Satellite-based IFC Performance Degraded by "Performance-Enhancing" Protocols

Peter Sevcik and Rebecca Wetzel
August 2019

In real-world tests of satellite internet performance, NetForecast identified key factors that adversely affect the user experience.

- Performance-Enhancing Proxy (PEP) [1] protocols commonly used in satellite networks have degraded performance when requested content is far from the first PEP proxy.
- Furthermore, the degradation causes excessive packet retransmissions which exacerbates load congesting the satellite link.
- Eventually, the above conditions can spiral to where the PEP proxies are forced to reset many TCP connections with significant negative impact to applications and users
- The above factors can be managed with careful deployment architecture and ongoing end-to-end QoE measurements.

There are four general Satellite Communications conditions that apply:

A – The ground proxy is close to the data source and the link is not congested.
B – The ground proxy is close to the data source and the link is congested.
C – The ground proxy is far from the data source and the link is not congested.
D – The ground proxy is far from the data source and the link is congested.

As seen below, NetForecast's tests under these four conditions show significant performance differences that directly influence user quality of experience (QoE).

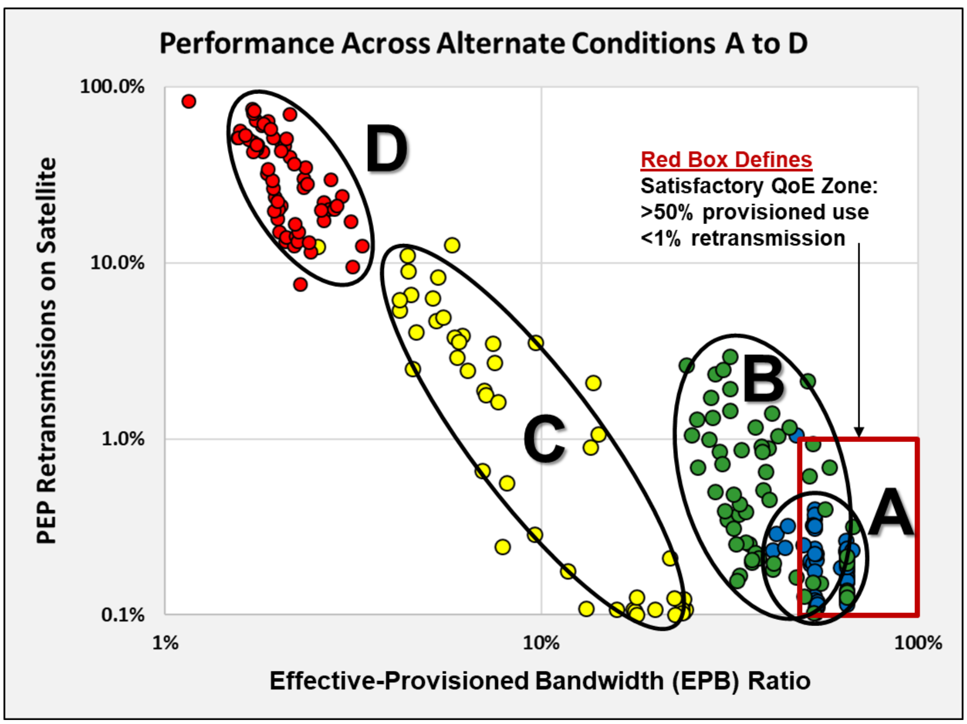

Figure 1 – Satellite-connected User Experience Across Four Conditions

NetForecast Report
NFR5132

Airline Internet Connectivity Implications

Satellite-based in-flight connectivity (IFC) services that use PEP protocols nearly always operate in condition C and D zones. This is particularly true for international flights, since passengers are highly likely to access content near "home," but the satellite ground station selected by the operator is likely far from that home location.

Note that the axes in Figure 1 are log scales. The vertical scale showing extra traffic clogging the satellite channel due to PEP retransmissions is 1000-to-1 (top-to-bottom). The horizontal scale showing the bandwidth effective-provisioned bandwidth (EPB) ratio is 100-to-1. Effective bandwidth is the nominal data rate delivered over a Transmission Control Protocol (TCP) connection. These scales are required to show the massive performance range actually measured on a satellite service. The small red box in the lower-right corner of the chart is the performance zone in which the user quality of experience (QoE) is generally satisfactory for browser-based web interactions. Only condition A can provide satisfactory QoE for the majority of internet use from an airplane.

How PEP Protocols Work

Originally designed to support remote retail locations or field offices accessing a corporate datacenter via satellite using VSAT (very small aperture terminal) services in a 'hub and spoke' configuration, PEP protocols are designed to overcome TCP performance limitations over high latency networks. A geosynchronous satellite link operates with extremely high latency.

The Bandwidth-Delay Product (BDP) Limitation

Very long network latency in satellite networks (the sum of source network, satellite, and destination network delays) creates a poor effective bandwidth system due to what is called the 'bandwidth-delay product' phenomenon. The essential issue is that the TCP window mechanism cannot fill the link because the quantity of data sent is insufficient compared with the bandwidth-delay product. The product of a data link's capacity (in bits per second) and its round-trip delay time (in seconds) determines the maximum amount of data transiting (in-flight) on the network path at any given time—i.e., transmitted data that has not yet been acknowledged.

