SkyPilot System Capacity and Performance (v1.4)

Introduction

This document details the performance of the SkyPilot system with the 1.1 release of embedded software. The behavior and performance of various IP protocols transmitted over a SkyPilot system are outlined. The IP “ping” protocol is used to measure system latency, while FTP, UDP and iPerf are used to measure system capacity. In addition to performance data, this paper briefly describes the SkyPilot protocol with particular attention paid to the aspects dictating system performance.

SkyPilot Synchronous Protocol

The SkyPilot system implements a synchronized Time Division Duplex (TTD) protocol. This protocol supports the dynamic allocation of bandwidth based on real-time user demands. This ability to statistically multiplex bandwidth allows an operator the ability to offer high data rate services to many users. The SkyPilot synchronous protocol has the following critical advantages:

- Predictable, Low Latency
- Scalability
- Accurate, Granular QoS Control
- Spectral Efficiency and Reuse

SkyPilot Mini-Slots

All SkyPilot nodes are synchronized to the same 1 second timing frame. The SkyGateway and SkyExtender house a GPS device that provides a highly accurate 1 Pulse Per Second (PPS) clock. SkyConnectors derive the 1 second frame from timing messages exchanged with the connected SkyGateway or SkyExtender.

To allow for the most efficient utilization of the wireless channel capacity, the timing of the upstream transmission is adjusted such that packets are received at the SkyGateway or SkyExtender regardless of the distance from the transmitting node.

The 1 second frame is divided into mini-slots each 100us in duration. These small time increments enable a fine granularity, low latency/low jitter bandwidth allocation.

Bandwidth Allocation

All SkyPilot nodes are synchronized to the same 1 second timing frame. The SkyGateway and SkyExtender house a GPS device that provides a highly accurate 1 Pulse Per Second (PPS) clock. SkyConnectors derive the 1 second frame from timing messages exchanged with the connected SkyGateway or SkyExtender.

To allow for the most efficient utilization of the wireless channel capacity, the timing of the upstream transmission is adjusted such that packets are received at the SkyGateway or SkyExtender regardless of the distance from the transmitting node.

The 1 second frame is divided into mini-slots each 100us in duration. These small time increments enable a fine granularity, low latency/low jitter bandwidth allocation.

Contention Slots

In order to obtain upstream bandwidth, a child node transmits a Bandwidth Request message in the scheduled contention slots. This Bandwidth Request message details the amount of upstream data in each upstream queue.

Currently there is one contention slot every 7.2ms, with every one in eight contention slots given over to Hello message exchange.

Multiple child nodes can transmit in the same contention slot, resulting in a collision. To address this issue, each child node implements a randomized back-off. After each Bandwidth Request transmission, a child node randomizes the number of contention slots opportunities before attempting to issue the next Bandwidth Request. Note that a child node will *not* transmit a Bandwidth Request message if has recently successfully communicated with its parent.

Exchanging Data

A parent node controls the transfer of data to and from its child nodes. A parent node makes the following real-time scheduling decisions:

Transmit Data to Child

Packets are retrieved from the output queues and transmitted to child nodes. Various QoS mechanisms dictate the ordering of packet transmissions to the various child nodes. Typically, the system transmits an Ethernet packet in its entirety; however, if the parent must interrupt the transmission due to traffic shaping mechanisms or to listen to its own parent node, it has the ability to fragment the packet transmission.

Ethernet frames may also be concatenated – more than one Ethernet packet bundled in the same wireless transmission. This concatenation is particularly powerful within a mesh architecture, as it allows packets for multiple endpoints to be efficiently transferred through the intermediary hops.

Having made the decision to transmit to a given child node, a parent starts the transmission at the next scheduled mini-slot with that child. The transmission continues until complete. While this transmission is in progress, the parent node does not communicate in mini-slots scheduled with any other child nodes. As previously noted, transmitting a large Ethernet frame can span many mini-slots.

The time to send a frame from a parent node to a child node is dictated by the modulation rate and packet size, the timing of the next scheduled mini-slot for communication between the nodes and any delays due to packets previously queued for transmission to other this or other child nodes. For a minimum size Ethernet frame, with no queuing delay due to other user traffic, the transfer time will be 2-4ms depending upon modulation rate.

