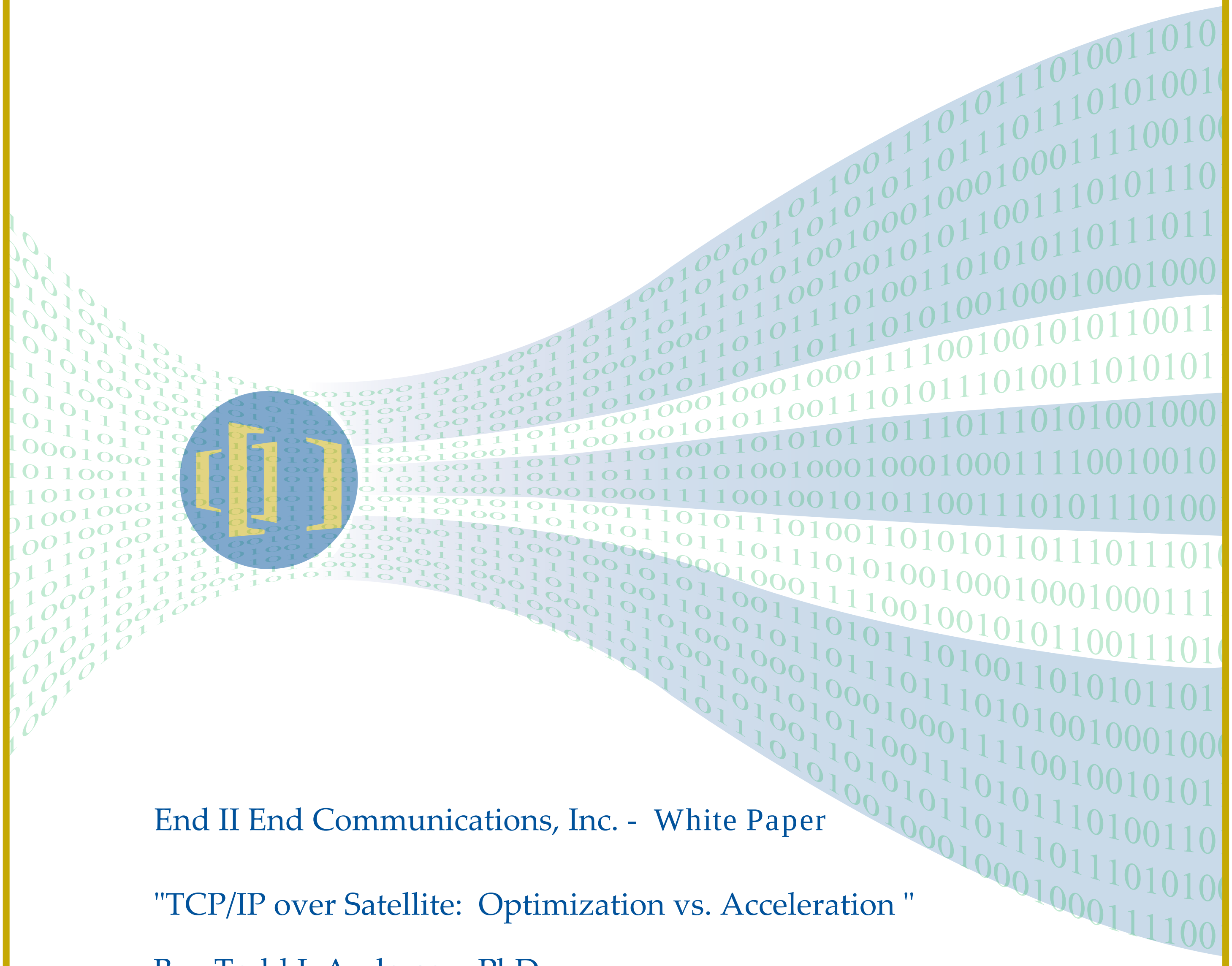
End II End

COMMUNICATIONS, INC.™



End II End Communications, Inc. - White Paper

"TCP/IP over Satellite: Optimization vs. Acceleration "

By: Todd J. Anderson, PhD

Dated: April, 2005

Extend] Your [Enterprise™

TCP/IP over Satellite: ACCELERATION vs. OPTIMIZATION

The Problem of Satellite Latency

Satellite communications have numerous advantages over terrestrial data connections (DSL, Frame Relay, etc.) including mobility, network topology, and the "anywhere and everywhere" benefit of global coverage. These advantages come at the price of increased data transit times, also known as latency.