> Bandwidth-Delay Product Math
>
> Delay for a geosynchronous satellite link is approximately 270 ms to the satellite and back. A TCP packet and returning acknowledgement (ACK) consume that twice in what is known as round-trip time (RTT), which is 540 ms or 0.54 seconds. Assume a service provides 10 Mbps to a TCP connection. The product determines the number of bits that must be transmitted (put into flight) before an ACK will arrive from the destination:
>
> BDP = 10,000,000 bps *X* 0.54 seconds = 5,400,000 bits
>
> A typical default 16-bit TCP window can authorize a 64K byte receive window permitting the source to transmit a maximum of 524,288 bits. Therefore, after the 524,288 bits are sent, the source will stop and wait for an ACK. The transmitter will use only 524,288 bits out of the 5,400,000 bits available within an RTT timespan. That means 9.7% of the time transmission occurs and the rest of the time is "dead air." The effective throughput of the path is 0.97 Mbps. Since the provisioned capacity of the channel is 10 Mbps, the effective-to-provisioned bandwidth ratio is 9.7%.
>
> ***Adding bandwidth does not help. In fact, a larger "pipe" increases the denominator thus forcing the ratio lower.***

TCP Window Scaling addresses this limitation permitting windows that are as large as 1GB as described in RFC7323, TCP Extensions for High Performance. [1]

There are many additional advanced mechanisms in TCP that can overcome BDP. However, most PCs and devices do not support them in their native operating systems. Furthermore, most of the enhancements require server software that consumes copious server resources to support many users. The greatest server impact is that most solutions require operating larger buffers for every TCP connection. Large buffers are costly and can have computing performance degradation impacts. For example, the most practical solution is TCP multithreading which all browsers support. The same small 1 Mbps effective bandwidth in the above example is applied to 4 or even 8 simultaneous TCP connections operating in parallel. That is an 8-fold increase in effective bandwidth with no change to TCP on either end. However, multi-threading consumes more resources and can have practical limitations depending on the type of content being transferred.

PEP to the Rescue

A Performance-Enhancing Proxy (PEP) is a type of proprietary protocol that modifies the way TCP functions to improve the user experience by overcoming latency-related issues endemic to satellite links. PEP protocols operate two proxies, one on each end of a satellite link. As figure 2 shows, PEP terminates the source-to-destination TCP connection at the first proxy. The packets are intercepted by the first proxy, which manages transmissions through the satellite link to the second proxy, which then sends the requested data to the user. The process works in reverse for up traffic.

Some PEP approaches—known as transparent proxy PEP—do not use a second proxy. However, in either approach there is a proxy between the data source and the satellite circuit. The path from the source to the common first proxy is critical.

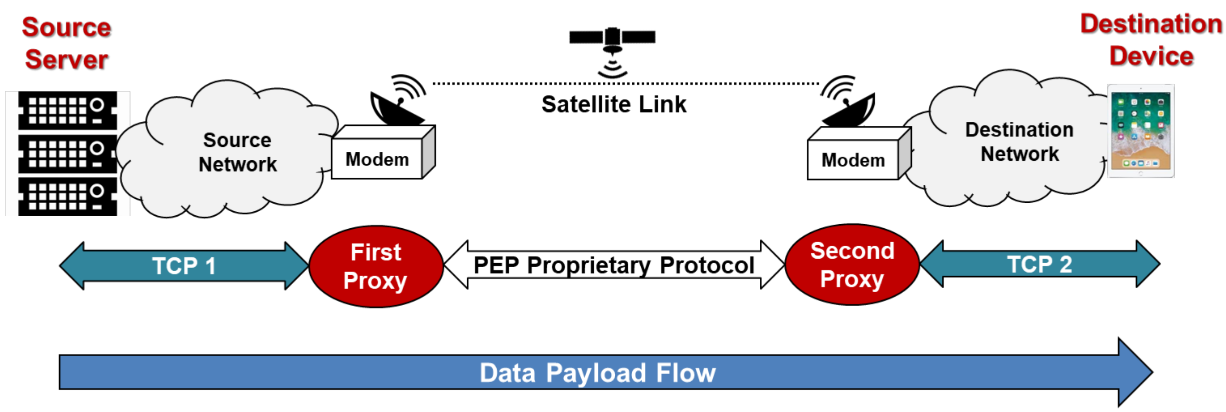

Figure 2 – General PEP Protocol Architecture

TCP is designed to guarantee data is delivered in the same order in which it is sent. It does this by returning an acknowledgement (ACK) when data packets (aka payload) are delivered, so the system knows it can safely send subsequent packets. TCP is a "stop-and-wait" protocol that needs a rapid end-to-end response to function well and maximize provisioned bandwidth utilization. To speed response the first PEP proxy does not wait for an ACK from the destination and instead returns a premature ACK to the source as soon as it receives the initial payload. The premature ACK includes a window update signaling the source to send more packets while the initial packets are still in flight over the satellite link. Packets that arrive but cannot be sent immediately are held in a PEP buffer at the first proxy.

