Why not use the Standard Internet Suite for the Interplanetary Internet?

Robert C. Durst, Patrick D. Feighery, Keith L. Scott
The MITRE Corporation

Introduction

In this white paper, we consider the pros and cons of using the Standard Internet Suite of protocols to build the "Interplanetary Internet." In order to do this we must consider the communication environment that underlies the Interplanetary Internet, both in the near term and into the future. Once this environment is clearly identified, we can consider its effects on the protocols that make up the Internet Suite. In some of our discussion, we will use communication with the surface of Mars as an example. Readers should note that this is an example of convenience, and in no way indicates that the Interplanetary Internet is exclusively focused on providing support to Mars to the exclusion of other potential destinations, such as asteroids, the outer planets, etc.

Architectural Drivers

The following architectural drivers affect the applicability of Internet technology in deep space:

- Very long speed-of-light delays
- Episodic connectivity
- Asymmetric data rates
- Low signal to noise ratios

Speed-of-light Delays

The speed-of-light delay from Earth to Mars, for example, is approximately 4 minutes when Earth and Mars are at their closest approach. The one-way light time can exceed 20 minutes when Earth and Mars are in opposition. The speed-of-light delay to the outer planets becomes significantly higher. (Speed of light delay to Jupiter varies between approximately 30 and 45 minutes, to Saturn between 70 and 90 minutes). As we will see, delays in these ranges have a significant effect on the applicability of the standard Internet suite for use over interplanetary distances.

Episodic Connectivity

The term "episodic connectivity" refers to the ability to establish and maintain a continuous communication path between local and remote endpoints. In the Interplanetary Internet, there is no assurance that one of the "endpoints" of a communication will be on Earth. It is conceivable that an asteroid mission will communicate with a station on Mars, so subsequent discussions of the Deep Space Network are for example only.

Currently, the Deep Space Network (DSN) uses three primary earth stations with 70-meter antennas to support interplanetary missions. These three earth stations are positioned at 120-degree intervals around the earth (Goldstone, California, USA; Madrid, Spain; and Canberra, Australia) to ensure that the earth's rotation does not obscure visibility. If one could secure continuous use of the DSN 70-meter array, one could point continuously from the Earth at the surface of Mars. The rotational period of Mars is just above that of Earth, so it would be conceivable to have connectivity to a point on the surface of Mars for a period of up to 12 hours at a time. If a set of crosslinked satellites in Areostationary orbit (the Mars equivalent of Geostationary) were deployed, one could conceivably maintain full-time connectivity to a lander on the surface of Mars.

However, in the near term and in general cases, such connectivity will not be attainable. Orbital obscurations, in which communicating systems lose line-of-sight due to the positions of planetary bodies, are and will continue to be a significant source of intermittent connectivity within the Interplanetary Internet. Even if a relatively continuous communication system could be developed to support communication between Earth and Mars, other elements of the Interplanetary Internet would not be so richly connected. Consider an asteroid mission, in which a large build-up of infrastructure is not yet justifiable. In this and many similar cases, communication with a remote system may be obscured by the rotation of the asteroid or by the orbit of the spacecraft.

Practical considerations also conspire to impose intermittent connectivity. The Deep Space Network is perpetually oversubscribed. Missions must schedule time on the DSN, and these blocks of time do not in any way constitute full-time connectivity. It should be noted that the DSN, in general, does *not* operate as a broadcast medium, but rather as a point-to-point communications mechanism. The antennas must be carefully pointed toward and track the remote system, and the time required to repoint and recalibrate the antenna is considerable. (At the time of this writing, a one- to one-and-a-half hour calibration phase for a particular contact was typical for DSN.)

Note that for long round-trip-time missions, the duration of a DSN user session may not be as long as a single round trip time. Further, it is possible that a user may have a receive-only or a transmit-only session scheduled. This is another aspect of episodic connectivity that must be taken into account, especially if one considers use of the standard TCP/IP suite. The effects of episodic connectivity will be considered later in this paper.

Asymmetric Data Rates

The term "asymmetric data rates" means that a system may have a different data rate for outbound traffic than for inbound traffic. This is not normally the case in today's networks, although the Asymmetric Digital Subscriber Loop (ADSL) and DirecPC systems exhibit this characteristic to a limited degree. The DirecPC system exhibits data rate asymmetry up to approximately 15:1 (400000 bps receive, 28800 bps transmit, or 13.9:1), while ADSL exhibits data rate asymmetry up to approximately 100:1 (1.544 Mbps receive, 16000 bps transmit, or 96.5:1). Data rate asymmetry in spacecraft missions is typically on the order of 1000:1 or higher.