The large latency associated with satellite communications originates in the 125 ms required for data packets to travel at the speed of light from the earth's surface to any one of the communication satellites positioned over the equator. This implies a minimum round trip time (up and down and back again) between the source and destination of a data session of at least 500 ms (Figure 1).

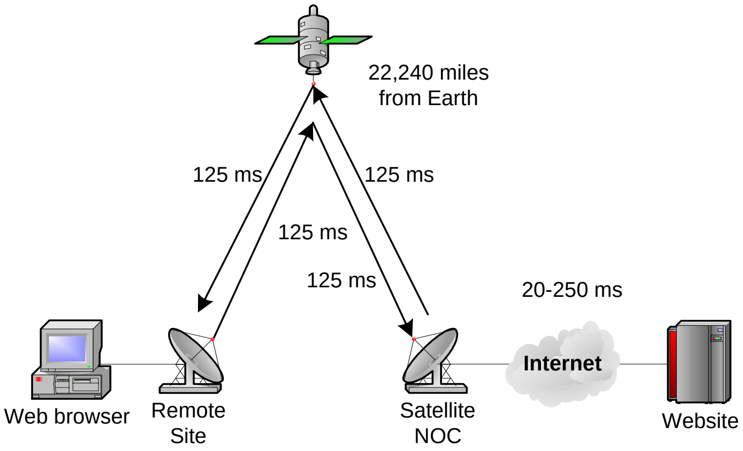


Figure 1. Total Round Trip Time using Broadband Satellite

Unfortunately, neither client–server applications nor the TCP protocol are designed to accommodate these high latencies. Most applications are written without regard for network latency, i.e. as though the client and the server are connected over a high speed Local Area Network (LAN). The TCP protocol was designed for use on terrestrial networks where latencies seldom exceed 250 ms to travel around the world over copper or fiber circuits. The satellite industry has developed a number of technologies to minimize the effects of latency on TCP satellite communications, which are most often termed "TCP Acceleration."

Protocols: A Quick Lesson

TCP (Transmission Control Protocol) is the dominant transmission protocol of the Internet. Web browsing, email, SAP, Oracle, Citrix, Lotus Notes, etc. all use TCP packets to communicate between the client (ex. your web browser) and the server (ex. web server hosting the website you're viewing). Remember, both the client and the server send and receive data during a TCP session. (Clicking on a hyperlink or filling in a text box and clicking "Go" sends data to the web server, which then responds by sending new web pages).

Each data packet exchanged between the client and server can be broken into 2 parts: the header and the payload.

- The payload is the data for your application, e.g. the bytes that represent the text and images of the web page you are trying to view.

- The header contains all the information needed to transport the packet across the Internet: the source and destination IP addresses (IP Header), and the session control data (TCP Header).

IP Header	TCP Header	Data (Payload)

Figure 2. A Simplified TCP Packet

TCP session data consists of groups of bytes that count and order the packets to ensure:

- Sent packets are the same as the received packets (no transmission errors)
- All the data packets sent are received (no packets lost)
- Packets are assembled in the correct order (packets often arrive out of sequence)
- Source and destination ports and other information

The sender and receiver in a TCP session also exchange special packets or “acknowledgements” to let the sender know when the receiver receives each packet. These special packets provide a mechanism for the sender to vary the rate at which packets are sent based on the rate at which acknowledgements are received back from the receiver. Because TCP has these control features and a flow control mechanism, TCP is termed a stateful and synchronous protocol.