This behavior improves the satellite network user experience when the source-to-near-proxy distance is short, but it hurts the experience if this distance is long, as described below.

How PEP Protocols Can Hurt the User Experience

Premature ACKs place a burden on the first proxy. A premature ACK falsely tells the source that the data was successfully delivered to the destination, and in response the source clears its transmit buffer. In so doing, the first proxy promises to deliver the data—but what if it cannot? Four major elements must handle the packets to the destination: satellite link, second proxy, destination network, destination device, and each of them must perform flawlessly.

The first proxy works to keep the source operating at a throughput pace greater than that at which TCP would normally operate given the high system latency. It counts on the proprietary protocol to continuously operate at the same high throughput. After all, if it does not run better than TCP, then why use PEP?

The first proxy maintains state information about packets received, packets sent to the destination, and packets prematurely ACKed and held in its buffer. The proxy uses a sophisticated set of parameters to store and forward packets to the destination proxy.[2]

It all works well if the source network has low latency and little loss, thus enabling rapid signaling between the first proxy and the data source. However, when the source-to-first-proxy path experiences high latency or loss, the following issues emerge:

> **Issue 1:** High latency diminishes the ability of the first proxy to manage the high flow rate due to a long feedback loop. For example, the first proxy may tell the source to reduce the rate by half, but that will apply to packets sent after the source receives the rate reduction signal. In the meantime, it has already put many packets into flight towards the first proxy that the proxy was not expecting. Remember that the proxy mechanism parameters are typically set assuming the source is nearby and therefore responsive.
>
> **Issue 2:** Another consequence of high latency on the source network is that packets are more likely to arrive out of sequence. The first proxy is expecting packets 1 through 8 in that order. Should they not arrive in order—e.g., 1, 2, 4, 5, 6, 7, 3, 8—then premature ACKs will be sent for packets 1 and 2, but packets 4 to 7 will not be prematurely ACKed because packet 3 is assumed to have been lost on the source network.
>
> When the proxy fails to see packet 3, it will send a TCP DUP ACK to the source for packet 2. Repeated ACKs at the last known value before the gap signals which packets the sender should retransmit without waiting for the acknowledgement timeout for the lost packet. This triggers a TCP fast retransmit which, as the name implies, means recovering a lot faster.
>
> The overall impact of lost and out of sequence packets causes the flow to stop until proper sequence is restored. This creates a big transmission gap (dead time). Once sequencing and smooth flow is restored, the proxy will ramp up the data acceptance rate again, but throughput is likely to be highly variable over time.
>
> **Issue 3:** The first proxy is always holding some prematurely ACKed packets that must be successfully transmitted over the satellite link. When the satellite link is congested, it drops packets. However, the prematurely ACKed packets must get through, so the PEP proxy retransmits them—perhaps even multiple times during peak congestion. These retransmissions further clog the satellite link, adversely affecting other user flows.
>
> **Issue 4:** If any of the first three issues occur repetitively or simultaneously, the first proxy buffer can max out and no longer hold prematurely ACKed packets. In these cases, the first proxy terminates the TCP connection with a full buffer. This is more likely when flows are large because they take longer and are more likely to be caught in the system when conditions deteriorate enough to kill connections. They are also prime candidates for removal since they likely have the greatest

number of packets in the proxy buffer.[1] The proxy will limit the number of TCP connections to be reset to those consuming the greatest amount of buffer space. TCP resets cause serious repercussions to the application and user experience. They often cause the user to restart the transfer process.

PEP issues 1 and 2 reduce effective system bandwidth, while issues 3 and 4 cause retransmissions on the satellite link which exacerbates link congestion.

NetForecast Test Methodology

NetForecast performed active reference tests in which a NetForecast test client (probe) and server generated the only traffic on dedicated test lines in satellite service users' residences. Although the users in the study were not airborne during the tests, the test results apply to airline passengers as the network behavior observed is common to all PEP-enabled satellite services. Figure 3 shows the server-to-client test configuration (aka down test). Tests were also performed in the opposite direction client-to-server, (aka up test) as shown in Figure 4. The distinction is important in this study since reversing the TCP payload flow reverses which proxy is operating as the first proxy.

