

ERRORS, LOSS, THE EFFECT OF ATMOSPHERIC CONDITIONS AND CHOICE OF WAVEBANDS

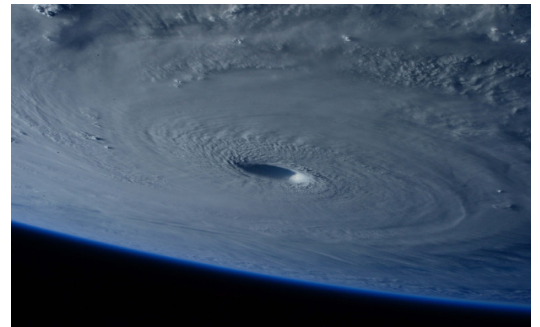
WHAT'S INSIDE

1. Atmospheric Conditions and SATCOM Data Transmission
2. A Quick Look at how Data is Transmitted in Packets
3. Increasing Bandwidth vs Transmission Quality
4. Forward Error Correction (FEC)
5. What Happens when Atmospheric Conditions Disturb Transmission
6. What about Wavebands?
7. Application Perspective on Layer 1 Effects
8. Should We Care About Atmospheric effects?
9. How Can You Test Your Applications with Satellite Bit Error, Loss and Bandwidth Limitations (as well as latency, jitter, etc.)?
10. Using Satellite Network Emulators

ATMOSPHERIC CONDITIONS AND SATCOMS DATA TRANSMISSION

There's no doubt that in an ideal world the transmission from Ground to Satellite and vice versa would be error free, and if that were the case there would be nothing to say here. The bottom line is that they are not error free and the problems occur primarily due to atmospheric conditions. So what are these conditions? Well, it all starts with the Sun and goes down from there. We may have:

- Space Weather - Solar Flares etc - geomagnetic effects
- Ionospheric Scintillation - Irregularities in the Earth's ionosphere which affect the amplitude and phase of radio signals
- Cloud - Water droplets absorb and reflect radio signals
- Rain - Raindrops themselves absorb and reflect radio signals
- Dust and Sand Storms



Well that's not an exhaustive list, but you get the idea.

So any of these factors can produce an error in the transmission stream and the more you get the harder they are to deal with.

A QUICK LOOK AT HOW DATA IS TRANSMITTED IN PACKETS

As humans we tend to think of transmitting bytes of data. We have "data plans" for so many Gigabytes per month, but data is not normally transferred in bytes between systems. Instead it is transferred in packets (blocks of bytes) aka Frames. These packets consist of a:

- Header - information on how to deliver the packet e.g. the destination address (and more)
- Data - the data we are actually sending

Now the data may itself contain a sort of sub-packet i.e. have a Header and Data itself, and if you think that's uncommon - no it absolutely isn't: in most businesses and homes IP packets are sent inside Ethernet Packets.

Why is data transmitted this way? Because a typical network operates like the "Post Office" handling network traffic on behalf of many customers. A packet is, to the network, like a letter is to the post office - it contains address information, including sender information, so that packets can be delivered to a variety of destinations and the recipients know where they came from.

If we sent one byte at a time it would still need a header and so the amount of header information would exceed the actual data we were transmitting by huge amounts - what a waste of bandwidth that would be!

The OSI Network Layer Model (diagram below courtesy of Wikipedia), lays out how these packets, and packets-in-packets, are carried starting at the physical layer.

A QUICK LOOK AT HOW DATA IS BEING TRANSMITTED IN PACKETS (CONT./)

OSI Layer		Protocol Data Unit	Function	
Host Layers	7	Application	Data	High-level APIs, including resource sharing, remote file access
	6	Presentation		Translation of data between a networking service and an application; including character encoding, data compression and encryption/decryption
	5	Session		Managing communication sessions, i.e., continuous exchange of information in the form of multiple back-and-forth transmissions between two nodes
	4	Transport	Segment Datagram	Reliable transmission of data segments between points on a network, including segmentation, acknowledgement and multiplexing
Media Layers	3	Network	Packet	Structuring and managing a multi-node network, including addressing, routing and traffic control
	2	Data Link	Frame	Reliable transmission of data frames between two nodes connected by a physical layer
	1	Physical	Bit, Symbol	Transmission and reception of raw bit streams over a physical medium

So in our Satcoms example the lowest layers (closest to the physical) are:

1. Satellite Physical (SATPHY), the radio (wireless) transmission as a bit or symbol stream
2. Satellite Medium Access Control (SMAC) & Satellite Link Control (SLC)
3. IPv4 or IPv6

From Layer 3 up the packets are the same as the ones our computers, devices, phones etc generate.