Receive Data from a Child

If a child node has data queued for transmission upstream and is not in active communication with its parent, it will signal the bandwidth need by transmitting a Bandwidth Request message in the next contention slot. The details of the bandwidth request is stored by the parent node and accessed by the scheduler.

In the current SkyPilot system, child data is polled by the parent. Having made the decision to poll data from a given child node, a parent transmits a Poll Request message in the next scheduled mini-slot with that child. Upon being polled, a child node shall immediately transmit a packet up to the maximum size specified in the Poll Request message.

The time to transmit a packet in the upstream is dictated by the packet size, the modulation rate of the link, the time to contend (if required), and the affects of other active nodes concurrently requesting bandwidth. The base latency is defined by the time to send a bandwidth request message, be polled by the parent and transmit the packet. The time to next contention slot is 0-7.2ms depending upon the timing of the packet reception and queuing for transmission upstream. Every one in eight contention opportunities the base latency is stretched to 0-14.4ms due to the Hello message (which takes one of the eight contention slots). Once a parent receives the bandwidth request message, assuming that there are no other packets queued and no contention opportunity to broadcast, then there is 2-3ms time to poll the unit and transfer a minimum size Ethernet packet upstream.

Contention opportunities are broadcast to all nodes connected to the SkyGateway or SkyExtender antenna sector. This means that multiple nodes can generate a bandwidth request message simultaneously, thus causing a collision. After generating a bandwidth request message, all nodes perform a random back-off before generating another message. This back-off is 1-3 contention slot opportunities, and is repeated until the node is polled by its parent SkyGateway or SkyExtender. This means that during times of higher upstream traffic demand the base upstream latency will expand to 0-28.8ms. In very heavy demand this can increase to 0-50.4ms.

The polling of a child continues as long as upstream bandwidth is queued and for an additional integration period of 45ms after the queue is emptied. This continued periodic polling allows for low latency upstream data transfer for an active node by avoiding the need for the child node to contend.

In addition to the dedicated Poll Request message, a parent node can also include a poll request with a data frame transmitted to a child.

Multiple Hops

With the SkyPilot mesh topology, subscriber nodes (SkyConnectors or SkyExtenders) can connect with the SkyGateway directly or over multiple hops. When communicating over multiple hops, the SkyExtenders forward the data from the subscriber node to other SkyExtenders and ultimately forward packets to the SkyGateway.

The SkyExtenders have a single radio, meaning that a SkyExtender can only receive *or* transmit at any given time. This results in an approximate halving of throughput over two hops when compared to the throughput of a node connected directly to a SkyGateway. However, this affect occurs only once, so as the number of hops grows beyond two, the throughput does *not* continuity to be halved. Also note that the system capacity, as represented by the capacity at the SkyGateway, is not affected by the reduction.

Figure 1 below illustrates the throughput obtained in a network of two 3 hop routes.