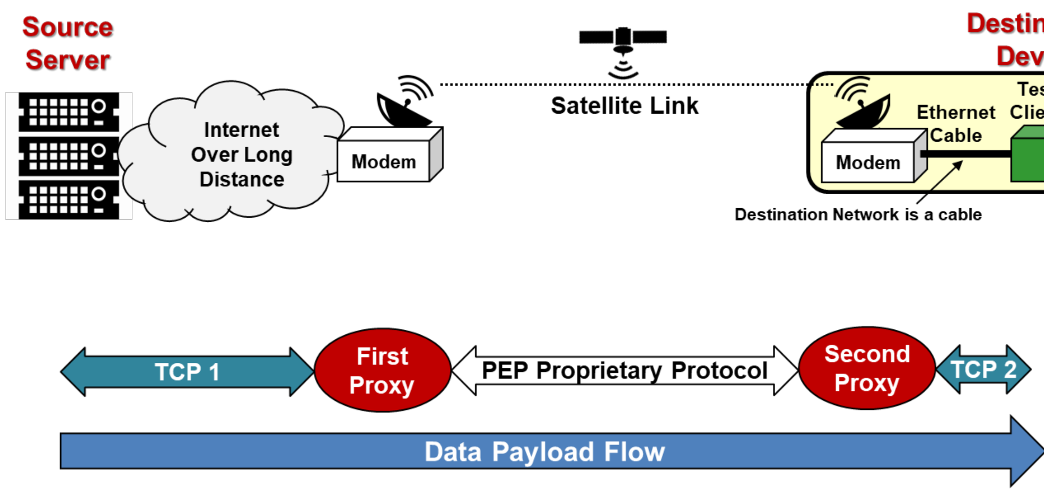

Figure 3 – NetForecast Server-to-Client Test Configuration

NetForecast transferred 20 and 50 MB files from a NetForecast server to the subscribers along with 2 and 5 MB files from subscribers to the NetForecast servers. We performed independent measurements in the users' homes.

All NetForecast measurements were performed using the NetForecast UMap[SM] service delivery platform. The UMap system is enabled by measurement and reporting software embedded into customized, fully featured, wireless home routers supplied and supported by NetForecast.

NetForecast's tests were lightweight so they would not stress the circuit. The tests consisted of sending small files in sequence from the NetForecast server to the home, with test sizes adding up to 50MB or

[1] Tests that terminated with a TCP reset were removed from this analysis and report.

20MB. To keep our tests lightweight and to ensure accurate results, NetForecast did not use multithreading of TCP or traffic flows (such as browsers do).[2]

To put the test volumes into better perspective they represented the following typical use case scenarios:

Down 50 MB - a high-resolution digital photo in RAW format
Down 20 MB - the same photo in high quality JPEG format
Down 20 to 50 MB - the first loading of a web page
Up 5 MB and 2 MB - one-tenth of the down file sizes

NetForecast measured the elapsed test time, the number of down and up bytes, and the number of down and up packets. NetForecast's measurements included every byte that passed between the home and the satellite access circuit. The satellite service provided NetForecast with the subscriber's hourly usage over the satellite link in both directions. We were thus able to determine the difference between traffic that was handled by the satellite link and what was delivered to/from the subscriber.

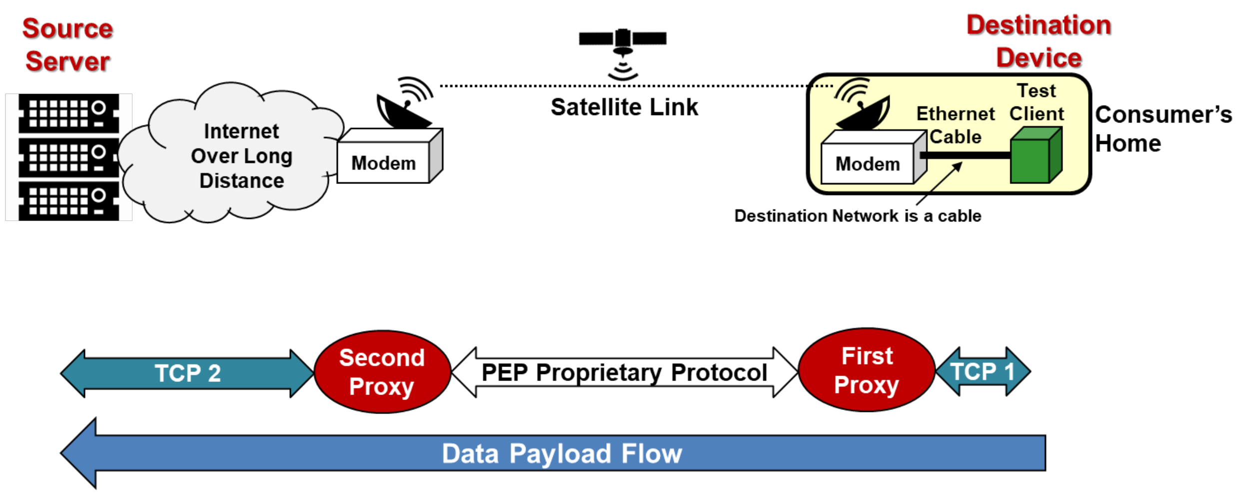

Figure 4 – NetForecast Client-to-Server Test Configuration

Down and up tests occurred on alternate hours in local time. Down tests occurred during odd hours and up tests during even hours, with no tests in hours 12 and 14.

The down test involved requests for files in a source server connected to an internet segment that was a long physical distance from the sending satellite ground station, and the up test involved a source that was connected via a dedicated Ethernet cable to a VSAT terminal—as close as physically possible to the transmitting ground station. This configuration enabled NetForecast to measure performance with content located close to and very distant from the transmitting ground stations where the first proxy is located.

Testing operated over a standard public VSAT service operating as the last mile to the internet. The satellite network experienced distinct low and high contention periods due to consumer use patterns, with highest usage during 21:00 and 22:00 hours local time, and lowest usage during 2:00 and 3:00 hours local time. The high usage hours often congested the satellite link while low usage hours showed very little congestion. The test results used in this report are based on the specific hours across three months of testing as shown in table 1.