Part of the TCP flow control includes a mechanism called “slow start” where just a few packets are sent initially and the sender increases/decreases the rate at which packets are sent in response to the rate at which acknowledgements are received. TCP wants to maximize the data rate, but not at the expense of having to retransmit data. In other words, lost packets carry the biggest penalty and are to be avoided.

By comparison, UDP (Universal Datagram Protocol) the other common protocol on the Internet is a stateless, asynchronous protocol, most often used for file transfers, video and audio streaming. The UDP header is very minimal and carries no data about session state and there are no special packets for flow control. UDP just sends packets as fast as possible right from the start and hopes the receiver gets them all. If there is packet loss during a UDP session, then UDP finishes the session and starts all over again, having to retransmit ALL the data, not just the missing packets. In contrast, if packet loss is detected during a TCP session (due to a congested network or a slow receiver), the TCP protocol will slow the data rate, and resend just the lost packets. In that sense, TCP is a “smart” protocol and UDP is a “dumb” protocol.

So what’s the problem with TCP sessions running over satellite? Simply put, it’s latency, or those long round trip times across a satellite network. For comparison, typical latency on a LAN is less than 5 milliseconds (ms). WAN latency using cable modem or DSL is in the range of 20-250 ms. and Satellite latency starts at 500 ms and can easily exceed 2000 ms. The TCP flow control mechanism interprets large latency values (long times between acknowledgements) as evidence of a congested network or packet loss and will not increase the rate at which it sends packets, even though there is no actual congestion or packet loss across the satellite link. TCP thinks a satellite network is always “congested and unreliable.” The sender never increases the sending rate and TCP session throughput never increases from the initial slow start rate.

TCP Acceleration for Satellite Links

Since TCP is the standard for data communications worldwide over the Internet, satellite operators have developed technologies to make TCP sessions perform better by minimizing the effects of high latency intrinsic to satellite links. These technologies fall into two basic categories: acceleration and optimization.

There are a number of different forms of TCP acceleration. The first is the use of Performance Enhancing Proxies (PEPs), which all have one important common feature. All PEPs involve techniques that change the TCP header data before and after the satellite link to hide the high latency of the satellite link from the TCP session.

TCP spoofing is one PEP technique that imitates a terrestrial TCP session by sending false TCP packet acknowledgements. The satellite modem can contain the spoofing software or an additional device can be deployed at the remote site and at the satellite carrier's Network Operations Center (NOC) (see Figure 3). The spoofing device acknowledges receipt of the data packet locally (from the sender) as if it were the receiver and the TCP session proceeds as though it was occurring just across the LAN, i.e. the TCP flow control mechanism sends packets at the maximum rate supported by the LAN connection. On the other side of the satellite link, the spoofing device/software suppresses the real acknowledgments from the actual receiver.

The spoofing devices/software contain storage buffers to allow the transmission of data across the satellite links and deal with lost packets/re-transmissions between the two spoofing points, since the actual sender and receiver think the session is always fine. Other PEP techniques include adjusting packet size, data rates and other TCP parameters to more closely match the characteristics of the satellite carrier protocols. In all cases, values are changed in each TCP packet header in both TCP-PEP devices.