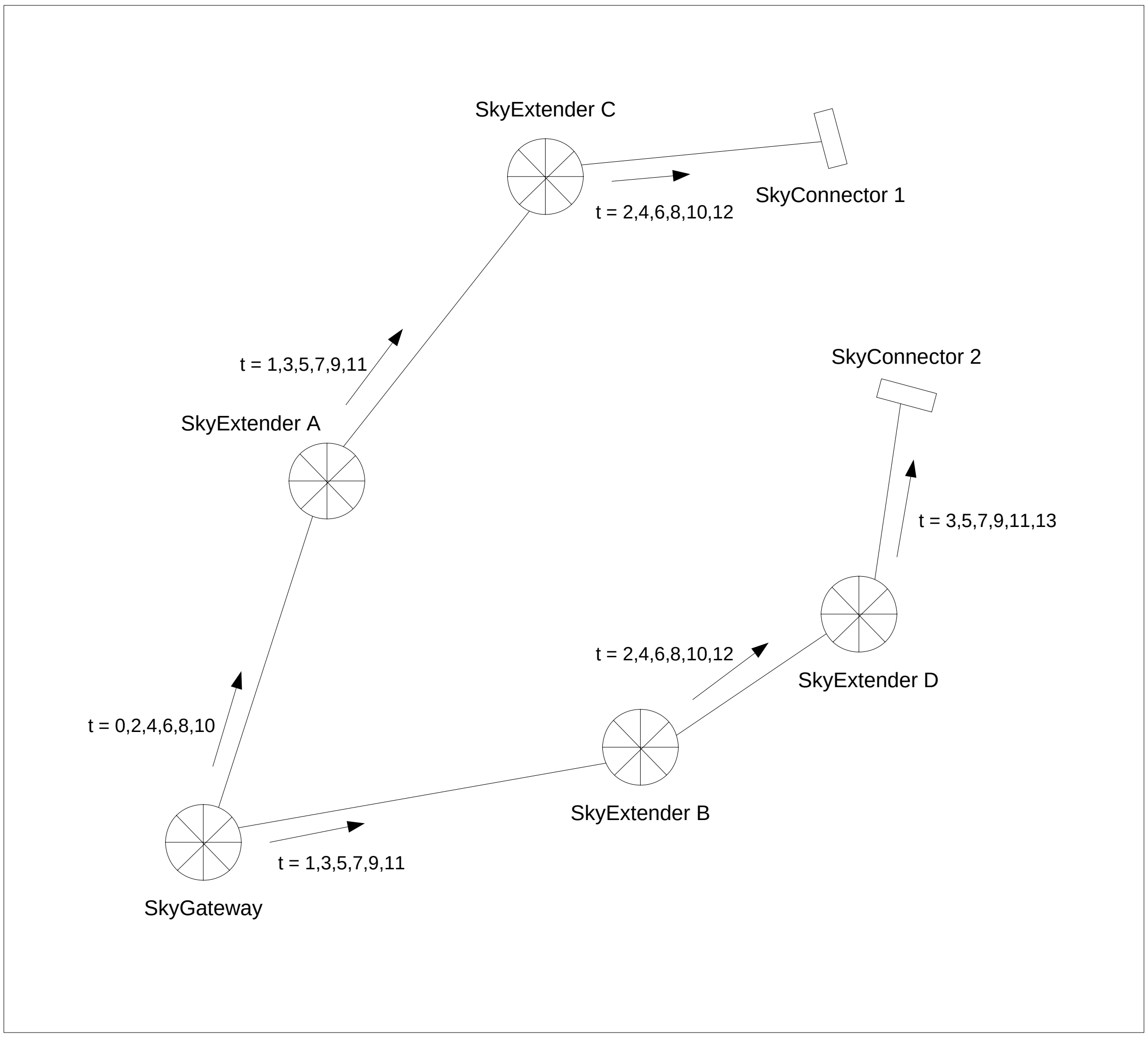


Figure 1 - Concurrent transmissions over multiple hops

In the above example, the timeslots are from t=0 to t=13. The SkyGateway alternates transmit between one of the two directly connected SkyExtenders and transmits on *every* slot. The SkyExtenders along the two routes alternately transmit and receive. For instance, SkyExtender A receives on t=0,2,4,6,8,10 and transmits on t=1,3,5,7,9,11. The fact that the Gateway is transmitting to SkyExtender A at the same time SkyExtender C is transmitting to SkyConnector 1 explains why the bandwidth is halved only once along a given route.

QoS

The SkyPilot protocol efficient supports many subscriber nodes with sophisticated scheduler and QoS mechanisms. The rate shaping and packet prioritization mechanisms manage the queuing of packets in the downstream direction on the SkyGateway and the upstream direction on the individual subscribers SkyExtender or SkyConnector.

The scheduler on the SkyGateway or SkyExtender transmits the downstream packets on a first-in-first-out basis. In the upstream direction, the scheduler polls the nodes based on the demand indicated in the bandwidth request messages. Note that this demand indicates the amount of upstream packet data that is queued (and previously controlled by the upstream rate control and packet prioritization).