[2] NetForecast did not use multi-threading because it could have placed an undue burden on the access circuit, and it would have interfered with the ability to calculate the effective bandwidth per test.

	Test Payload Direction	
Satellite Contention	Up (C-to-S)	Down (S-to-C)
Low	2:00	3:00
High	22:00	21:00

Table 1 – Test Hours in Local Time

Test Results

The tests permitted us to uniquely evaluate the possible PEP proxy conditions as shown in Table 2.

		Source-to-First Proxy Distance	
		Near	Far
Satellite Congestion	Low	A	C
	High	B	D

Table 2 – The Four Possible PEP Proxy Conditions

To understand the test results, we use terms with the following definitions.

Provisioned Bandwidth is the theoretical capacity of a single channel/circuit to pass bits over a unit of time (bps). However, provisioned bandwidth is typically associated with capacity near a user and often shared among many users (aka the "last mile"). Despite the sharing, each TCP connection is expected to operate at the provisioned bandwidth when it is transferring data. If there is an explicit traffic shaping mechanism operating on the circuit that limits the bandwidth to a class of service, then the limited bandwidth becomes the provisioned bandwidth. In other words, TCP should be able to transfer payload at this rate unless there is congestion preventing this rate within this service class.

Effective Bandwidth is the number of bits per second (bps) successfully transferred from sender to receiver on a single TCP connection through the above provisioned bandwidth within an end-to-end system of networks and links. Effective bandwidth is less than provisioned bandwidth due to limitations imposed by the TCP bandwidth-delay product. But it may also be limited by processing capacity of the source and destination devices, and contention from other users of the provisioned bandwidth. This is the "real" single flow bandwidth seen by users in a network system.

Effective-Provisioned Bandwidth Ratio (EPB Ratio) is the percentage of the provisioned bandwidth that is successfully used by the single TCP connection within a network system. It is the single connection effective bandwidth divided by the provisioned bandwidth.

NetForecast calculated the effective bandwidth (all bytes transferred divided by the elapsed test time) for each test. We also calculated the quantity of bytes transmitted on the satellite relative to the number of bytes sent or received to/from the satellite. The additional traffic on the satellite was due to PEP retransmissions of payload that was prematurely ACKed. As Figure 1 shows, when the data source was a long distance from the ground station and there was little satellite congestion—condition C—effective bandwidth had a wide range of poor performance with one-third of the tests delivering an EPB ratio worse than what could be delivered with the EPB limitation. Under condition D, the performance became unacceptable for all tests with extremely poor EPB ratio and at time nearly 90% retransmission.

The striking finding shown in Figure 1 is that the four conditions (A to D) are not on a continuum. They are clearly different clusters of performance. The performance zones are pre-determined by the network conditions. Table 3 below shows that average values of the performance zones.

	Near Scenarios		Far Scenarios	
Satellite Link Congestion	Proxy is Near To Source		Proxy Is Far From Source	
	EPB Ratio (high % is good)	Retransmissions (low % is good)	EPB Ratio (high % is good)	Retransmissions (low % is good)
Low	57%	0%	12%	2%
High	40%	1%	2%	36%

Table 3 – Results Summary Of the Four Performance Zones

Independent Validation of Findings

The issue of degraded performance as the source to first proxy distance increases is documented by other sources. Yuichi Nashida, et al [2] show a simulation of the phenomenon across a range of distances based on 10 Mbps source network bandwidth. More interestingly, Bequant S.L. in Madrid, Spain supplies PEP solutions performed tests with different source-to-proxy distances as documented on their website [3]. We learned the provisioned speed of the wireless LTE tests. Given the effective and provisioned bandwidth of these other sources, we applied the simple EPB ratio formula to the various RTT distances. The results for all three sources align very well as shown in Figure 5.