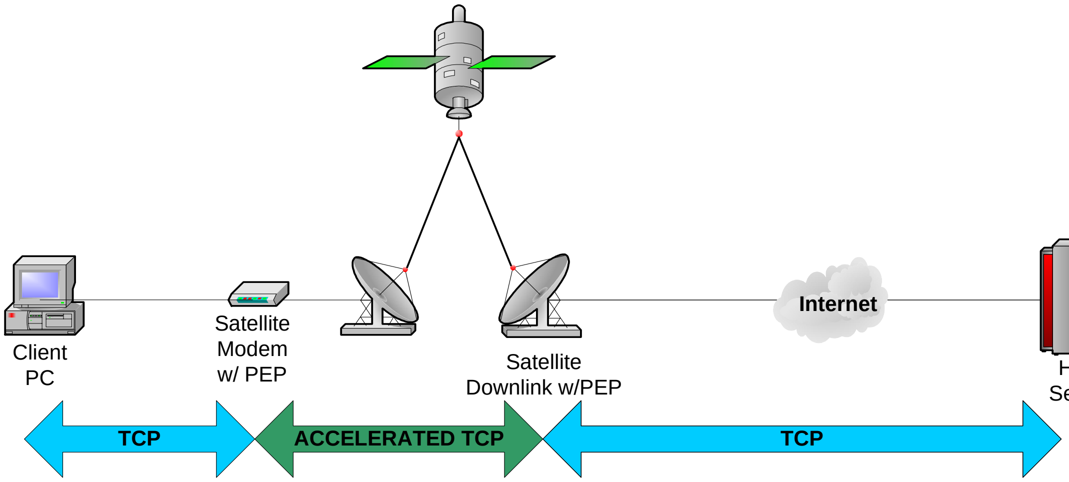


Figure 3. TCP Acceleration via PEP in the Satellite Modem

Another common technique is TCP multiplexing (see Figure 4). The proxy device/software spoofs the TCP session with the sender and receiver on each side of the satellite link. The proxy device/software converts the single incoming TCP session into multiple data sessions (using TCP, UDP or another protocol) to accelerate data transfer rates across the satellite link. On the other side of the link, the multiple data streams are recombined into a single TCP session and transmitted to the receiver.

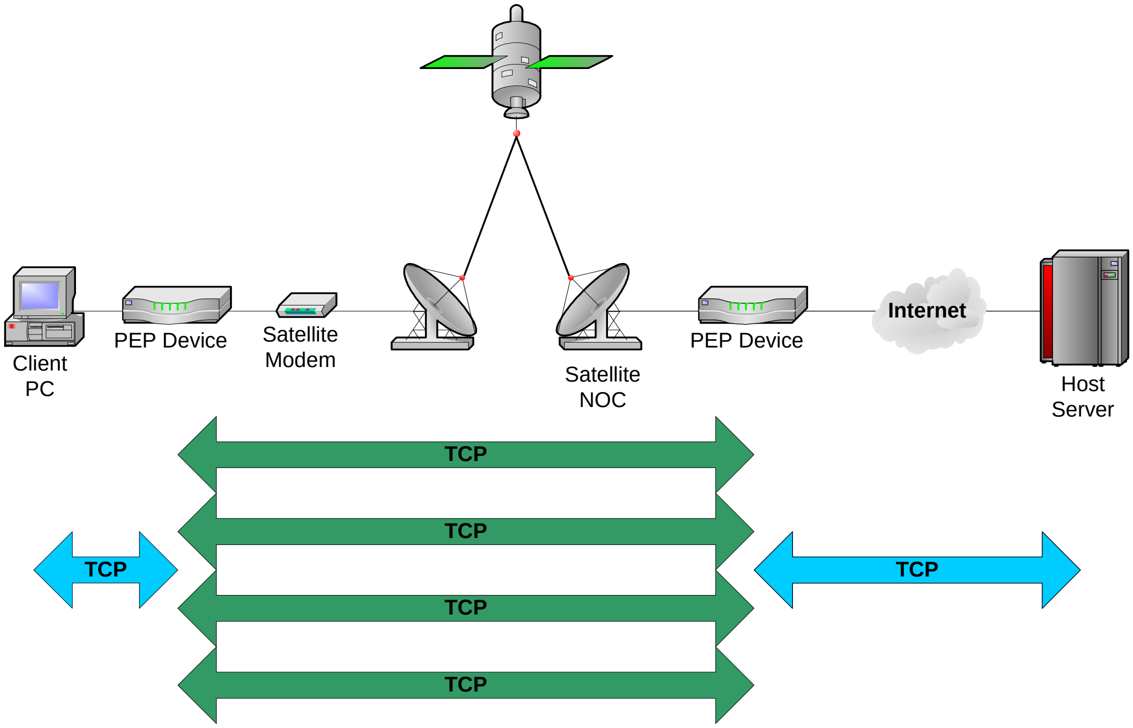


Figure 4. TCP Acceleration via TCP Multiplexing using Standalone Devices

This form of acceleration minimizes the effects of latency by turning a single TCP session into multiple simultaneous data sessions. This technique is effective for long TCP sessions that contain large amounts of data such as file transfers, database synchronization and video streaming. This "series to parallel" conversion of packet transmission utilizes more of the available bandwidth on the satellite link to shorten a TCP session and minimize the number of acknowledgements. However, it is not effective when used with interactive or "chatty" applications, where the client-server session consists of the exchange of large numbers of small data packets. Examples of "chatty" applications include most legacy mainframe applications, SAP, Peoplesoft, and Lotus Notes.