Signal to Noise Ratios

In a communications system, the transmitter sends symbols to the receiver, where each symbol corresponds to some number of information bits. The receiver's job is to decide, based on the received signal plus noise, what it thinks the transmitted symbols were in order to reconstruct the original information stream. In any system there is some noise that perturbs the transmitted signal before it reaches the receiver; this noise can cause the receiver to make errors in its decisions about the received symbols. A parameter that is a measure of how prone the receiver is to making errors is the ratio of the received signal power to the noise power at the receiver, or signal-to-noise ratio (SNR). The higher the SNR, the less likely it is that the receiver will make a mistake in decoding a received symbol. For a given data rate, coding, and modulation scheme, there is a mapping from received SNR to the error characteristics of the decoded information stream. When errors are relatively uncommon and widely spaced, it is appropriate to refer to the bit error rate (BER) of the system as the rate at which the receiver makes mistakes in the decoded bitstream. While fiber-optic systems can achieve bit error rates as low as 10^{-12} to 10^{-15}, deep space missions typically operate with uncoded bit error rates on the order of 10^{-1}. They use a concatenated code composed of a rate 7, constraint-length 1/2 inner convolutional code and a 223,255 Reed-Solomon outer code to bring the error rates down to the order of 10^{-9} or better.

Summary

Of these architectural drivers, only the light delays are considered to be immutable. However, for the foreseeable future, episodic connectivity will be present, at least in portions of the Interplanetary Internet. This is analogous to the build-up of internet capabilities in developing regions of the world today. Asymmetric data rates and low signal to noise ratios are also “facts of life” that will persist, although significant developments in power conversion may make these less of an issue. Although not stated previously, the cost of using the Deep Space Network is very large, and cost *is* an issue. As noted before, the DSN is a directional network, operating in an essentially point-to-point manner. The ability of a pair of communicating systems to fully-consume the capacity of the DSN is, therefore, an important consideration in the *cost-effectiveness* of its use.

A Hypothetical Internet Across Deep Space

The previous section described architectural characteristics of the Interplanetary Internet. This section considers the effects of these characteristics on the Internet protocols in common use today.

In considering the use of TCP over very long delays, we must be aware that TCP uses end-to-end signaling to maintain a consistent view of the state of the endpoints and of the network. If we assume that a TCP can be configured to operate over these very long delays (theoretically possible but, as we shall see later, practically difficult), we can avoid most of the endpoint state issues – the endpoints’ state evolves as a result of the data received from its peer endpoint. However, this does not take into account the fact that TCP is expected to respond to *network* state issues. That is, TCP must deal with changes in the network’s state. *This* expectation is

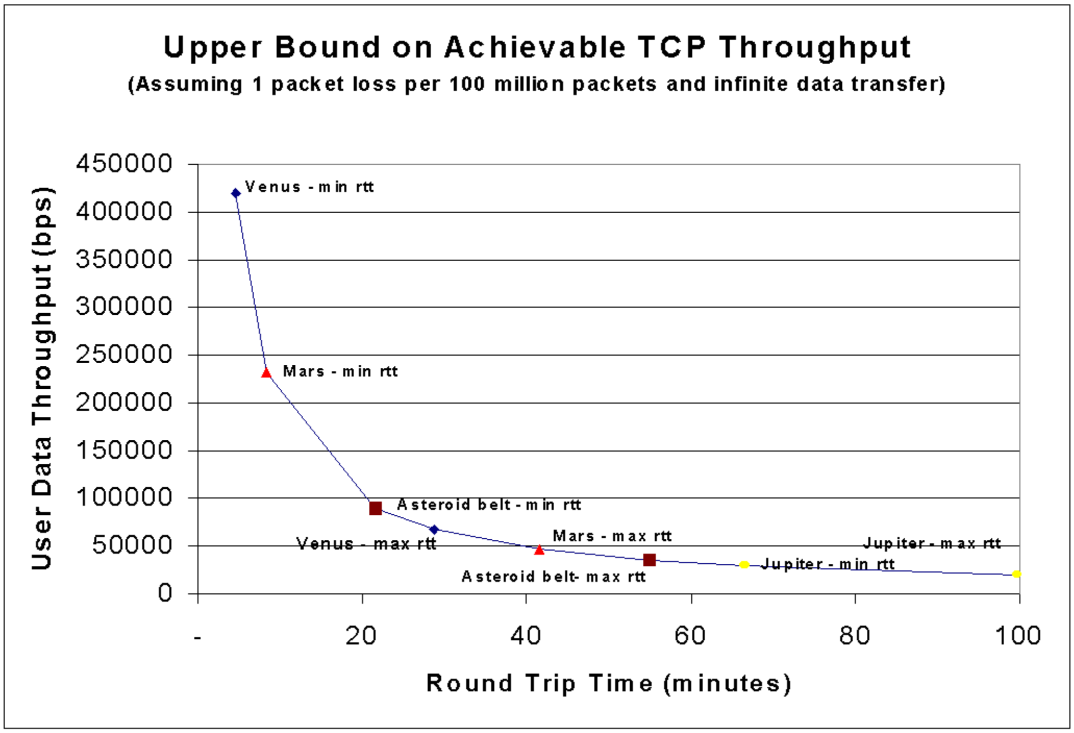

Figure 1. Upper Bound On TCP Throughput for Various Interplanetary Round Trip Times Assuming 1e-8 Packet Loss Rate

problematic in long-delay environments, since TCP uses end-to-end signaling for its control loops. The signal of an event (say, a congestion event) must propagate to the receiver (from some arbitrary point between the sender and the receiver), and then must be relayed to the sender for response. With TCP, the congestion signal is a packet loss, which, when detected by the receiver by a break in sequence numbering, is signaled back to the source via duplicate acknowledgments. The sender responds to this signal by cutting its rate in half, which is perceived by the congested point approximately one round trip time after the initial transmission of the signal (the packet loss).

