Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computer Networks 47 (2005) 661–678
Computer Networks
www.elsevier.com/locate/comnet

On the performance of additive increase multiplicative decrease (AIMD) protocols in hybrid space-terrestrial networks ☆

Chunmei Liu, Eytan Modiano *

Massachusetts Institute of Technology, Cambridge, MA 02139, United States

Available online 18 September 2004

Abstract

It is well known that TCP does not perform well in networks with large propagation delay and high link loss probability, which are typical in hybrid space-terrestrial networks. Implementing Automatic-Repeat-Request (ARQ) retransmission over satellite links in hybrid networks has been proposed to enhance TCP performance by hiding the link losses from TCP. This paper analyzes the impact of implementing ARQ over satellite links on TCP performance. In particular, this paper focuses on the window flow control mechanism of TCP, and provides an exact model for a hybrid space-terrestrial system with transport layer Additive-Increase-Multiplicative-Decrease (AIMD) protocols and satellite link layer ARQ. Both Go-Back-N (GBN) and Selective-Repeat (SRP) are considered, and the delay of acknowledgements (ACKs) of ARQ are taken into account. The numerical solutions for the throughput as a function of different protocol and packet loss parameters are presented. Both the analysis and numerical results show that in most cases, implementing ARQ over satellite links can significantly improve TCP performance, and that proper choices of the protocol parameters can significantly improve the system performance as well. Although the paper focuses on hybrid space-terrestrial networks, the analysis and results are also applicable to other networks with large bandwidth-delay product and one error-prone bottleneck link.

Keywords: AIMD protocol; TCP; ARQ; Hybrid networks; Interactions

1. Introduction

Currently TCP is the dominate reliable transport layer protocol in data networks [20]. For a reliable transport layer protocol, generating and receiving acknowledgements (ACKs) by the transmission nodes is essential to guarantee reliable

☆ This work was supported by NASA Space Communication Project grant number NAG3-2835.
* Corresponding author.
E-mail addresses: mayliu@mit.edu (C. Liu), modiano@mit.edu (E. Modiano).

1389-1286/$ - see front matter
doi:10.1016/j.comnet.2004.08.002

transmissions. To achieve better throughput, it is also desirable for the end nodes to maximize the utilization of the available bandwidth provided by the network. The beauty of TCP is that it can simultaneously provide reliable transmission and explore the available bandwidth in the network without requiring any additional information beyond the necessary ACKs. Specifically, TCP makes use of the rate and order of ACKs to regulate its window size, thus its transmission rate, to perform congestion control and explore the available bandwidth [20]. It works very well in classical terrestrial networks, where the loss probability is relatively low and bandwidth-delay product is relatively small. Unfortunately, TCP does not perform well in space networks [2,17–19].

Compared with classical terrestrial networks, hybrid space-terrestrial networks have many properties that limit TCP performance, such as long propagation delays, asymmetric traffic and high error probability on satellite links. Many solutions are proposed to enhance TCP performance over hybrid networks as well, such as window scaling, time stamps, selective acknowledgements and link layer solutions. Comprehensive descriptions of the above solutions can be found in [1,4,13,19]. Among all properties of hybrid networks, the long propagation delay and high satellite link error probability (even after link layer Forward Error Correction in many cases) have the greatest impact on TCP performance. The main reason that these two properties deteriorate TCP performance is that TCP does not distinguish link losses from congestion losses. Satellite link losses in hybrid networks lead to TCP's timeouts and duplicate ACKs. TCP assumes that all losses are due to congestion, and reduces its window size accordingly, thus its transmission rate. Although it will recover its window size later, the long propagation delay of hybrid networks makes the recovery rate extremely low. If the link loss probability is very high, TCP will reduce its window size frequently, and the probability that TCP recovers its window size to a level compatible with the available bandwidth becomes very small. As a result, the utilization of the available link bandwidth becomes very low, and overall TCP performance is significantly deteriorated.