Security Considerations

Data security concerns have imposed certain criteria for enterprise Wide Area Networks. Virtual Private Networks (VPNs) have become standard requirement for companies using the public Internet to connect their remote sites to the headquarters. VPNs encrypt a company's data traffic when traveling over the public Internet so that the data streams cannot be easily intercepted and viewed. VPNs also authenticate the data to ensure it has not been altered while in transit across the Internet. A VPN between two sites constitutes a private connection (across a public network) through which data can securely pass between hosts at the two sites. VPN packets are much different than plain TCP packets, as shown in Figure 5.

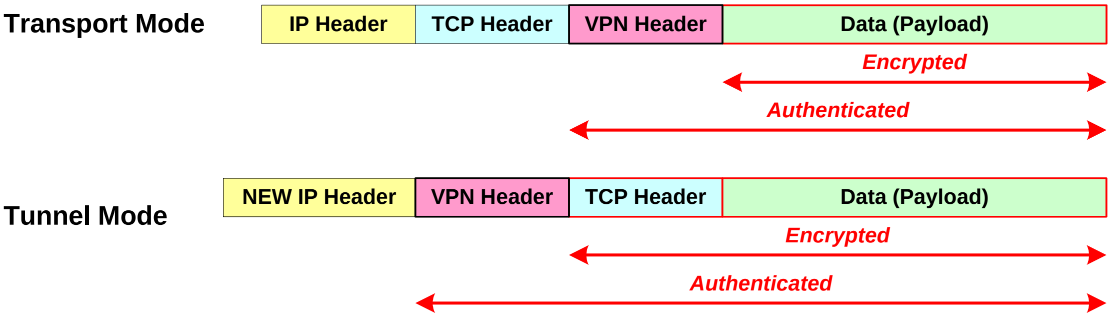


Figure 5. VPN Tunnel Mode vs. VPN Transport Mode

VPNs operate in two modes, tunnel and transport. When a data packet enters into a tunnel mode VPN connection, the whole data packet (TCP header and payload) is encrypted and given a new header, thereby the original packet becomes the encrypted payload of a new VPN tunnel packet.

In transport mode, only the payload of the original data packet is encrypted; the original TCP packet header becomes the new header and remains unencrypted. Transport mode is fundamentally less secure than tunnel mode because the data header of the original packet is still used, i.e. the source and destination IP addresses of the two hosts are still used, and all the TCP session data remains in clear text in the header when traveling over the Internet.

Tunnel mode VPNs have become the corporate security standard because of the superior security features. By completely concealing (through encryption) the original data packet header and payload, the packet is impervious to the "man in the middle" attacks used to intercept, record and retransmit TCP sessions as the packets traverse the public Internet. With the header of the original data packet fully encrypted, no information can be obtained about the original TCP session running between the client and server. In transport mode, only the original payload is encrypted and the header is left unencrypted and subject to the prying eyes of a would be attacker. As such, transport mode VPNs are seldom used because they do not meet companies' security criteria for modern wide area networks.

Problems with VPNs and TCP Acceleration

Well, these TCP acceleration methods all sound good to improve TCP data rates over satellite, so what's the problem? Recall how all the TCP acceleration technologies rely on altering the TCP header of each data packet. Herein lies the problem when combining TCP acceleration and VPNs to deliver secure, high performance data communications over satellite. The conflict arises in the TCP session data in the original packet header.