Theoretical Results

We first use the equations generated by Mathis, et al [1] to characterize an upper bound on throughput that is *sustainable* as a result of TCP's congestion avoidance mechanisms. Figures 1 and 2 show the upper bound on achievable data rate based on these equations, applied to the relatively long light times associated with interplanetary communication. Figure 1 considers the case in which the probability of packet loss is exceedingly small (on average, 1 packet in 100 million is assumed to be lost). The source of loss is irrelevant: it may be congestion loss, corruption loss, or a loss in a station handover. Since the average length of a TCP connection is much less than 100 million packets, it must be noted that instantaneous throughput can exceed this upper bound, and may well. Figure 2 illustrates the upper bound on throughput assuming a

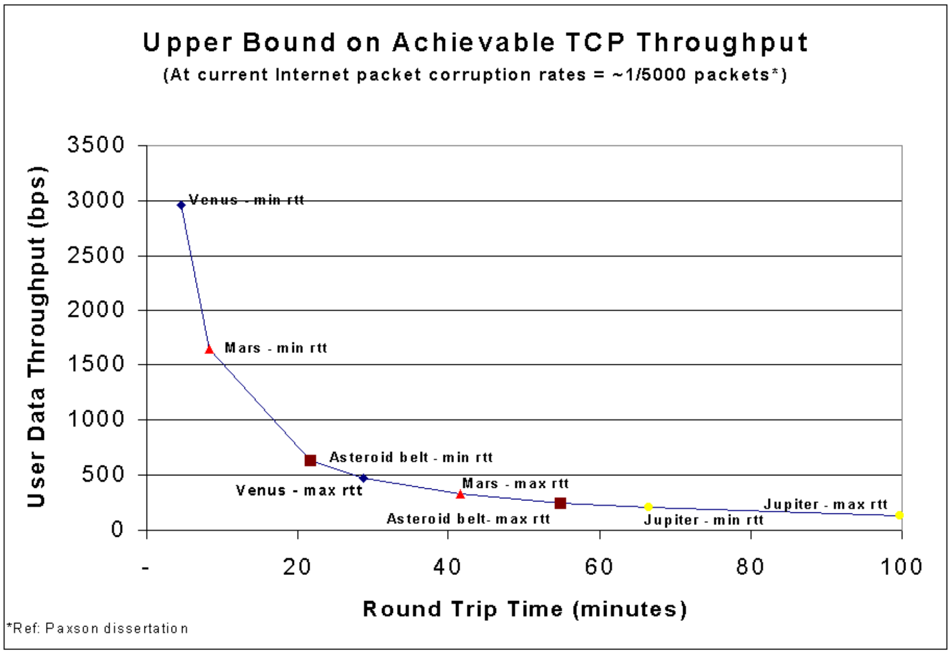

Figure 2. Upper Bound On TCP Throughput for Various Interplanetary Round Trip Times Assuming 2e-4 Packet Loss Rate

probability of packet loss of 1 in 5000 (this figure was reported by Paxson [2] as the packet *corruption* rate in the Internet). These figures show the bound placed by TCP's congestion avoidance algorithm on sustainable throughput. Note that since most connections are less than 100 million packets in duration, and in fact most are less than 5000 packets in duration, the steady-state results represented here will probably not dominate a TCP connection's performance. Rather, the start-up effects of TCP's slow start algorithm will have a much more dominant effect.

Figures 3 and 4 illustrate the effects of the TCP slow start algorithm on instantantaneous data rate and on channel utilization, respectively. The context for these figures is a 240-second one-way delay (somewhat lower than the Mars closest approach), and a 1 Mbps channel. Figure 3 shows that, after an hour of operation, the instantaneous data rate of the TCP connection is still significantly below 5 kbps. (Figures 3 and 4 disregard the delays of connection establishment.) The cause of the low data rate is the fact that TCP's slow start algorithm ramps up the data rate by increasing the rate by one additional packet per acknowledgment received. As the round trip time increases, the instantaneous data rate is adversely affected. Similarly, Figure 4 shows the channel utilization of a 1 Mbps channel. After approximately one hour of operation, the channel is less than .3% utilized. This utilization must be viewed in the context of the cost of available resources, which, for the Deep Space Network's 34-meter antenna array is over $2000/hour, and for the 70-meter antenna array (primary for Mars missions), over $6000/hour. To date, we know of no weekend discount plans available. In considering the time to move a particular volume of