When an intermediary SkyExtender receives more downstream data than it can forward (it becomes congested) it will issue a “backpressure” request to its parent node. The backpressure indicator is included in the upstream acknowledgement messages. This mechanism allows packets to be dropped at the source, the SkyGateway, before they enter the wireless network.

A combination of the backpressure mechanism and the rate shaping ensure fairness. In situations of oversubscription, when more demand for bandwidth than is available, the bandwidth granted to each subscriber will be reduced proportionally. For instance, if the aggregate demand is double the available bandwidth, then each subscriber will receive half their configured rate.

Ping Protocol

IP “ping” utilizes the ICMP protocol to discover another IP device. The host computer issues an ICMP Echo Request message directed to the IP address provided. Upon receiving the request message the destination IP endpoint transmits an ICMP Echo Response message. After receiving the response message, the host computer records the time elapsed since the request message was generated. Note that a typical computer operating system (MS Windows for instance) generates one ICMP Echo Request per second.

Figure 2 below shows a typical network topology. In this example network, a ping exchange can be initiated either by the Server attached (indirectly) to the SkyGateway or the computer connected to the SkyConnector.

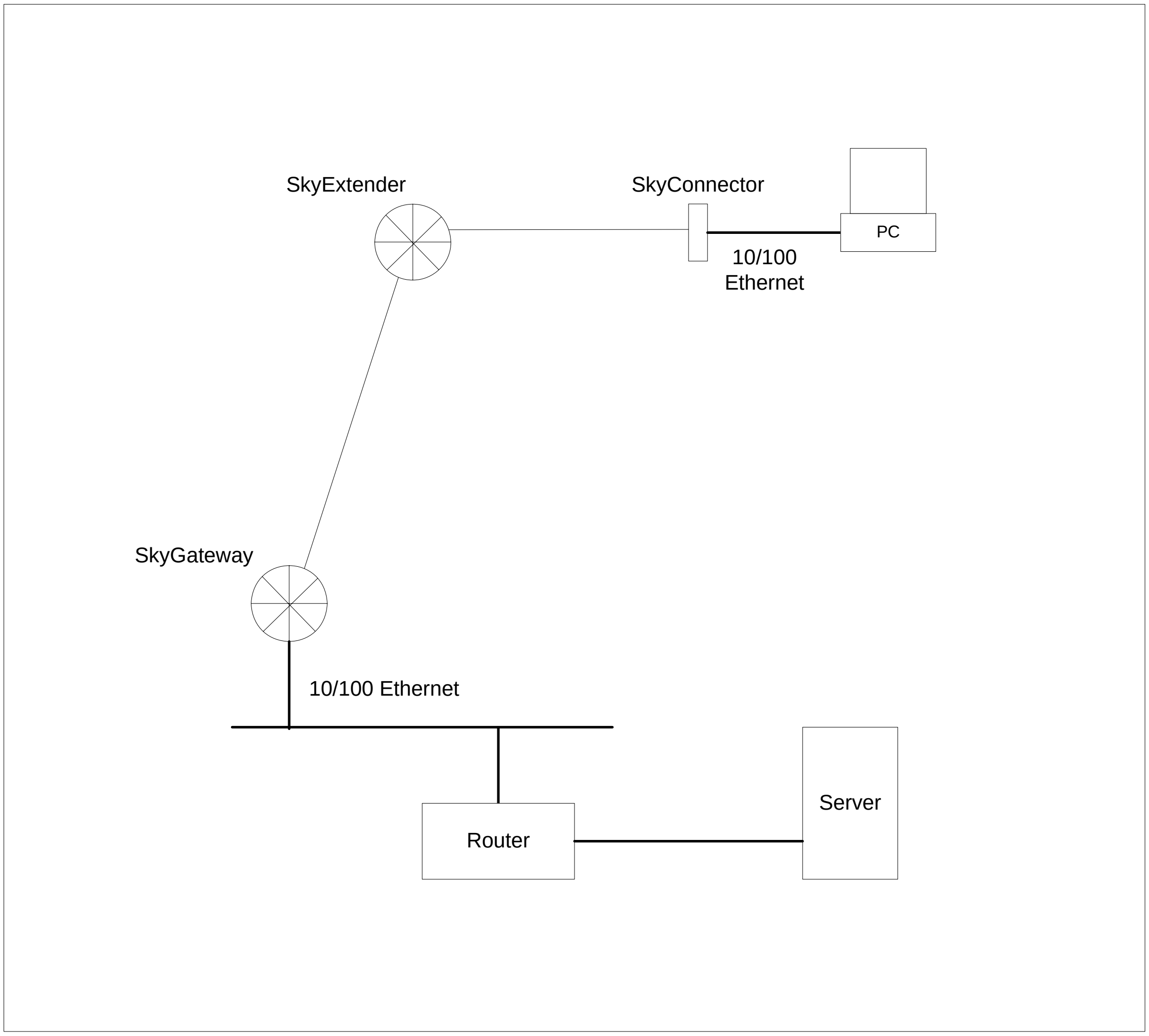


Figure 2 – Network topology for Ping test

Ping Initiated from SkyGateway

I. An ICMP Echo Request message is generated by the Server. It is transmitted via the Router to the SkyGateway. The SkyGateway directs the received Ethernet packet to one of the two queues associated with the destination SkyConnector, based upon the configured prioritization. Following any configured rate control, the SkyGateway transfers the packet downstream to the SkyExtender utilizing the mechanism outlined in section "Transmit Data to Child".

II. Once the SkyExtender receives the packet, it enqueues the packet to the appropriate priority queue for transmission to the SkyConnector. Transmission is on a first-in-first-out basis with high priority packets always transferred first.

III. When the packet is received by the SkyConnector it is immediately transmitted via the Ethernet interface and received by the attached computer.

IV. The computer responds to the received ICMP Echo Request by generating an ICMP Echo Response message which is transmitted via the Ethernet interface.