Several approaches exist that can deal with this incomparability between TCP and the long propagation delay and high link loss probability. These include: TCP spoofing, split-TCP, and link layer ARQ [3,13,19]. Among all these solutions, link layer ARQ has the advantage that it fits naturally into the layered structure of networks. The main idea of implementing ARQ is that it can "hide" the link losses from TCP. In this sense, implementation of ARQ can greatly reduce the number of unnecessary TCP window closings and significantly improve the overall TCP performance. However, the presence of an ARQ protocol can lead to other more subtle problems. In particular, due to the high bandwidth-delay product, ARQ retransmissions introduce high variability of the packet round trip time (RTT) seen by TCP, and lead to TCP timeouts due to ARQ retransmissions. We call these timeouts *false timeouts*.

Recently a number of papers have examined the TCP performance over link layer ARQ protocols. Most of them involve simulations in wireless environment [3,10,11,15,23]. There is some analytical work on the performance of TCP over ARQ as well [6–9,16]. These papers provide approximate analysis of TCP performance over a link layer protocol with the assumption of instantaneous ACK feedback. The large propagation delay of space links makes these models unsuitable for hybrid networks. The authors in [6] further assume independence between window size and RTT of each window, which is not suitable for hybrid networks either. In addition, there are many papers on the analysis of ARQ as well. In [21] queuing models are developed for the Go-Back-N (GBN) protocol, and [14] provides queuing models for both the GBN and Selective-Repeat (SRP) protocols. Generally queuing models are used as a tool to analyze the ARQ protocols for different channels and network structures [24,26].

This paper studies the performance of TCP over link layer ARQ in the context of hybrid space-terrestrial networks with satellite links, examines whether implementing ARQ is beneficial, and explores the influence of protocol and loss parameters on the overall system performance. The analysis and results are applicable to other networks with large bandwidth-delay product and one error-prone bottleneck link as well. The paper is different from earlier works in several ways. First, this paper studies a hybrid space-terrestrial network where the long propagation delays and high satellite link error probability is essential to the system performance. Second, this paper focuses on TCP's window flow control mechanisms and deliberately disregards other aspects of TCP, such as RTT measurements and estimation, the detail retransmission mechanism, and timer granularity. Since the window size evolution is the main factor that affects TCP throughput, we believe that for the purpose of investigating TCP throughput, the protocols considered capture the essence of TCP. Third, the ARQ protocols considered include both GBN and SRP protocols. The delay of the ACK signals is also taken into account, which is significant for satellite links. Furthermore, this paper gives an exact analysis of our system instead of approximations and simulations, thus providing an analytical framework for future joint study of TCP and ARQ in hybrid space-terrestrial networks.

Specifically, the system investigated consists of two end nodes communicating over a hybrid network that includes a satellite link and some other space and terrestrial links. The satellite link is the bottleneck link as well as the error prone link, which is typical in such networks. The sender has an unlimited number of packets to be sent to the receiver, and the performance metric considered is the throughput. The losses incurred by packets include the link losses of the error-prone satellite link and other random losses, for example, losses of other links and congestion losses. The error-prone satellite link implements a GBN or SRP retransmission mechanism to recover the satellite link losses. The transport layer of the end nodes implements a variation of the Additive-Increase-Multiplicative-Decrease (AIMD) protocol that is similar to TCP window flow control protocols, which will be described in detail later.

This paper models the system as a finite state Markov chain with reward functions. The transition probabilities and the reward functions are expressed in simple window-based product form and sum form, respectively. Moreover, queuing models for GBN and SRP are also developed. These models are used to obtain the probabilities needed for solving the transition probabilities and the reward functions of the Markov chain, and the throughput of the system is derived by the theory of Markov chains with reward functions. The numerical results show that in most cases implementing ARQ over the satellite link can achieve significant improvement in system throughput. Moreover, we show that by proper choice of protocols parameters, such as the packet size and the number of transmission attempts per packet at the satellite link ARQ, significant performance improvement can be obtained.

The paper is organized as follows: Section 2 describes in detail the system under consideration. Sections 3 and 5 model different systems considered, and Section 4 provides the simple queuing models for the ARQ protocols. Finally, Section 6 discusses the numerical results for different protocol and packet loss parameters, and Section 7 concludes the paper.