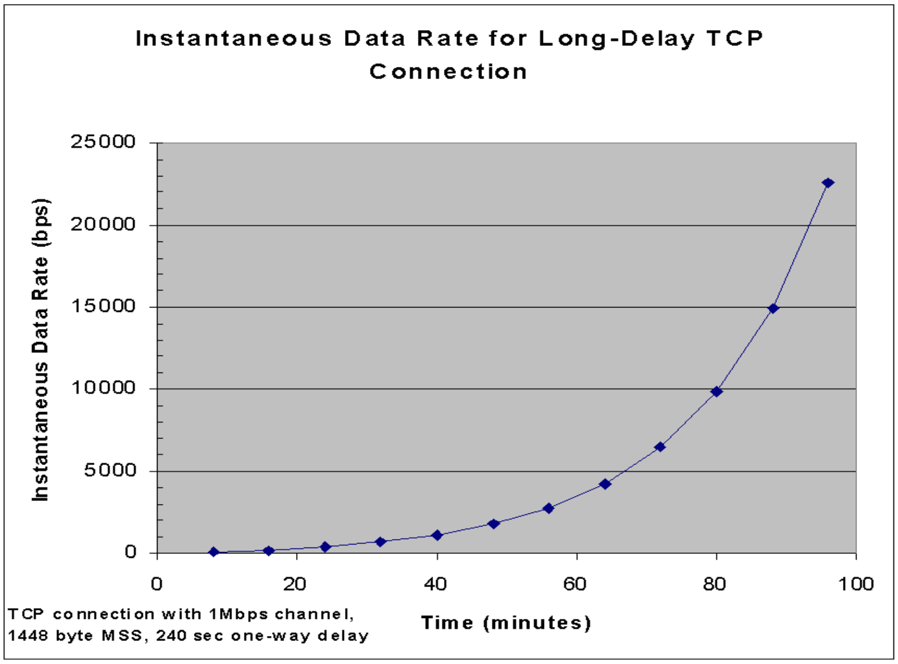

Figure 3. Instantaneous Data Rate for TCP Connection with Eight-Minute RTT)

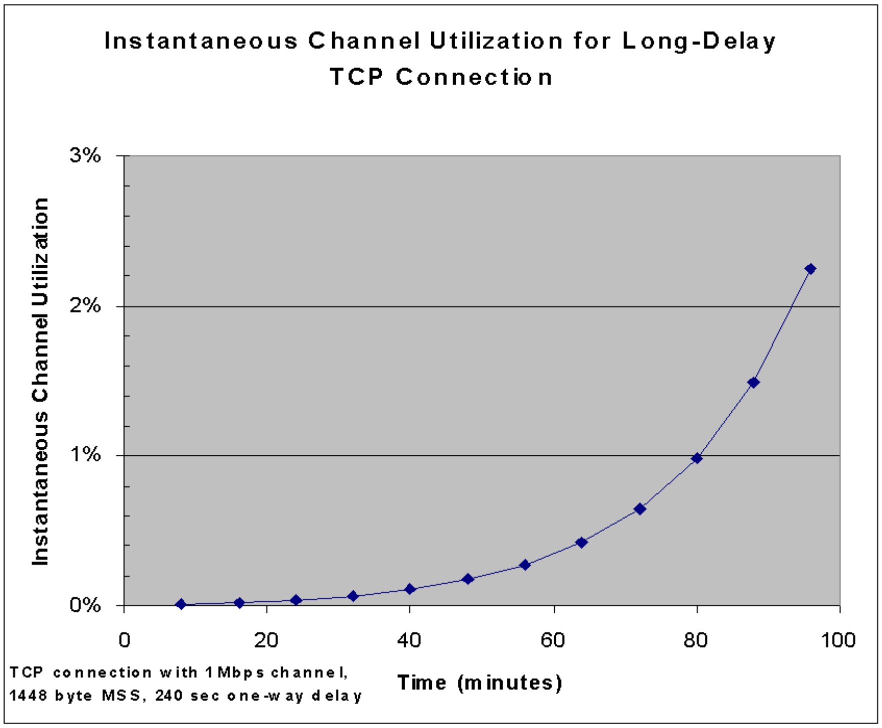

Figure 4. Instantaneous Channel Utilization for TCP Connection with Eight Minute RTT and 1 Mbps Channel

data, this example can transfer approximately 1M-byte of data in 90 minutes over a 1Mbps channel.

We did *not* test name-to-address binding in the course of our experiments, but use of the Domain Name System across long delays poses a number of significant problems. If an application on a remote planet wished to resolve an earth-based name to an address, it could use one of three alternatives in today's Internet: it could query an earth-resident name server; it could query a local secondary name server; or it could maintain a static list of host names and addresses. Querying an earth-resident name server results in a delay of a round trip time in the commencement of communication. This delay may be significant in terms of the available communication time, since delays are long and contact times are typically fairly short. Alternatively, one could maintain a secondary server locally, but zone updates would dominate communication channel utilization to the exclusion of other data. Finally, one could maintain a static list of host names, but this has the disadvantage of not scaling well as the system grows. An attractive alternative that remains to be fully explored is to use a facility similar to the Domain Name System's Mail Exchanger (MX) records, which specify an intermediate host to contact in case the named host is not reachable. This capability could be used to direct traffic to off-planet systems to a gateway system (all off-planet domains could be wild-carded), and is discussed in a subsequent document.