INCREASING BANDWIDTH VS TRANSMISSION QUALITY

Let's talk about the physical transmission layer (OSI Layer 1) in SATCOMS and note that it might be in Bits or Symbols.

What's a Symbol? Well SATCOMS looks at transmission at the lowest level in Hz (Hertz - cycles per second). Now we could send data one bit per cycle in the standard binary fashion, as typically happens in wired and optical circuits, or if conditions allowed, we might try to have several different "levels" per cycle and encode 2, 4, 8, 16, 32 bits or more in a cycle. These are called Symbols.

Technically, the method of doing this is to modulate the signal. In a popular form of modulation: 64-QAM (64 Level Quadrature Amplitude Modulation) both amplitude and phase modulation are used.

The problem is, the higher the encoding levels, the better we need the transmission quality to be and all the atmospheric factors mentioned above can dash that by interfering with amplitude and phase, and more. A solution to this is to use Forward Error Correction (FEC) but this decreases net throughput - see below for more on Forward Error Correction.

FORWARD ERROR CORRECTION (FEC)

A standard network approach to error correction is to send data, then wait for an acknowledgement (ACK) from the receiver and if none is received then resend the data. Well at least that's it at its most simple. This kind of system is used by the TCP part of the IP networking stack, for example.

The problem with this method is that if you have large round trip latencies of, say, 700ms (GEO orbit) then it will take over 700ms to get a retransmission of the data. This would seriously hamper transmission rates.

Enter Forward Error Correction (FEC): If we can send some redundant data with the real data that allows for the correction of one or more errored bits then we don't need to retransmit the data - thus saving at least 700ms in the example above, at the expense of sending more data than required.

WHAT HAPPENS WHEN ATMOSPHERIC CONDITIONS DISTURB TRANSMISSION

1. Unlike standard Ethernet at Layer 2 which has no error correction at all, many SATCOM circuits support Forward Error Correction (FEC), as explained above. When it works, not much additional delay is incurred.
2. Ultimately, if there are too many errors to correct, the encoding level could be reduced making the data more likely to be successfully decoded.
3. If there are more errors even than the above can fix then we have bit errors getting through at Layer 2, and for IP a checksum will fail somewhere at Layer 3 or above and the packet will be discarded, and require re-transmission.

WHAT ABOUT WAVEBANDS?

There is no magic formula for how wavebands perform because different SATCOMS providers may have different power outputs and therefore better Signal to Noise ratio, but the general trends are:

Higher Frequencies implies higher throughput but also higher susceptibility to attenuation by rain/cloud etc.

Here is a table we put together as a quick guide, but the more we looked into it the more complex it got as you have to take individual services into account due to power, adaptive coding and modulation (ACM), or not etc.

Waveband	Frequency	Throughput (Bandwidth)	Rain/Cloud Resilience
L-Band	1-2Ghz	400kbps	Premium
C-Band (inc VSAT)	4-8Ghz	Cost Effective	Good
X-Band (Inc VSAT)	9-12Ghz	Similar to C	OK
Ku-Band (Inc VSAT)	12-18Ghz	1-12Mbps	Susceptible
Ku-Band HTS Spot Beam	12-18Ghz	80-200Mbps	Susceptible
Ka-band (inc VSAT)	26.5-40Ghz	30-50Mbps	Very susceptible (but modern Ka has a lot of power to compensate)

APPLICATION PERSPECTIVE ON LAYER 1 EFFECTS

The application is, in general, going to experience a few things:

1. Lowering of the available Bandwidth, where FEC repeatedly fails e.g. ACM mentioned above
2. Loss of data for unacknowledged Layer 4 protocols e.g. when the transport layer (4) is UDP
3. Re-transmission of data for guaranteed Layer 4 protocols e.g. when the transport layer (4) is TCP

So, if we want to test an application for these effects we need to be able to produce similar effects for Layers 2 (&3) which will have a similar impact on the transport layer (Layer 4) and above.