2. System description

The system we consider consists of two end nodes communicating over a hybrid space-terrestrial network with one satellite link and some other space and terrestrial links, as shown in Fig. 1. The sender has unlimited packets to be transmitted to the receiver, and these packets have fixed length. The transport layers (TL) at the end nodes implement a variation of AIMD protocol, which is similar to TCP window mechanisms and will be described in detail later. The link layer over the satellite link, called satellite link layer (SLL), implements an ARQ protocol, where both GBN and SRP are considered. The other link layers (LLs) do not employ ARQ.

The satellite link in the system is the bottleneck as well as the error-prone link. As mentioned before, this is typical in hybrid space-terrestrial networks. The time for the satellite link to transmit one packet is defined to be one time unit. In this way, the time is divided into time slots. The round trip delay of one transmission over the satellite link, defined to be the interval between the time the satellite link sender sends out a packet and the time the

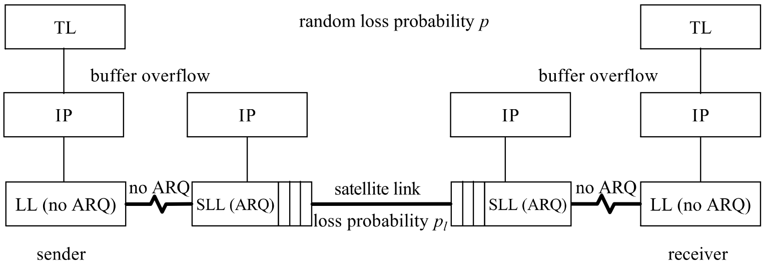

Fig. 1. System with two nodes communicating over a hybrid space-terrestrial network.

sender receives the acknowledgment of this transmission, is fixed to be d time slots. The remaining time needed for the packets to go through the network is assumed to be negligible, a reasonable assumption in hybrid space-terrestrial networks.

Packets in the system incur two types of losses. One type refers to losses over the satellite link. For brevity, we call those *link losses*. The other type refers to all the losses other than the satellite link losses, such as congestion losses and link losses at the other space and terrestrial links that do not employ ARQ. We call those *random losses*. Note that the link losses can be recovered by the employed SLL ARQ, and the random losses can only be recovered by the end-to-end transport layer retransmissions.

Packets incur losses independently of each other. Each packet incurs a random loss with probability p, and each transmission over the satellite link incurs a link loss with probability p_l. The probability that a packet incurs no random loss and a transmission over the satellite link incurs no link loss, denoted by q and q_l, respectively, are thus $q = 1 - p$ and $q_l = 1 - p_l$. All acknowledgements (ACKs), including both the transport layer ACKs and the ACKs of ARQ, are assumed to be loss free. Notice that by proper choice of the above loss probabilities, one can also model systems employing no SLL ARQ.

The TL has two ways to detect losses, the *timeout signal* and the *random loss signal*. When the age of an unacknowledged packet exceeds the TL timeout value, denoted by TO, a TL timeout occurs. We also assume that any random loss can be detected by the node where the loss happens after a fixed time interval t_d. After detecting the loss, the node will generate a signal indicating this loss and send it via the same path as that of a packet to the receiver. We call this signal the random loss signal. The timeout signal is essential for reliable TL transmissions [25], and the random loss signal is used to approximate the duplicate ACKs used in TCP and allows us to examine the fast retransmit and fast recovery mechanism.

The AIMD protocol employed at the TL works as follows. The TL sends packets in batches. The size of the batches is the current window size of the TL. Once the TL receives all the acknowledgments of its previous window of packets, it sends out the next window of packets. After either a timeout signal or a random loss signal is received, the TL changes the window size according to its window-update algorithm, and restarts the transmission from the packet that incurs the loss signal.

We consider two window-update-algorithms, the Tahoe window update algorithm and the Reno window update algorithm [20]; corresponding to TCP Reno and Tahoe, respectively. Both algorithms have two parameters: current window size W and a threshold W_t. In addition, there is a maximum window size W_M which the window size can never exceed and which resembles the maximum window size in TCP advertised by the receiver. The Tahoe algorithm works as follows:

- The initial window size is 1.
- After receiving all the acknowledgments of the last window:
  - If $W < W_t$, the window size is doubled;
  - If $W \geqslant W_t$, the window size is increased by 1.