Experimental Results

The results in this section are based on tests run in a laboratory using a delay and bit-error-rate emulator with hosts running commercial implementations of the Internet protocols. We tested file transfer (via the File Transfer Protocol), and electronic mail (via the Simple Mail Transfer Protocol).

Due to practical limitations of the commercial protocols, some of our results are extrapolations of the laboratory data. These are clearly noted.

Test System Configuration

Figure 5 illustrates the test configuration used to gather the results presented in this section. The delay and error emulator queues packets based for a period of time calculated from a fixed propagation delay plus queueing delay based on the size of the queue and the service time for each packet (which is calculated from the outbound data rate and the queue length). A single Ethernet LAN segment hosts all of the workstations used in the testing, and the link emulator is configured for a data rate of 1000000 bits per second in each direction. Readers should recall that data rates are typically asymmetric, with ratios of 1000:1 being common. However, for these tests, we decided not to impose that restriction on the communication system, nor did we impose significant bit-error rates.

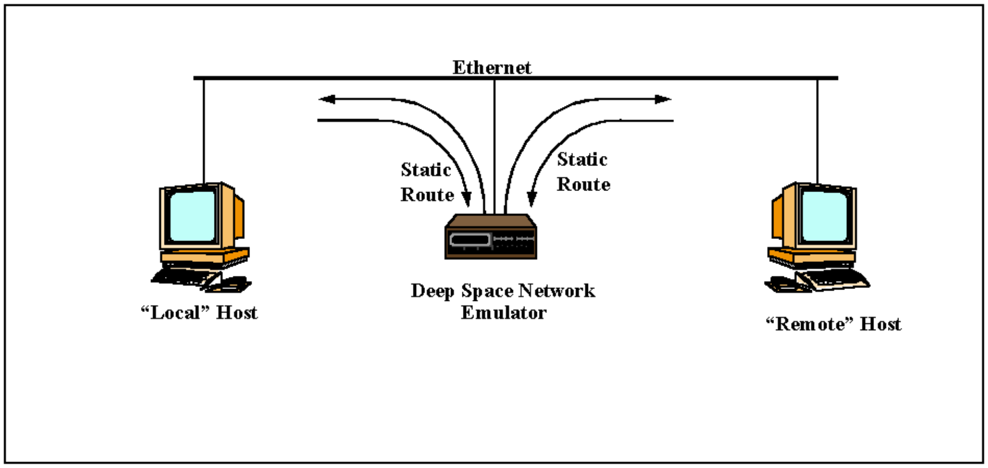

Figure 5. Laboratory Test Configuration for End-to-End Internet Testing

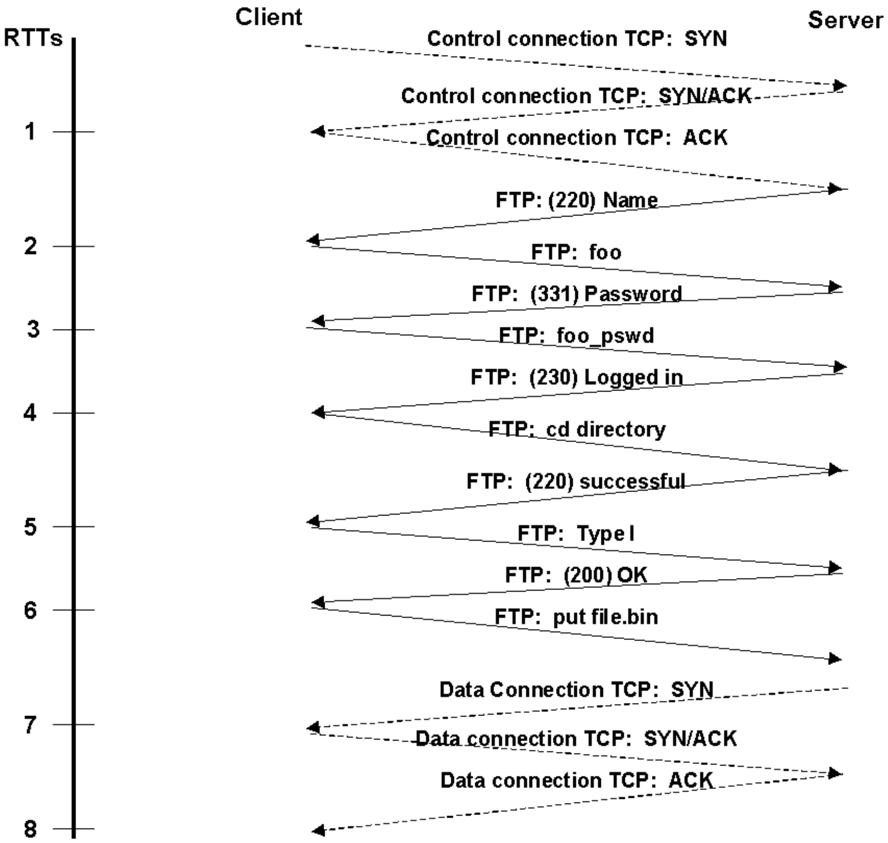

Figure 6. Typical Interaction Between FTP Client and Server

Example: File Transfer