If the data packet enters a VPN in tunnel mode before the TCP acceleration takes place, then two problems arise. First, the original TCP packet is completely encrypted (header and payload) and so the session header data is unavailable to be altered by acceleration. Second, applying TCP acceleration methods to VPN packets would alter the VPN header and/or spoof the VPN acknowledgements and violate the authentication safeguards which require the VPN packet (and headers) remain unchanged in transit between VPN endpoints. The only alternative is to reduce the level of VPN security by utilizing transport mode and leaving the packet headers unencrypted. Even then, some forms of TCP acceleration and VPN transport mode both attempt to alter certain values in the TCP packet header and may not function together at all (Figure 6).

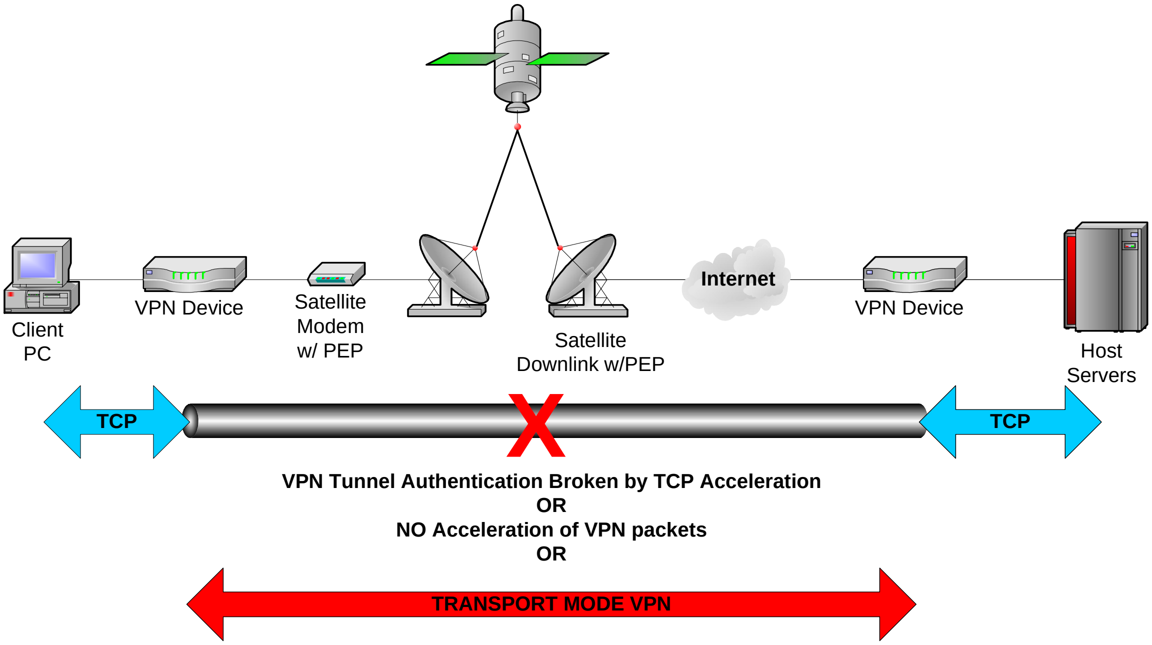


Figure 6. Conflicts Between VPNs and TCP Acceleration in the Satellite Modem

One form of TCP acceleration that is compatible with tunnel mode VPNs is the TCP multiplexing technique described earlier. A single TCP session is converted into multiple parallel TCP or UDP sessions, all of which enter and exit the VPN tunnel as separate sessions to be recombined into a single TCP session at the TCP multiplexer on the other side of the VPN tunnel. Again, there are several shortcomings to this solution (see Figure 7).

- Stream multiplexing does not help most enterprise applications across satellite links, unless large data transfers are involved.
- When deployed as a separate device, TCP multiplexing is not only expensive to configure and deploy, it also complicates network troubleshooting and support of the multiple devices deployed at the remote sites.
- TCP multiplexing does not offer any performance increase if the software is located inside the satellite modem, as the TCP packets have already been encrypted by the VPN tunnel endpoint in front of the satellite modem. As a result, the TCP packets cannot be separated into multiple parallel sessions without altering the VPN packet header and violating the VPN authentication safeguards.