If the resulting window size is larger than W_M, set the window size to be W_M; otherwise, keep it.

- Upon a timeout signal, the window size is set to be 1, and W_t is set to be half of the window size when the timeout occurs.

The Reno algorithm consists of all the above rules that the Tahoe algorithm has and the following additional rule:

- Upon a random loss signal, W_t is set to be half of the window size when the loss signal is received, and the window size is set to be W_t.

One can see that both algorithms are variations of Additive-Increase-Multiplicative-Decrease (AIMD) algorithms. They are similar to the TCP Tahoe and Reno window-update algorithms, respectively. The two phases, $W < W_t$ and $W \geqslant W_t$, correspond to the slow start phase (SS phase) and the congestion avoidance phase (CA phase) of TCP, respectively. For convenience, we also call these two phases the SS phase and the CA phase. The maximum window size W_M corresponds to the receiver advertised window size. The differences between the two algorithms and the TCP Tahoe and Reno lie in two aspects. One is that in TCP the window size is updated upon receiving each acknowledgment, while here the window size is updated when a batch of acknowledgments is received. Nevertheless, upon receiving a loss signal, the rates of window update are the same for TCP and our corresponding algorithms. The other difference is that in the TCP Reno fast retransmit and fast recovery is triggered by duplicate ACKs, while in our Reno window update algorithm fast retransmit and fast recovery is triggered by random loss signals. The random loss signals are used to simplify the analysis of the protocols. Note that our goal is to explore whether or not implementing ARQ at the error prone satellite link will improve the TCP performance, and the key factor is the window size. We therefore believe that our model is a reasonable model for this purpose and can give us useful insights into the system performance.

The satellite link layer employs the standard GBN or SRP [5]. The GBN or SRP window size is no less than the round trip delay d, so that the satellite link capacity can be fully utilized and the actual transmission rate over it is limited only by the TL window size. Furthermore, negative acknowledgement (NAK) signals are assumed to be used. The order of packet transmissions follows the standard GBN or SRP rules and packets are delivered to the corresponding higher layer in order [5].

From the above descriptions we can see that the system considered can also model other networks with large bandwidth-delay product and one error-prone bottleneck link. The following analysis and results are therefore applicable to such networks as well.

In the next two sections, we will first model the system with the Reno algorithm and ARQ (both GBN and SRP), and then derive the system throughput as a function of the protocol and loss parameters. This model can be easily extended to systems with the Tahoe algorithm and systems with some variations of ARQ, which will be shown in Section 5.

3. Modelling the system with Reno algorithm

3.1. System Markov model and throughput

Consider the system behavior after the TL receives a loss signal. Let W^e denote the window size upon the loss signal and S indicate the type of loss signal received, with $S = TO$ referring to a timeout signal and $S = RL$ referring to a random loss signal. Let W and W_t denote the window size and threshold right after the loss signal. Then, according to the Reno window update algorithm, $W_t = W^e/2$ and if $S = TO$, $W = 1$ or if $S = RL$, $W = W^e/2$. This means that the pair (W^e, S) uniquely determines W and W_t. Moreover, by the algorithm, W and W_t together are the state of the system, that is, given W and W_t, the future behavior of the system is completely determined and is independent of its past. Therefore, the pair (W^e, S) is also a state of the system. The system can thus be modelled by the following Markov chain: the states are the (W^e, S) pairs, and the transitions take place when the HL receives a loss signal. Furthermore, since the window size will never exceed the maximum value W_M, the system chain is a finite-state Markov chain.

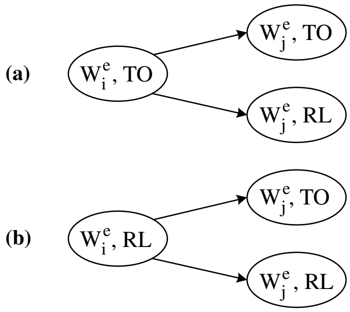

Fig. 2. Typical transitions in Markov chain for system with Reno algorithm.