Figure 6 illustrates the typical flow of an FTP control connection for putting or getting a file. Note that it takes, typically, eight round trips to get to a state where file data will begin to flow. While this may not seem like a great deal of time, consider the case in which each round trip time takes 8 minutes. This means that over an hour will elapse before the file transfer can actually begin. (Recall that round trip times to Mars are at *minimum* 8.5 minutes, and that they vary between approximately 8.5 minutes and 40 minutes, meaning that the same operation would take over 5 hours to initiate. Recall further that the realities of deep space communication are that the amount of Deep Space Network time that will be dedicated to any particular user is limited, and it is highly likely that a file transfer operation will not finish, or even be well started, by the time that the user's particular DSN session is over.)

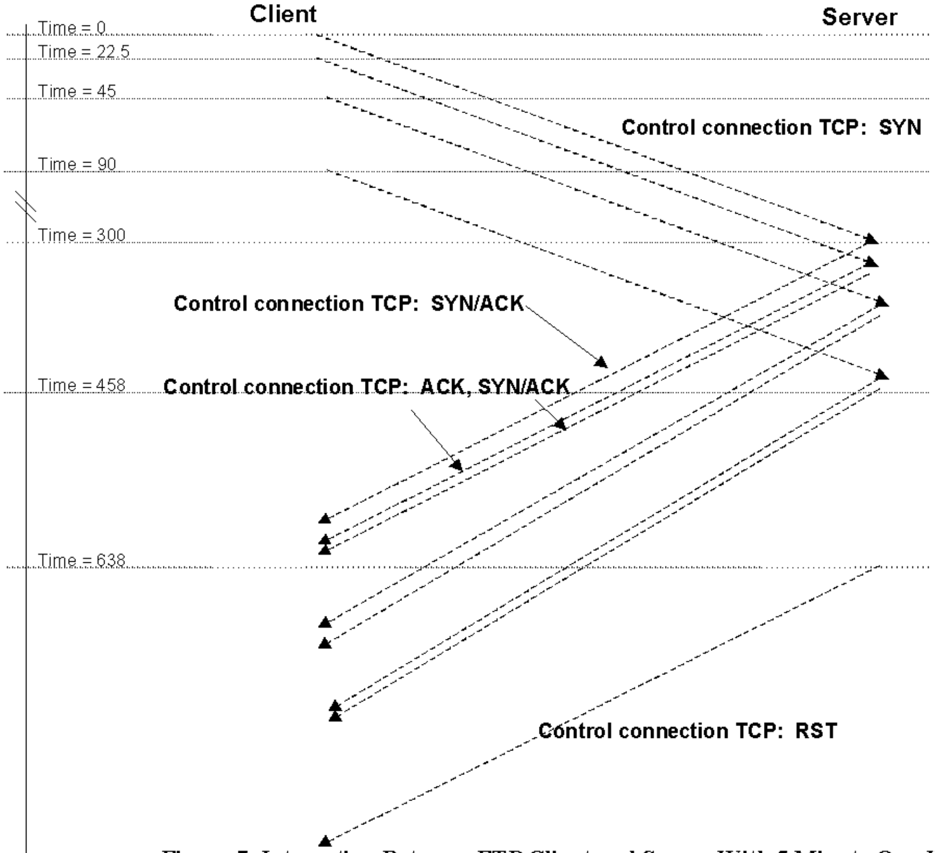
Client
Server
Time = 0
Time = 22.5
Time = 45
Time = 90
Time = 300
Time = 458
Time = 638
Control connection TCP: SYN
Control connection TCP: SYN/ACK
Control connection TCP: ACK, SYN/ACK
Control connection TCP: RST

Figure 7. Interaction Between FTP Client and Server With 5 Minute One-Way Delay, Minimum RTO = 20sec (max value settable in Solaris 2.7)

Figure 7 shows the actual packet exchanges between a pair of hosts attempting an FTP connection over a 5-minute one-way delay. The operating systems on both hosts were Solaris 2.7, and the minimum retransmission time out value for the TCP's in each host was configured to its maximum possible value (implementation defined and undocumented) of 20 seconds.