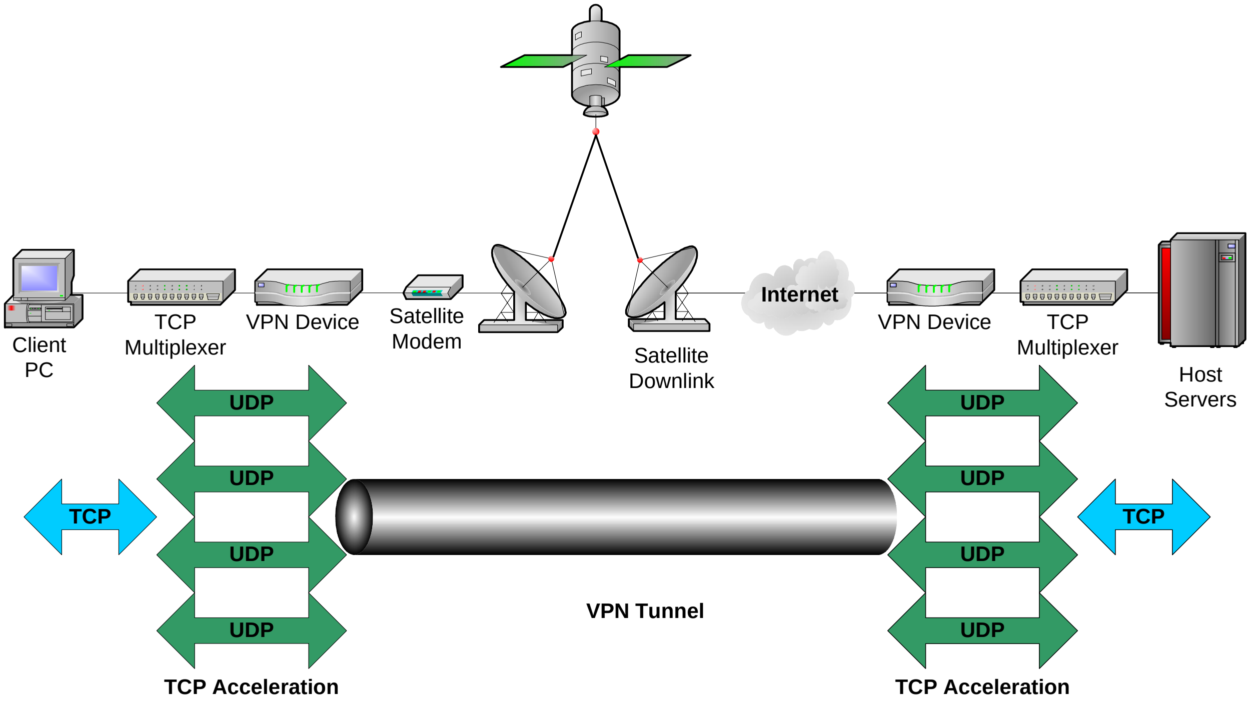


Figure 7. Complicated and Expensive Combination of Multiplexing and VPNs

Broadband Network Optimization

End II End's patent-pending Broadband Network Optimization (BNO) is fundamentally different than the TCP acceleration methods described above (see Figure 8). BNO is a unique process that measures the characteristics of the broadband connection (in this case, the satellite link) and adjusts the parameters that control the session values for each protocol traveling across that particular broadband connection. This process enables every session to utilize the maximum available bandwidth of that broadband connection.

End II End's BNO process optimizes VPN tunnel data packets for the satellite link, which maximizes all data traffic passing through the VPN independent of application type or client-server data protocol. Since the optimization takes place in the protocol stack of the End II End device, which also serves as the VPN endpoint, there is no conflict or interference between the VPN function and BNO. All IP Security Protocols (IPSec) remain in place, unaltered and End II End's VPN runs in tunnel mode over satellite just like any other broadband connection. For more information about BNO, see End II End's free report, "3-Way Secure Optimization for Broadband Networks." End II End's BNO technology offers several distinct advantages over conventional forms of TCP acceleration.