Fig. 2(a) and (b) show the transitions from a typical state i with $S_i = TO$ and $S_i = RL$, respectively. Here the subscript i denotes the state. Later in the next subsection, we will explore in detail the system behavior during one transition and derive the transition probabilities.

In order to obtain the throughput of the system, for each transition of the Markov chain, we further define the following two reward functions:

- V_{ij}^n: the expected number of successfully transmitted packets during the transition from state i to state j.
- V_{ij}^t: the expected time taken for the transition from state i to state j.

The corresponding reward functions associated with state i, denoted by v_i^n and v_i^t, respectively, are thus

$$v_i^n = \sum_j V_{ij}^n P_{ij}^T \quad \text{and} \quad v_i^t = \sum_j V_{ij}^t P_{ij}^T, \tag{1}$$

where P_{ij}^T is the transition probability from state i to state j. Here the superscript T denotes transition. Let π_i be the steady state distribution of the Markov chain. The steady state expected rewards per transition are therefore given by [12],

$$v^n = \sum_i \pi_i v_i^n \quad \text{and} \quad v^t = \sum_i \pi_i v_i^t. \tag{2}$$

Theorem 1. *The throughput of the system described is* $\lambda = v^n/v^t$.

Proof. Starting from time 0, let $N(t)$ and $M(t)$ be the number of successfully transmitted packets and the number of transitions of the Markov chain up to and including time t, respectively. Let N_i be the number of successfully transmitted packets during the ith transition, and T_i be the time taken by the ith transition. Moreover, let t_0 be the epoch of the first transition. Then

$$N(t_0) + \sum_{i=2}^{M(t)} N_i \leqslant N(t) \leqslant N(t_0) + \sum_{i=2}^{M(t)+1} N_i,$$

$$t_0 + \sum_{i=2}^{M(t)} T_i \leqslant t \leqslant t_0 + \sum_{i=2}^{M(t)+1} T_i.$$

Therefore, $N(t)/t$ is upper bounded by $\left(N(t_0) + \sum_{i=2}^{M(t)+1} N_i\right) \Big/ \left(t_0 + \sum_{i=2}^{M(t)} T_i\right)$ and lower bounded by $\left(N(t_0) + \sum_{i=2}^{M(t)} N_i\right) \Big/ \left(t_0 + \sum_{i=2}^{M(t)+1} T_i\right)$. For the upper bound, we have

$$\begin{aligned}
\lim_{t\to\infty} \frac{N(t_0) + \sum_{i=2}^{M(t)+1} N_i}{t_0 + \sum_{i=2}^{M(t)} T_i} &= \lim_{M(t)\to\infty} \frac{N(t_0) + \sum_{i=2}^{M(t)+1} N_i}{t_0 + \sum_{i=2}^{M(t)} T_i} \\
&= \lim_{M(t)\to\infty} \frac{\left(N(t_0) + \sum_{i=2}^{M(t)+1} N_i\right) \Big/ M(t)}{\left(t_0 + \sum_{i=2}^{M(t)} T_i\right) \Big/ (M(t)-1)} \\
&= \lim_{M(t)\to\infty} \frac{\sum_{i=2}^{M(t)+1} N_i / M(t)}{\sum_{i=2}^{M(t)} T_i / (M(t)-1)} \\
&= \frac{v^n}{v^t}.
\end{aligned}$$

Similarly, the limit of the lower bound as $t \to \infty$ can be shown to be v^n/v^t as well. Thus, the throughput of the system is

$$\lambda = \lim_{t\to\infty} \frac{N(t)}{t} = \frac{v^n}{v^t}. \qquad \square \tag{3}$$

In the next subsection, we derive the transition probabilities of the Markov chain as well as the reward functions for each transition. Then the throughput of the system can be obtained using Eqs. (1)–(3).