The following excerpt from RFC 1123 is intended to prevent FTP servers from waiting indefinitely on a client that has crashed. It recommends that FTP servers implement an idle

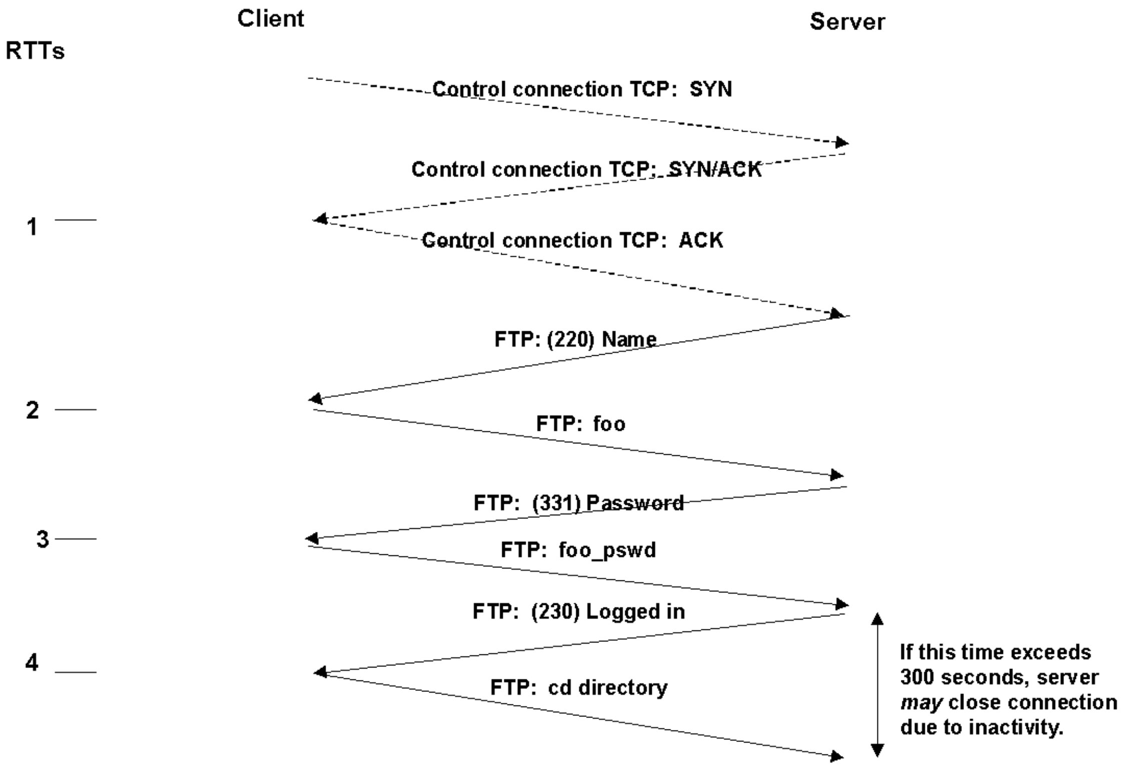

Figure 8. Effects of FTP Application Level Timers on Long Delay Connections

timeout, and recommends a minimum timeout value of greater than or equal to 5 minutes (in practice, many servers use a value of 900 seconds, or 15 minutes).

4.1.3.2 Idle Timeout

> A Server-FTP process SHOULD have an idle timeout, which will terminate the process and close the control connection if the server is inactive (i.e., no command or data transfer in progress) for a long period of time. The idle timeout time SHOULD be configurable, and the default should be at least 5 minutes.

The description of inactivity is important here: “no command or data transfer in progress.” Consider Figure 8. The transmission of the “230” message that concludes the third round trip terminates the command transfer from the standpoint of the server. The server has no way to query the underlying TCP state machine to determine that the message has not been acknowledged, and believes that the connection is idle. At this point, the “ball” is in the client’s court. It will take at least one round trip time from the time that the “230” message is sent until a client command can be received. If this time exceeds 5 minutes, the server may terminate the connection due to inactivity and be in complete conformance with the spirit of RFC 1123. Since