SO AGAIN, SHOULD WE CARE ABOUT ATMOSPHERIC EFFECTS?

Summarizing:

- For TCP-based applications – http, https, cifs (NetBIOS), ftp, buffered video, buffered audio etc. reduction in bandwidth and retransmission due to packet loss are significant factors

Fundamentally we will see a slowdown in transmission which may be very significant

- For UDP-based applications - VoIP, Real Time Video, Telemetry etc

Humans have trouble with breakup and quality loss in live video and voice calls and video conferencing

Telemetry may be lost

As ever, the consequence depends on the application.

HOW CAN YOU TEST YOUR APPLICATIONS WITH SATELLITE BIT ERROR, LOSS AND BANDWIDTH LIMITATIONS (AS WELL AS LATENCY, JITTER ETC.)?

[If you read Part 1 or Part 2 of this satellite blog series then you can skip to “The End” – the arguments are similar and you can “also” simulate Bandwidth Restriction, Bit Errors, Loss, Latency and Jitter. If you didn’t please read on...]

You need to test!

That may not be as formal as it sounds: we could say you need to try the application in the satellite network. However, there are issues with testing or trying using actual (real) satellite networks:

- Satellite time is expensive and the equipment not at all easy to deploy
- It will be just about impossible to mimic your or your customers’ real locations
- If you find an issue which needs attention, getting it to the developers for a solution will be difficult (and if the developers say they have sorted it out it is likely to be very difficult to retest)
- You won’t be able to try out other satellite environments e.g. MEO or LEO without purchasing them
- You won’t be able to have a rainstorm appear just when you need it during your testing.

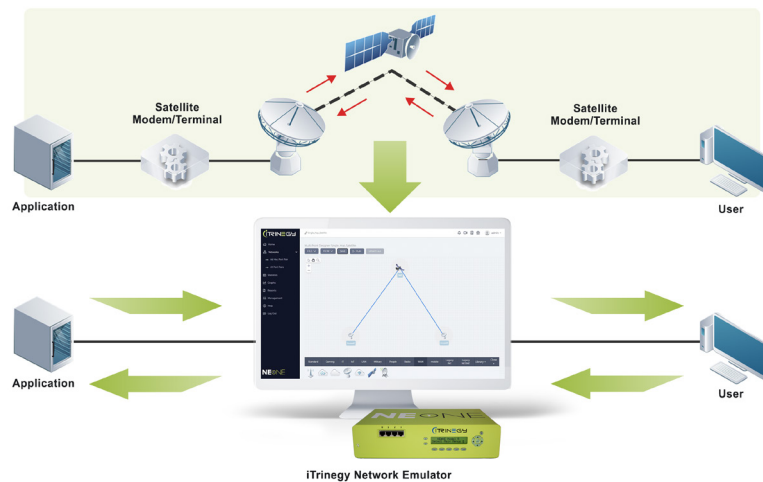
USING SATELLITE NETWORK EMULATORS

Because of the issues of “real network testing” in Satellite networks we’ve brought Satellite Network Emulation capabilities to our NE-ONE Professional and NE-ONE Enterprise Network Emulators.

People think of anything with the name “emulator” in it as some sort of complex mathematical device which predicts behaviors.

They may be complex, but only internally. Externally we make them very straightforward. And, they don’t predict behavior, you get to actually try out (“test”) your application using your real clients and servers just as though they were in the satellite network.

All you need to do is plug them in between a client device and the servers and set them for the satellite situation you want. You can even try out other options like LEO or MEO within seconds.



Plugging them in is easy because they have Ethernet ports, you don’t need any satellite equipment at all.

The End

NE-ONE EMULATORS

With iTrinegy’s NE-ONE Professional, you simply select and run the type of point-to-point or point-to-point multi-point, dual-hop network scenario you require. You can also dynamically adjust the network conditions using the Network Scenario Builder.



NE-ONE ENTERPRISE

Where you need to map multiple network links or large numbers of devices in a more complex network environment, our NE-ONE Enterprise with its network drawing interfaces and ability to connect 200+ devices is the ideal choice.



VIRTUAL APPLIANCES

Both the NE-ONE Professional and NE-ONE Enterprise are also available as part of our Flex virtual appliances range, certified to run on VMware’s ESXi Server.