V. Having received the ICMP Echo Response, the SkyConnector will queue the packet to one of the two upstream queues following any configured rate control. The SkyConnector then follows the mechanism outlined in section "Receive Data from a Child" to transfer the packet to the SkyExtender.

VI. Once the SkyExtender receives the packet, it enqueues the packet to the appropriate priority queue for transmission to the SkyGateway. Transmission is on a first-in-first-out basis with high priority packets always transferred first. Again, the SkyExtender follows the mechanism outlined in section "Receive Data from a Child" to transfer the packet to the SkyGateway.

VII. Upon receiving the ICMP Echo Response, the SkyGateway immediately transmits the packet via the Ethernet interface. The Router then forwards the packet to the Server, where the latency is recorded.

Ping Initiated from SkyConnector

When the computer initiates the ping, the steps are outlined in the previous section are reordered. The SkyConnector receives and forwards the ICMP Echo Request message (Step 5 above) via the SkyExtender (Step 6) to the SkyGateway (Step 7). Upon receiving the message, the Server generates an ICMP Echo Response message that is forwarded to the SkyGateway (Step 1). This response message is forwarded via the SkyExtender (Step 2), to the SkyConnector (Step 3), and onto the computer, were the latency is recorded.

Factors in Ping Latency

There are a number of factors that govern the ping latency recorded across a SkyPilot system:

1. **Number of Hops**
   Latency is proportionate to the number of hops in an individual route – the number of hops between an end user and the SkyGateway. Note that higher levels of traffic aggregated over particular hops (typically closer to the SkyGateway) will increase latency.

2. **Upstream Contention**
   Because ICMP Echo Request messages are generated once a second, upstream transmissions will always contend for bandwidth. The only exception to this rule is if the same subscriber is concurrently generating higher volume traffic on the hop.

3. **Packet Size and Modulation Rate**
   The default packet size for ICMP packets is just 64 bytes. With that size of packet, the transmission times on the wireless and Ethernet networks is very small, and the packet is easily fit into a single SkyPilot mini-slot regardless of modulation rate. However, many ping programs allow the size of the ICMP packet to be increased up to 1514 bytes, the maximum Ethernet packet size. These larger packets will significantly increase the transmission size and, depending upon modulation rate, may be fragmented during transmission across the SkyPilot network.

