red. The link RSSI is low, at 17, indicating increased susceptibility to RF interference. The RTxMod rate is low indicating that interference is present on the link. The LTxMod is also low indicating similar conditions on the far-end. It is possible that an interferer is located part way between the two nodes and is radiating energy toward both of them.

The RxDuplicate, RxMissed statistics are high, indicating the inability of this node to consistently receive and decode radio frames. Because the RxMissed statistic is so high it indicates that when interference is present it's present for long periods of time – long enough to interrupt the transmission of five consecutive radio frames for each RxMissed increment.

The high numbers of TxRetries, TxRLimit and TxAborts also indicate RF interference on the far-end and inability to receive consecutive transmissions – also suggesting that interference, when it occurs, occurs for extended periods of time.

These high numbers of retries are causing the queues to fill while the node is attempting to transmit frames successfully. As the queues fill toward their total capacity RED Drops start to occur, latency increases, customer throughput begins to drop, etc. When the queues reach capacity every inbound frame is dropped, as shown by Queue Drops. This is the period when VPN's will drop, VoIP calls may terminate and other similar events occur with network sessions. For this link, subsequent refreshes of this data by re-entering the show **link stats all** command would yield significant changes in each highlighted area for every burst of interference that occurred.

```
skypilot_Gateway> show link stats all
MAC Address       LType RSSI LTxMod RTxMod TxPow RxFrames    TxFrames    RxPackets   TxPackets
----------------- ----- ---- ------ ------ ----- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  34   36     36     55    253818      315671      68360       48000
00:0a:db:01:0a:4a data  17   18     12     50    44043086    57764332    20294163    24942460
00:0a:db:01:0b:18 data  19   24     24     50    3051537     3259875     989125      1002096

MAC Address       LType RxBytes     TxBytes     RxMalformed RxDuplicate RxMissed    Queue Drops
----------------- ----- ----------- ----------- ----------- ----------- ----------- -----------
00:0a:db:01:09:cf data  15125157    29350596    0           0           9916        0
00:0a:db:01:0a:4a data  2607020918  1665748152  0           1606        28191       3200
00:0a:db:01:0b:18 data  188672452   408529570   0           0           415         0

MAC Address       LType RED Drops   TxRetries   TxRLimit    TxAborts    TxDataFrames TxAcks
----------------- ----- ----------- ----------- ----------- ----------- ------------ -----------
00:0a:db:01:09:cf data  0           6696        588         6           58822        0
00:0a:db:01:0a:4a data  200777      2664435     14438       104         20373152     0
00:0a:db:01:0b:18 data  5           20183       367         5           941392       0

MAC Address       LType Counter Start Time
----------------- ----- ---------------------- --------------------
00:0a:db:01:09:cf data  JAN 01 1970 00:01:51
00:0a:db:01:0a:4a data  JAN 01 1970 00:02:01
00:0a:db:01:0b:18 data  JAN 01 1970 00:00:58
```

## Summary

The **`show link stats all`** command displays a wealth of diagnostic information that can be used to narrow issues down to a specific component within the SkyPilot network. Properly used this output can assist in identifying which links are prone to interference, which nodes are prone to interference and the approximate location of the interferer in the network. From this established base the interferer can be located using site survey tools such as a spectrum analyzer.

The **`show link stats all`** output also allows the system manager to see the impact of RF issues on the transmit data queues and will provide ample warning when users are being impacted through the use of the RED Drops and Queue Drops data.

RSSI and modulation rate values obtained from the **`show link`** command also allow for identification of a links susceptibility to interference (RSSI) and overall link quality (TxMod). The impact of low values in either of these two areas can be validated through the use of **`show link stats all`** and viewing TxRetries, TxRLimit, TxAborts, RxDuplicates and RxMissed. As RF interference and signal strength issues loom larger, these statistics will increase in kind.

On nodes that handle large volumes of data, such as the SkyGateway or the first-hop SkyExtenders, the transmission and receive losses represent an impact to a larger group of users due to the fact that these nodes are most likely to concatenate multiple user data frames into a single radio frame. Because of the traffic volumes on these nodes, issues at the RF level will translate into transmission queues moving toward capacity, the start of RED Drops and potentially Queue Drops – all of which will impact an even larger body of users.

On some networks, RF issues will not play a role in the incrementing of the RED Drops or the Queue Drops. When RED Drops and Queue Drops are incrementing at high rates and few TxRetries, TxRLimit, TxAborts, RxDuplicate or RxMissed are occurring it indicates that the node is running low on bandwidth scheduling resources. It is important to determine if the traffic on the network is wanted or unwanted traffic. If there is unwanted traffic, implementing traffic management policies will free up scheduler and bandwidth resources and reduce the overall impact to the end-user.

Product specifications are subject to change without notice. This material is provided for informational purposes only; SkyPilot assumes no liability related to its use and expressly disclaims any implied warranties of merchantability or fitness for any particular purpose.

Leading the Mesh Revolution

1100 Island Drive
Redwood City, CA  94065
408.764.8000
US Toll Free 866 SKYPILOT
sales@skypilot.com