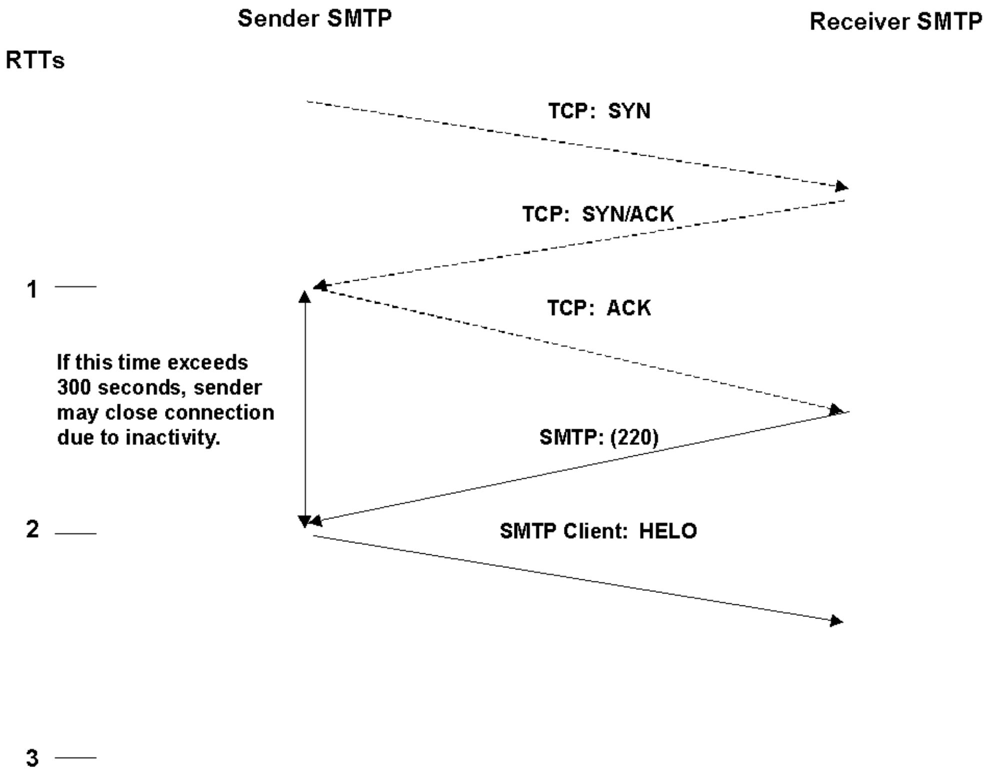

Figure 9. Effects of SMTP Application Level Timers on Long Delay Connections

not *all* FTP servers will (or should) be modified to support interplanetary idle times, this represents a problem in the end-to-end use of FTP.

In summary, FTP may not work at all in long delay environments, and if it does, it will be horribly inefficient in its use of very scarce resources.

Example: Electronic Mail Transfer

Of all of the existing Internet applications, the Simple Mail Transfer Protocol seems best suited for use in an interplanetary communication environment, due to its inherent store-and-forward nature. This store-and-forward behavior is perfect for operating in environments in which there is not continuous connectivity between all hosts. The SMTP model evolved to address this issue, albeit the causes for the discontinuous connectivity were different than those seen in the Interplanetary Internet. However, we see that while the electronic mail *model* may be well suited to interplanetary communication, SMTP as a *protocol* is unfortunately not.

Similar to FTP, RFC 1123 specifies timeouts for SMTP senders. The initial timeout is a minimum of five minutes, and as shown in Figure 9, if the round trip time is greater than this, the sender SMTP is free to consider that the receiver is unresponsive and terminate the connection. Further, receiver SMTPs *should* (according to RFC 1123) have timeouts that govern how long they wait for commands from the sender SMTP. As a result, both SMTP entities would have to be configured to be interplanetary-aware.

Also similar to FTP, there is a significant amount of back-and-forth interaction between the participants of an SMTP dialog. This style of interactivity is inappropriate for the very long delay environment of interplanetary communication, even though the email model is otherwise well suited.

Summary Application Behavior

We have considered two popular protocols, FTP and SMTP, that do not work well over interplanetary distances. These protocols do not work well in part due to the repeated use of request/response interactions to exchange information incrementally over the course of several round trip times. This style of interaction takes quite a long time when conducted with one-way propagation delays characteristic of interplanetary communication, and results in very poor use of the communication resources, which are typically dedicated during the course of the communication. Further, the presence of embedded timers poses another pervasive problem, since the minimum valid values of these timers is typically less than interplanetary round trip times. Finally, the name-to-address resolution issue remains problematic, requiring additional request/response interaction, the exchange of potentially huge databases of name-to-address information, or the maintenance of static, private name-to-address tables.

Could we *make* it work? Should we?

One could conceivably make TCP operate end-to-end over very long delays, merely by ensuring that timers can be min-max bracketed appropriately, etc. However, there are a number of problems with putting this suggestion into practice. First, the notion that one could arbitrarily connect to an off-planet host implies that the communication path will be shared with local (on-planet) traffic, at least some of which will be running TCP. It is a bad idea to mix widely divergent round trip times in a system that depends on round trip times to respond to network events. The unfairness seen by satellite TCP connections that share a bottleneck router with shorter-delay connections [3] is evidence that a very-long-delay connection will suffer.

The application protocols that we have examined are also ill suited to efficient operation over very long delays, resulting in inefficient use of the scarce communication resources (if they could be made to work at all).

We feel that, rather than attempting to coerce the existing Internet suite of protocols into operating poorly in an environment for which they were not intended, we should concentrate on using them in environments where they work well. These environments are the shorter-delay environments of the near-planet communication, such as between landers and orbiters and between communicating systems in an outpost of robots. To span the interplanetary distances, we feel that the timing-sensitive elements of the Internet suite are inappropriate. Further, we feel that attempting to maintain a single name-to-address binding space is futile. We explore these concepts more fully in subsequent documents.

References:

[1] Mathis, M. et al, "The Macroscopic Behavior of the Congestion Avoidance Algorithm",Computer Communications Review, volume 27, number 3, July 1997.

[2] Paxson, V., Section 13.3, “Measurements and Analysis of End-to-End Internet Dynamics,” Ph D Thesis, Computer Science Division, University of California, Berkeley, 1997.

[3] Lakshaman, T. V., and U. Madhow, “The Performance of TCP/IP for Networks With High Bandwidth-Delay Products and Random Loss,” IEEE/ACM Transactions on Networks, Volume 5, Number 3, June 1997, pp. 336-350.