4. **Quality of Service**
   The very low number of packets generated by a ping session means that any configured rate control is unlikely to cause additional latency. However, rate control combined with concurrent traffic could significantly add to latency.

5. **Packet Prioritization**
   The prioritization of ICMP packets will preserve the ping latency in the face of (unprioritized) traffic. Conversely, if ICMP packets are *not* prioritized but other packet types *are*, then latency will be significantly increased.

6. **Multiple Users and Traffic Volume**
   If there is significant traffic from multiple users, then latency will increase. This increase is caused by queuing delay through the Gateway in the downstream or intermediary SkyExtenders in either direction. Increase in latency is also caused by multiple nodes simultaneously contending for upstream bandwidth, which results in collisions and the following back-off.

7. **Interference**
   ICMP packets can be destroyed by external interference, which requires the need for retransmission.

Note that there may be other factors outside the SkyPilot network that contribute to ping latency. This includes congested Ethernet networks and routers, and overload computers or servers. It is recommended that whenever a latency test is undertaken, that the latency of the non-SkyPilot elements be tested – connected the computer directly to the same Ethernet segment to which the SkyGateway is connected and then run a latency test to the server (through the Router).

Ping Latency Test Results

Below are the results for a one per second ping test over one, two and three hops. This testing was done in a quiescent state with only management and protocol messages present and *no* other user traffic. Each of the hops has a modulation rate of 36Mbps in both directions. As can be seen, the latency increases linearly with the number of hops with an average of approximately 9ms of round trip latency per hop. The variance in ping times is explained by timing of the upstream ICMP message with respect to the next contention window, which results in an approximate 7ms spread in the upstream. In the downstream the variance is approximately 2ms resulting in an approximate 9ms spread for the round time latency per hop. See the “Exchanging Data” section for more details of the latency factors in the SkyPilot system.

Number of Hops	Packets Transmitted	Min Latency	Average Latency	Max Latency
1	120	3.647	7.985	17.512
2	120	12.647	24.303	47.507
3	120	20.217	27.345	61.860

FTP Protocol

The File Transfer Protocol (FTP) is often used as a benchmark for network system performance. FTP is used to transfer large files over an IP network. Files can be transferred from one computer to another via a push ("put") or pull ("get") mechanism.

FTP breaks the file into blocks of data that are forwarded to the receiver. FTP is an acknowledged protocol, meaning that for each of these blocks of data the sender must receive an acknowledgement message from the receiver. A "sliding window" is implemented such that a number of blocks can be forwarded before the sender waits for acknowledgement. The FTP results presented in this document were collected utilizing a block size of 2048 bytes and a sliding window of 4096 bytes (2 blocks).

One other aspect of FTP that should be noted is the "slow start" algorithm. This is the means by which two ends of the FTP session coordinate the speed of the file transfer based upon the available capacity. For a full description of the FTP protocol consult one of the many references (TCP/IP Illustrated Volume 1, W. Richard Stevens is highly recommended).

Note that the throughput number provided by the FTP application is based only on the volume of file data exchanged. The TCP and Ethernet headers are not included nor are the acknowledgement packets.

Factors in FTP Performance

There are many factors that dictate the throughput reported by an FTP transfer. These relate to both the computing elements as well as the link layer protocol. Listed here are the key factors, particularly those related to the SkyPilot protocol:

1. **Computer Processor Performance**
   FTP data rates are directly affected by the processor and disk performance of the Server and Client computers. Before performing FTP tests across a SkyPilot network it is recommended that the base performance of the computers is tested by exchanging files between the Server and a computer on the same Ethernet segment as the SkyGateway.

2. **Acknowledgement Packets**
   The default MS Windows FTP settings are a block size of 2048 bytes and sliding window of 4096 bytes. These settings will result in one FTP acknowledgement message for every block transmitted. This represents a significant amount of bandwidth that is *not* recorded as a part of the FTP data rate - the amount of bandwidth utilized for acknowledgement packets is an addition 50% of the bandwidth used to transfer file data and only the file data is reported in the throughput result.