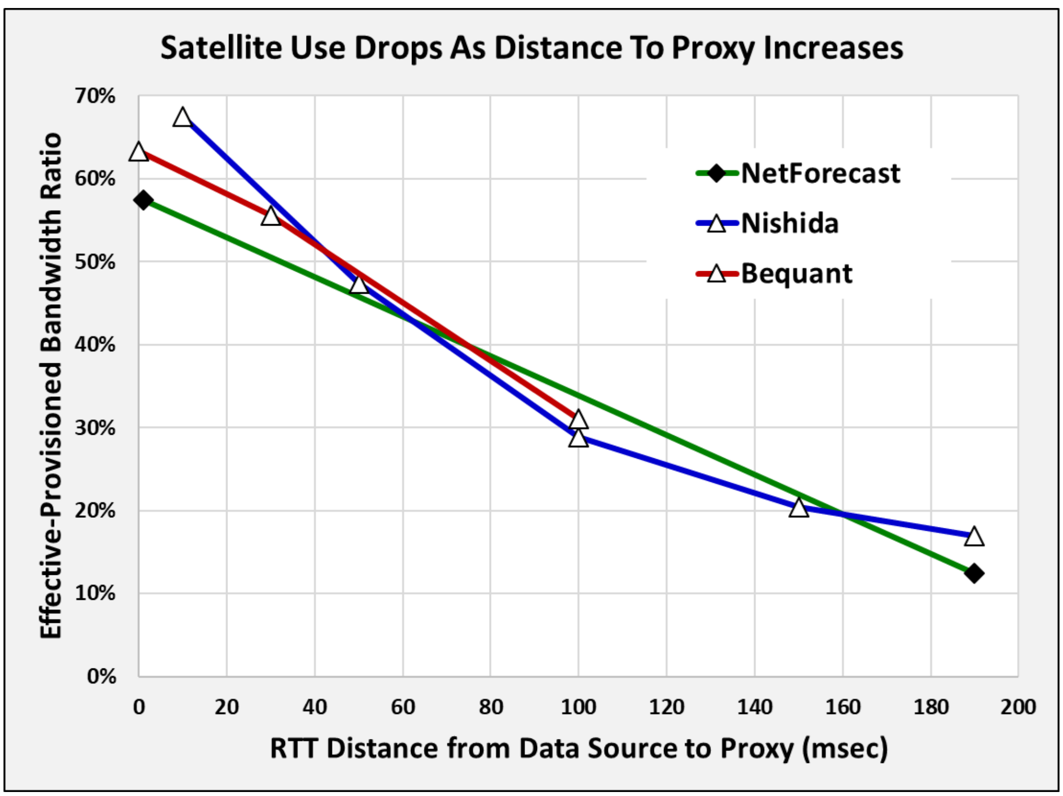

Figure 5 – Collaborative Proxy Distance Degradation Evidence from Other Sources

When is a PEP Proxy Too Far From the Data Source?

In order to be of practical use to network engineers, we need to identify a boundary between near and far distance to the first proxy. Such a goal is subject to many system considerations. We chose the point in Figure 5 where EPB drops below 50%. Using the NetForecast curve, that is at a 30 ms RTT.

NetForecast operates a large system of probes across the continental United States that continuously tests to major internet peering and hosting locations within the United States. Among the tests performed are packet “pings” which measure RTT. The system generates more than 600,000 pings test results per day. We know the locations of both ends of each ping test so we can calculate the great circle distance traversed by each ping. Mining this data shows that on average, a 30 msec RTT is achieved at 2000 km (1200 mi) for fiber-based transit network conditions.

The implication is that TCP data coming from a datacenter located within a 2100 km radius of a satellite ground station will be able to deliver better than 50% of the provisioned bandwidth using PEP.

Implications for Aircraft Wi-Fi Services

Satellite-based in-flight communications (IFC) services are comprised of three network elements: Wi-Fi within the plane, an air-to-ground satellite link, and a terrestrial path over the internet. This study does not consider the added issue of an on-board Wi-Fi network which may have access or congestion issues. Thus, we are only addressing two-thirds of the major networks in an IFC system. However, we are addressing the internet segment which many IFC vendors appear to brush off with “the internet is something that we can’t do anything about.”

Satellite IFC-enabled aircraft are equipped with a Very Small Aperture Terminal (VSAT) dish antenna capable of tracking a satellite during flight. VSAT access satellites in geosynchronous orbit relay data to and from earth stations. Data passes to and from the satellite link through a modem on the aircraft and another in the ground station. The ground station location may change during flight to optimize signal strength or avoid cloud cover at the ground station. When the ground station location changes the point of entry into the internet changes.

Note that PEP proxy 1 is always on the ground intercepting traffic from content server to the satellite link. It can be anywhere but is often located relatively near the satellite ground station. The PEP2 proxy must be on the aircraft for the system to operate properly. Proxy 1 and 2 roles flip when content is traveling from the aircraft to the ground. An example of such a pattern would be a user streaming video from the plane to someone on the ground. The PEP software is designed to handle both scenarios. This report focuses on the more typical passenger consuming content from the group scenario.

Because of PEP, the physical distance between the ground station and content/services a passenger wishes to access over the internet dramatically influences the passenger’s IFC experience quality.

We make the following assertion. A passenger aboard a commercial aircraft will likely use the IFC service like they would from their home residence. They will go to the same news, weather, email, banking, shopping, etc., services. After all, that is where they have their accounts, and those services are personalized to meet their needs. Given this use pattern, their TCP connections will be near their home city—in fact probably within 2100 km of their home. The airline must, therefore, provide good quality service to the ground AND to each passenger’s home area.

Consider a hypothetical flight carrying four IFC-connected business travelers native to London, Lisbon, Chicago, and Dallas on a flight from Paris to Washington DC. Early in the flight the satellite service is homed to a Frankfurt ground station. Figure 6 shows their probable internet land paths from the Frankfurt ground station to the destination cities. London and Lisbon are within the 2100 km radius. Chicago and Dallas are not.

On Figure 6 land paths are shown as black lines while the satellite links are shown as red lines. In either case, solid lines indicate near distances (less than 2100 km) while dashed lines indicate far distances.