- All traffic that passes through the VPN tunnel is optimized, regardless of protocol.
- BNO maximizes the throughput for all types of applications, including: Citrix, Lotus Notes, Peoplesoft, etc. not just high volume data sessions such as file transfers and video streaming.
- BNO allows the above enterprise applications to run a full service speed over satellite through an IPSec VPN tunnel.
- BNO technology is integrated into the End II End Gateway Security and Optimization solution, so that a single device delivers all the essential network services (Router, Firewall, VPN, QoS, Web Caching, and Intrusion Detection) required to connect a remote site to the headquarters. No additional TCP acceleration devices are required at the remotes sites, which reduces the number of devices that must be remotely installed, configured and managed, thus minimizing operational costs and simplifying the WAN topology.

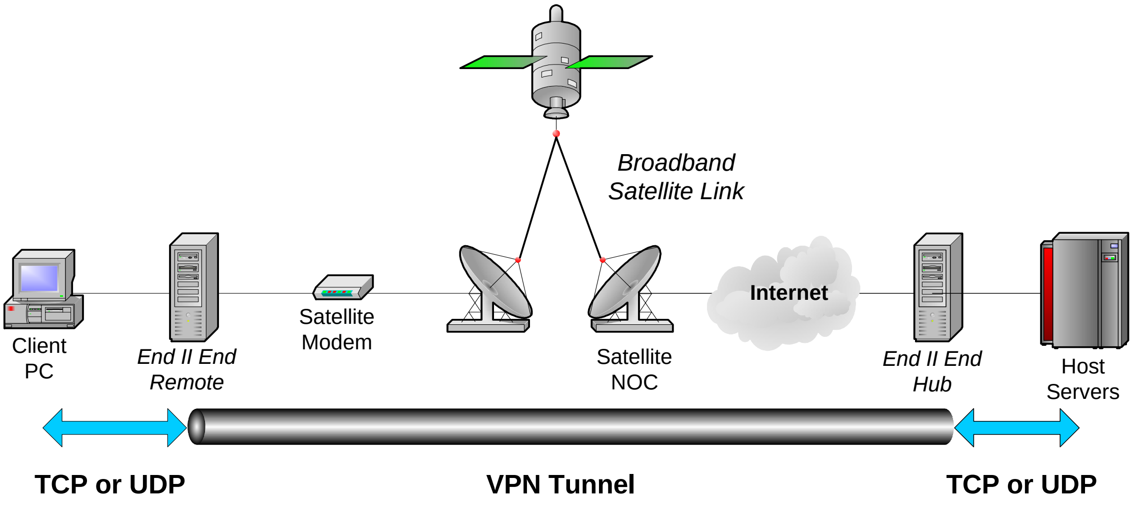


Figure 8. End II End's Optimized IPSec VPN Tunnel Solution

Conclusion

Both acceleration and optimization technologies help overcome satellite latency to varying degrees. Due to the nature of TCP transmissions, any performance increases achieved by the various acceleration methods are offset by serious compromises in data security. Fundamental incompatibilities arise when one attempts to operate conventional VPN devices over satellite links, due to the nature of TCP acceleration techniques commonly used for broadband satellite connections. VPNs may not function at all, security is compromised through the use of transport mode VPNs, and/or the lack of TCP acceleration can degrade application performance over satellite links so as to be unusable. Apart from this reduced security, most acceleration technologies enhance only certain types of applications (large file transfers, video). Commonly used enterprise software such as SAP, Siebel, Lotus Notes, Citrix, etc. do not benefit from acceleration.

End II End's patent-pending Broadband Network Optimization is vastly superior to the various satellite acceleration technologies. BNO maximizes the data throughput across satellite links, as well as any other broadband connection, enabling enterprise applications to run with maximum security (in IPSec tunnel mode with AES 256-bit encryption) and maximum performance (ALL data traffic runs, at full service speed, with a "LAN-Like" remote user experience). End II End's Gateway Security and Optimization Solution, including its powerful BNO technology, is the clear choice for any company concerned about performance and security in satellite networks.