3.2. Transition probabilities and reward functions

First consider the system behavior during the transition from an arbitrary state i with $(W_i^e, S_i = TO)$ to state j. By the definition of states and the algorithm, at the beginning of the transition, $W = 1$ and $W_t = W_i^e/2$. The window size then evolves as $1, 2, 4, \ldots, W_c, W_c + 1, \ldots, W_M$, where W_c is the least integer that is a power of two and greater than W_t. Denote this set of window sizes by L_i and the kth element by w_{ik}. Notice that L_i is a function of state i only. Obviously, any possible end state j must have $W_j^e \in L_i$. Let N_{ij} denote the integer such that $w_{iN_{ij}} = W_j^e$. Then by the definition of the system, we know exactly how the system behaves during the transition for state i to state j as follows: the TL sends out windows of packets with size $w_{i1}, w_{i2}, \ldots, w_{iN_{ij}}$. All packets in the previous $N_{ij} - 1$ windows were successfully received, and at least one packet in the last window (the N_{ij}th window) incurs a loss. The first loss detected within the last window is of type S_j.

For state i with $S_i = RL$, the system behaves similarly during the transition from state i to state j. The only difference is that now the set $L_i = \{W_t, W_t + 1, \ldots, W_M\}$.

The system behavior described above shows that the system can be analyzed based on windows. Specifically, since each window incurs a timeout or a random loss independently, the transition probability from state i to state j, P_{ij}^T, becomes the product of probabilities of no loss on the first $N_{ij} - 1$ windows multiplied by the probability of loss S_j on the last window. For the same reason, the probabilities that no loss signals are received for different windows with same size w are the same. Denote it by Q_w^W, where the subscript w is the window size and the superscript W denotes that the quantity is related to a window. Similarly, the probabilities that a timeout signal or a random loss signal is received for different windows with same size w are the same as well. Denote them by $P_w^{W,TO}$ and $P_w^{W,RL}$, respectively. Then, the transition probability can be expressed as

$$P_{ij}^T = \begin{cases} 0 & \text{if } W_j^e \notin L_i, \\ \left(\prod_{n=1}^{N_{ij}-1} Q_{w_{in}}^W\right) P_{w_{iN_{ij}}}^{W,TO} & \text{if } W_j^e \in L_i \\ & \quad \text{and } S_j = TO, \\ \left(\prod_{n=1}^{N_{ij}-1} Q_{w_{in}}^W\right) P_{w_{iN_{ij}}}^{W,RL} & \text{if } W_j^e \in L_i \\ & \quad \text{and } S_j = RL. \end{cases} \tag{4}$$

The reward function V_{ij}^n can be obtained in a similar way as follows. From the system behavior presented previously, the number of successfully transmitted packets during the transition from state i to state j is the sum of the first $N_{ij} - 1$ windows of packets plus the number of successfully transmitted packets of the last window (the N_{ij}th window), that is, $\sum_{n=1}^{N_{ij}-1} w_{in} + k - 1$, where k is the integer such that the first $k - 1$ packets in the N_{ij}th window were successfully received and the kth packet incurs the loss of type S_j. Note that k is a random variable.

Let $P_k^{P,TO}$ and $P_k^{P,RL}$ denote the probabilities that a timeout signal and a random loss signal on the kth packet of a window is received, respectively. Here the superscript P indicates that the quantity is related to packets. Since within the same window, packets incur losses independent of later packets, these two quantities are independent of the window size, given that the window size is larger than or equals to k. Then, by straightforward derivation, the expected reward V_{ij}^n can be shown to have the following window-based sum form:

$$V_{ij}^n = \begin{cases} 0 & \text{if } W_j^e \notin L_i, \\ \sum_{n=1}^{N_{ij}-1} w_{in} + \frac{\sum_{k=1}^{w_{iN_{ij}}} (k-1) P_k^{P,TO}}{P_{w_{iN_{ij}}}^{W,TO}} & \text{if } W_j^e \in L_i \\ & \quad \text{and } S_j = TO, \\ \sum_{n=1}^{N_{ij}-1} w_{in} + \frac{\sum_{k=1}^{w_{iN_{ij}}} (k-1) P_k^{P,RL}}{P_{w_{iN_{ij}}}^{W,RL}} & \text{if } W_j^e \in L_i \\ & \quad \text{and } S_j = RL. \end{cases} \tag{5}$$