3. **Number of Hops**
   The number of hops in a route is a key factor. The section "Multiple Hops" provides details of the SkyPilot system performance over multiple hops. For FTP, the bandwidth is halved as a route increases from one to two hops. However, there is no further decrease as the number of hops increases beyond two. The only affect for FTP beyond two hops is the increase latency (which is proportionate to the number of hops). The increased latency can reduce FTP throughput by delaying the reception of acknowledgement message. This can be more significant when the FTP sliding window is small in size, as data transmission can stall await acknowledgement for pervious block transmissions. Increasing the sliding window can help to prevent throughput decreases particularly when the number of hops is greater than 3 or 4.

4. **Modulation Rates**
   The modulation rate of the links within the network under test can have a significant affect on FTP performance. A series of FTP test results obtained with differing modulation rates are shown on the next page. Note that the modulation rate in the direction of the data packet flow dictates most significantly the FTP throughput. Because the acknowledgement message is only 64 bytes in length, the modulation rate in that direction has less affect.

5. **Upstream Contention**
   FTP performance is particularly sensitive to the latency of the acknowledgement messages. If the acknowledgement packets are transmitted in the upstream direction (i.e. a file is being transferred downstream to a computer connected to a SkyExtender or SkyConnector), then contention collisions can slow the transfer and decrease throughput.

6. **Quality of Service**
   The FTP throughput rate will be constrained by any configured rate limits. If a test of maximum throughput is required, it is recommended that rate limiting be disabled for the node under test.

7. **Packet Prioritization**

FTP performance can be reduced if prioritized traffic is present within the network. The affects are most notable when the prioritized traffic is in the upstream direction and this is the *same* direction as the acknowledgement packets. This can cause acknowledgement messages to be delayed and force the need to contend for upstream bandwidth.

FTP Test Results

Below are the Downstream FTP results of a 10 MB file, obtained with various modulation rates over one to four hops:

Number of Hops (All links 36/36 prior to last hop)	36/36 Downstream	36/36 Upstream	6/6 Downstream	6/6 Upstream
1	10.662 Mbps	8.199 Mbps	2.476 Mbps	2.44 Mbps
2	5.212 Mbps	4.397 Mbps	1.752 Mbps	1.709 Mbps
3	3.983 Mbps	3.770 Mbps	1.567 Mbps	1.272 Mbps
4	3.419 Mbps	3.286 Mbps	1.533 Mbps	1.334 Mbps

iPerf Protocol

iPerf is a freeware networking performance tool that provides both TCP and UDP test capabilities. iPerf is available for Linux and Windows operating systems, and can easily be located and downloaded via the Internet.

iPerf , just like FTP, is a computer based tool and is therefore sensitive to issues of processor and disk performance. When testing a SkyPilot system using iPerf, it is recommended that a baseline test be performed that excludes the SkyPilot network – include just the server and client computer, and any Router or Ethernet switch used as a part of the overall network.

The iPerf TCP feature utilizes an acknowledged protocol very similar to the FTP protocol. When used to test the SkyPilot system, the factors effecting performance and the actual throughput rates observed, should be very close to those obtained in an FTP test of the same network.

For iPerf UDP with respect to the one hop tests, throughput efficiency compared to the modulation rate is fairly consistent. With large Ethernet packet sizes, the highest efficiency is 66.1% for the 18Mbps modulation rate and lowest for 9Mbps modulation rate with a peak efficiency of 55.3%. The differences in efficiency are dictated by the size of the wireless transmission slots compared to Ethernet frame sizes. The higher efficiency indicates that there are less unused bytes within the wireless packets.

For each of the modulation rates the throughput is generally consistent across the Ethernet packet sizes. The exception is when the packet sizes reach the lowest size and the throughput becomes limited by the number of packets that can be processed. For 64byte packets the throughput is limited to 3.5Mbps, for 124byte packets the limit is approximately 7Mbps, and for 256byte packets the limit is 13.65Mbps. The 256byte packet processing limit is an issue only for the 36Mbps modulation rate. Note that the average packet size over the internet is usually estimated to be approximately 512bytes.

iPerf Best Case Test Results

This test represents the best case scenario in which devices are tested with an absence of interference or range propagation delay.