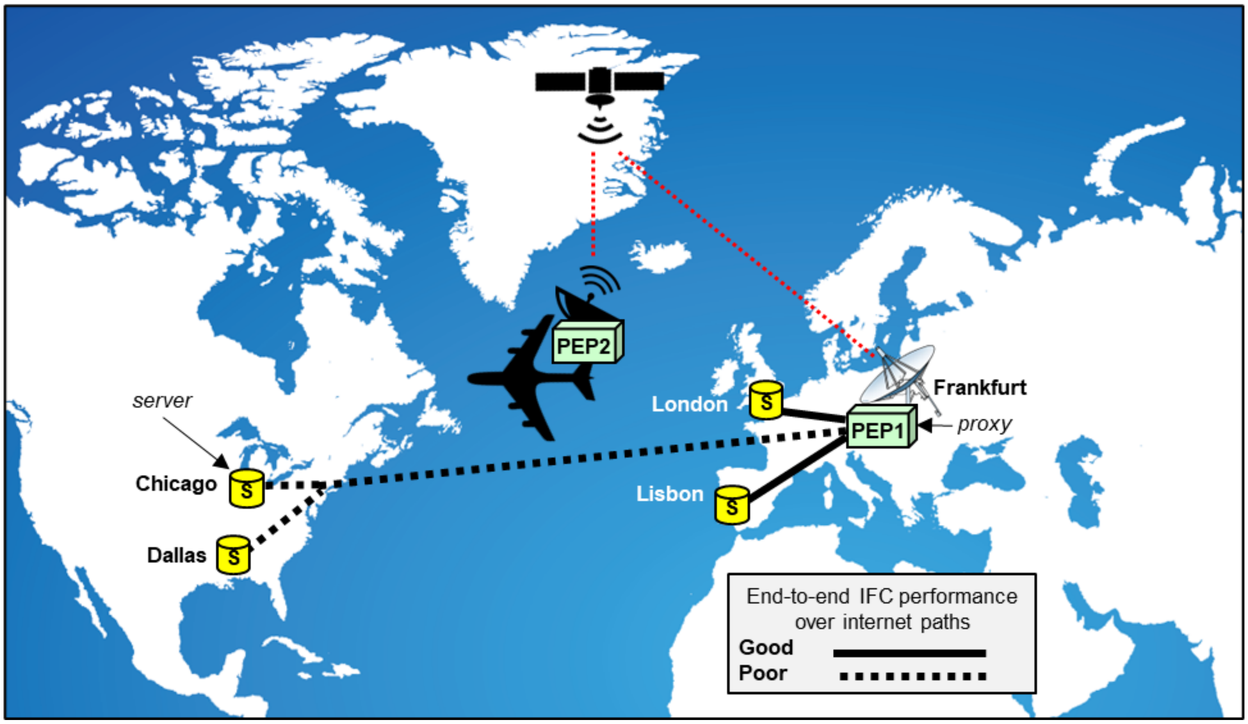

Figure 6 – Example of Internet Paths from Frankfurt Ground Station

Partway through the flight, the satellite service changes to a New York ground station. Figure 7 shows the new land internet paths. This change improves service for the Chicago native, but degrades QoE for the London and Lisbon natives. The Dallas native never receives good service.