Similarly, the reward function V_{ij}^t is the sum of the expected time taken by the N_{ij} windows. Let $\overline{T_w^W}$

denote the expected time taken by a window with size w and no loss, and $\overline{T_w^{W,RL}}$ denote the expected time taken by a window with size w and a random loss signal. Notice that the expected time taken by a window with a timeout signal is fixed to be TO. Then V_{ij}^t has the following window-based sum form:

$$V_{ij}^t = \begin{cases} 0 & \text{if } W_j^e \notin L_i, \\ \sum_{n=1}^{N_{ij}-1} \overline{T_{w_{in}}^W} + TO & \text{if } W_j^e \in L_i \\ & \quad \text{and} \quad S_j = TO, \\ \sum_{n=1}^{N_{ij}-1} \overline{T_{w_{in}}^W} + \overline{T_{w_{iN_{ij}}}^{W,RL}} & \text{if } W_j^e \in L_i \\ & \quad \text{and} \quad S_j = RL. \end{cases} \tag{6}$$

Eqs. (4)–(6) express the transition probabilities and reward functions in terms of five unknown probabilities Q_w^W, $P_k^{P,TO}$ and $P_k^{P,RL}$, $P_w^{W,TO}$ and $P_w^{W,RL}$, as well as two expected time $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$. The next subsection further examines the relationship between different probabilities and reduces the number of unknown probabilities to two.

3.3. Relationship between probabilities

As mentioned before, the possible long time taken by the retransmissions of the SLL ARQ due to the satellite link transmission errors can cause the TL timeout. Since the TL timeout signal is designed to recover the losses that cannot be recovered by the SLL, we call the TL timeout caused by the SLL retransmissions the false timeout, and let $Q_k^{P,F}$ denote the probability of no false timeout on the first k packets of one window, given no random losses on the first k packets. Here the superscript P indicates that the quantity is related to packets, and F means false timeout.

Let Q_k^P be the probability that no loss signal is received on the first k packets of a window with size larger than or equals to k, and P_k^P be the probability that a loss signal on the kth packet of a window is received. Here again, the superscript P means that the quantity is related to packets. Since each packet incurs a random loss independent of each other and independent of the SLL retransmissions, we have

$$Q_k^P = Q_k^{P,F} q^k. \tag{7}$$

By definitions of the following probabilities and noting that the events of receiving a timeout signal and a random loss signal are exclusive, we have

$$Q_w^W = Q_w^P, \tag{8a}$$

$$P_k^P = Q_{k-1}^P - Q_k^P \quad \text{with } Q_0^P = 1, \tag{8b}$$

$$P_k^P = P_k^{P,TO} + P_k^{P,RL}, \tag{8c}$$

$$P_w^{W,TO} = \sum_{k=1}^{w} P_k^{P,TO}, \tag{8d}$$

$$P_w^{W,RL} = \sum_{k=1}^{w} P_k^{P,RL}. \tag{8e}$$

The above equations show that once we have $Q_k^{P,F}$ and $P_k^{P,RL}$, we can obtain the five probabilities needed, Q_w^W, $P_k^{P,TO}$ and $P_k^{P,RL}$, $P_w^{W,TO}$ and $P_w^{W,RL}$. Specifically, Equation (7) gives Q_k^P from $Q_k^{P,F}$. Then Q_w^W can be obtained from (8a). Eq. (8b) then gives P_k^P. Together with the given $P_k^{P,RL}$, $P_k^{P,TO}$ can be obtained from (8c). Finally, $P_w^{W,TO}$ and $P_w^{W,RL}$ can be obtained from (8d) and (8e). From these five probabilities, together with the quantities $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$, we can solve the transition probabilities and the reward functions. The next section derives the LL queuing models for both GBN and SRP and gives us these two probabilities, $Q_k^{P,F}$ and $P_k^{P,RL}$, and the two expected times $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$.

4. Link layer ARQ queuing models

Section 3 shows that for the system with the Reno mechanism, the throughput of the system can be obtained once we know the four quantities $Q_k^{P,F}$, $P_k^{P,RL}$, $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$. This section develops simple queuing models for the SLL ARQ protocols with batch arrivals and derives these four quantities.

4.1. Queuing model for the GBN

This subsection first considers the distribution of packet round trip times and describes the GBN queuing model. Then, based on the queuing model, the four quantities required are derived.