Multi hop results reflect the modulation of the last hop. All prior hops are 48/48 modulation rates. Results are in Mbits/sec.

TCP

# of Hops	48/48		36/36		24/24		18/18		12/12		9/9		6/6	
	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up
1	10.5	10.8	10.8	10.7	8.2	8.2	6.5	6.7	4.3	4.5	3.5	3.6	2.5	2.6
2	5.9	5.0	5.4	4.8	4.3	4.2	3.5	3.6	3.0	3.0	2.5	2.5	1.9	1.9
3	4.8	3.9	4.5	3.9	3.7	3.5	3.1	3.1	2.7	2.5	2.3	2.1	1.7	1.5
4	4.0	3.5	4.0	3.5	3.6	3.1	3.0	2.6	2.6	2.4	2.1	2.0	1.7	1.5

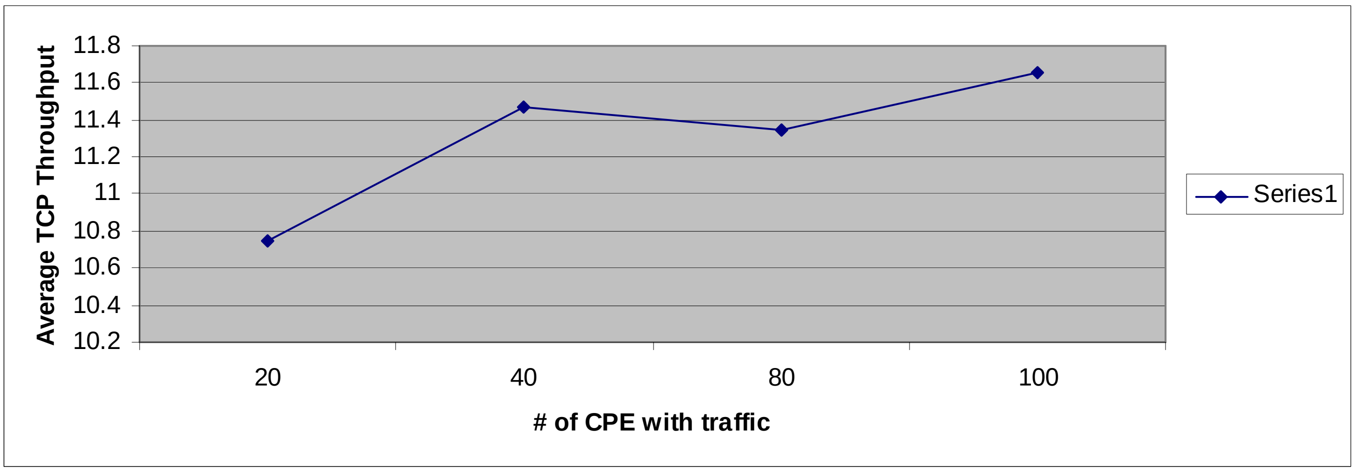
UDP

# of Hops	48/48		36/36		24/24		18/18		12/12		9/9		6/6	
	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up	Down	Up
1	18.2	18	18.5	18.3	13.0	12.3	9.5	9.6	7.2	7.3	5.3	5.3	3.4	3.4
2	8.3	7.3	6.2	6.6	4.1	5.3	3.0	4.4	3.3	4.0	2.4	3.2	1.5	2.4
3	8.0	6.6	6.3	6.2	4.1	4.8	3.0	4.0	3.1	3.3	2.3	2.7	1.5	1.8
4	6.8	6.1	5.3	5.4	3.5	4.2	2.5	3.4	1.9	3.2	1.4	2.6	866 Kbps	1.8

iPerf Point to Multi-Point Scaling Test Results

This test is conducted with iPerf TCP to a varying number of SkyConnectors linked to a single SkyGateway parent.

Total SkyConnectors	SkyConnectors receiving data	Average Aggregate Throughput
23	20	10.75 Mbps
44	40	11.47 Mbps
81	80	11.35 Mbps
103	100	11.66 Mbps



This document is covered by the terms of your existing non-disclosure agreement with SkyPilot and should not be shared outside of these parties.

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

9/05