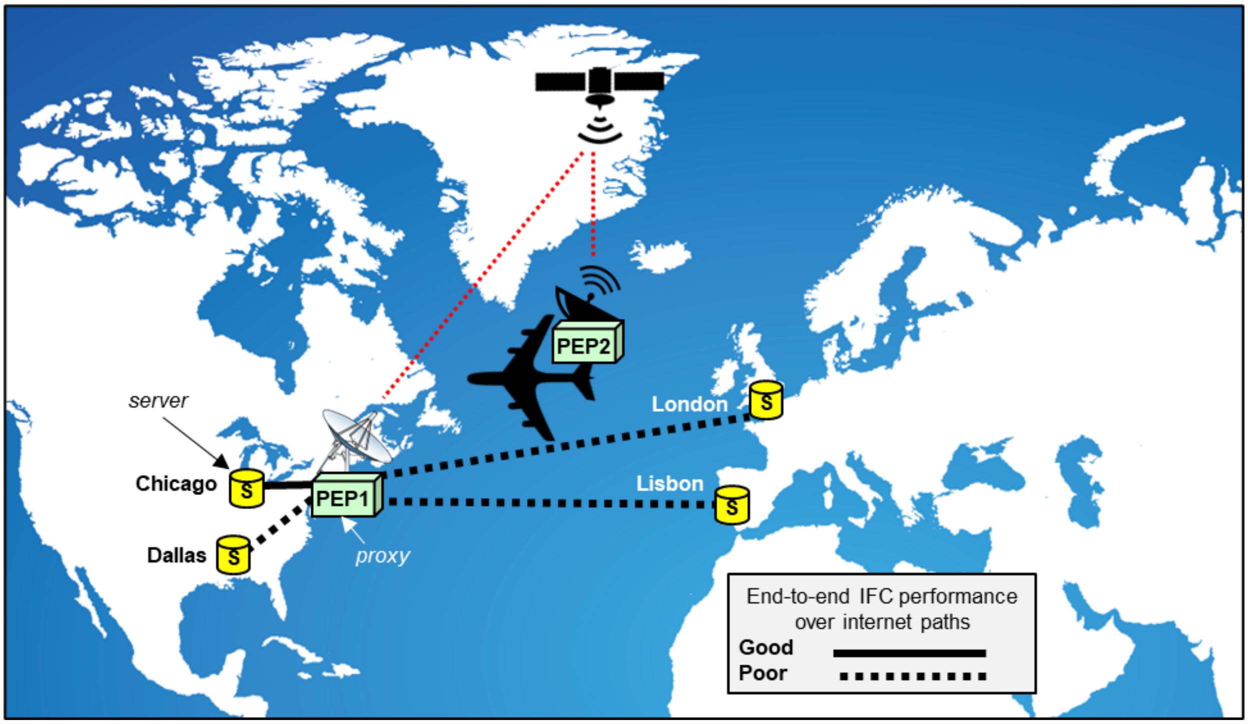

Figure 7 – Example of Internet Paths from New York Ground Station

Satellite services operate several satellite hub ground stations for diversity and general capacity management. The ground stations are widely distributed around the globe with only a few on each continent, and many of them are in rural locations. A service supporting an airline will dynamically assign an aircraft in flight to one – or very few – ground stations based on a variety of factors optimized for the satellite service. The ground locations have a material impact on internet content performance to the aircraft.

	IFC E-E Performance	
Passenger Home City	Frankfurt Hub	New York Hub
London	Good	Poor
Lisbon	Good	Poor
Chicago	Poor	Good
Dallas	Poor	Poor

Table 4 – End-to-end Performance Summary for the Hypothetical Passengers

The radius of satisfactory performance is centered on the first location with diverse peering and paths to the general internet. For example, a satellite service provider might prefer to place its ground station in a place like Boise, Idaho due to generally sparse cloud cover that could degrade the satellite signal. The typical path from the ground station into the internet in this example is a dedicated fiber circuit into a major internet peering point in Seattle Washington, a distance of 640 km. Under this scenario, the true radius of satisfactory performance is 1460 km centered in Seattle. In this case, the backhaul from the internet to the ground station consumes one-third of the satisfactory distance budget!

Conclusions and Recommendations

NetForecast's real-world testing shows that Performance-Enhancing Proxy (PEP) protocols used in satellite networks to improve the user experience can often dramatically actually degrade performance for content accessed over long terrestrial internet paths. This protocol behavior affects internet-connected airline passengers when the content or services they need are far from the satellite ground PEP proxies servicing their aircraft throughout a flight.

Consistent with a warning in the PEP RFC [1] that: "potential negative implications associated with using PEPs . . . related to the possibility of breaking the end-to-end semantics of connections . . . is one of the main reasons why PEPs are not recommended for general use", NetForecast confirms that when used on a satellite link to access the internet, *PEP does indeed break end-to-end connection semantics, causing poor data transfer rates which, in turn, cause high retransmission rates that needlessly increase traffic on the satellite link.*

To alleviate this problem, NetForecast recommends IFC suppliers make significant improvements to PEP. This work should begin by having the PEP technology suppliers publish how their protocols operate. To-date, the proxy vendors have kept their protocols unique and secret. This competitive wall assures that PEP gateways are not interoperable across vendors. It is one of the last bastions of closed systems in an otherwise open internet world.

Eliminating PEP may look like a simple solution. But when properly applied, PEP does outperform the basic TCP connection suffering from unavoidable satellite BDP.

Recommendations

- IFC suppliers should carefully design and implement the ground delivery architecture of their services, being mindful of the source-to-proxy distance on the internet. This will require a new system view of how ground stations are selected, and which ground-based proxies should be engaged during a flight.
- IFC suppliers should consider partnering with international CDNs to have a more robust PEP proxy deployment and dynamic internet pathing to better accommodate passenger needs. However, this will be difficult without open PEP protocol standards.
- IFC providers should monitor the actual round-trip times via the transit providers onto the broader internet. NetForecast has often found paths taking twice as long as expected due to incorrect transit routing.
- Airlines should continuously measure the end-to-end IFC passenger experience to better understand and improve it.

About the Authors

Peter Sevcik is the founder and CTO of NetForecast and is a leading authority on Internet traffic and performance. Peter has contributed to the design of more than 100 networks, including the Internet, and is the co-inventor of three patents on application response-time prediction and congestion management. He pioneered Internet usage tracking techniques and invented the Apdex methodology. He can be reached at peter@netforecast.com.

Rebecca Wetzel is a director at NetForecast, and a data communications industry veteran. She helped realize the commercialization of the Internet in its formative years and worked to design and market some of the Internet's first value-added services. She also spent many years as an Internet industry analyst and consultant.

References

1. IETF RFC3135, *Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations*
2. IETF RFC7323, TCP Extensions for High Performance
3. Yuichi Nishida, Wang Huit, Hiroaki Matsumoto, Tokumi Yokohira, and Yukinobu Fukushima, *Retransmission Control in TCP with a Performance Enhancing Proxy*. IEEE International Conference on Advanced Communication Technology, March 2008.
4. Bequant, http://bequant.com/bta.html

Additional information is available at: www.netforecast.com