Let $RTTH_k$ denote the TL round trip time of the kth packet given that the kth packet incurs no random loss, and $RTTL_k$ denote its SLL round trip time. Since in GBN the packets are acknowledged in order at SLL and delivered to the corresponding higher layer in order as well, we have

$$RTTH_k = RTTL_k, \tag{9a}$$

$$RTTH_l \leqslant RTTH_k \text{ for all } l \leqslant k. \tag{9b}$$

Moreover, the GBN protocol in our model can be modelled as a queuing system with independent geometrically distributed service time [5] with mean p_l/q_l. Since packets arrive at the SLL in batches and are acknowledged in order, the SLL round trip time of the kth packet $RTTL_k$ is thus the sum of its service time and the service times of the $k-1$ previous packets within the same window. Therefore, $RTTL_k$, as well as $RTTH_k$, is binomially distributed as follows:

$$\begin{aligned}
&\Pr(RTTH_k = md + k + d - 1)\\
&\quad = \Pr(RTTL_k = md + k + d - 1)\\
&\quad = \binom{m+k-1}{m} p_l^m q_l^k,
\end{aligned} \tag{10}$$

where m is the total number of retransmissions of the first k packets in the window. Note that extra $d-1$ slots are added for allowing the GBN ACK to come back to the SLL sender after the packet leaves the SLL sender's queue.

Next we derive the four quantities based on the distribution of $RTTH_k$ given in Eq. (10). First consider $Q_k^{P,F}$. By the definition of $Q_k^{P,F}$, together with Eq. (9b) and the distribution of $RTTH_k$ in (10), $Q_k^{P,F}$ is given by

$$\begin{aligned}
Q_k^{P,F} &\triangleq \Pr\left(\bigcap_{l=1}^{k} RTTH_l < TO\right)\\
&= \Pr(RTTH_k < TO)\\
&= \sum_{m=0}^{M_k^{TO}} \binom{m+k-1}{m} p_l^m q_l^k,
\end{aligned} \tag{11}$$

where $M_k^{TO} \triangleq \lfloor (TO - k - d)/d \rfloor$. Notice that the physical meaning of M_k^{TO} is the maximum number of retransmissions for the first k packets without causing a false timeout.

Now consider $P_k^{P,RL}$. Let F_k be the event that the first $k-1$ packets of a window incur no random losses but the kth packet incurs a random loss. Then F_k is independent of $RTTH_l$ for all l and $\Pr(F_k) = q^{k-1}p$. The probability $P_k^{P,RL}$, which is defined before to be the probability that a random loss signal on the kth packet of a window is received, can thus be obtained from its definition as follows:

$$\begin{aligned}
P_k^{P,RL} &\triangleq \Pr\left(\bigcap_{l=1}^{k-1} RTTH_l < TO, RTTH_k + t_d \leqslant TO, F_k\right)\\
&= q^{k-1}p \Pr(RTTH_k \leqslant TO - t_d)\\
&= q^{k-1}p \sum_{m=0}^{M_k^{RL}} \binom{m+k-1}{m} p_l^m q_l^k,
\end{aligned} \tag{12}$$

where $M_k^{RL} \triangleq \lfloor (TO - t_d - k - d + 1)/d \rfloor$. The second equality follows from Eq. (9b), and the third equality follows from Eq. (10). Notice that contrary to M_k^{TO}, the physical meaning of M_k^{RL} is the maximum number of retransmissions for the first k packets such that a random loss on the kth packet will be detected by a random loss signal, but not by a timeout signal.

For the same reason, i.e., the packets are delivered to the corresponding higher layer in order, the time taken by each window without loss signals is the TL round trip time of its last packet, i.e., $T_w^W = RTTH_w$. Eq. (10) therefore gives $\overline{T_w^W}$ as follows:

$$\begin{aligned}
\overline{T_w^W} &\triangleq \overline{RTTH_w | \text{no timeouts, no random losses}}\\
&= \sum_{m=0}^{M_w^{TO}} (md + w + d - 1) \binom{m+w-1}{m} p_l^m q_l^w / Q_w^{P,F}.
\end{aligned} \tag{13}$$

Notice that $Q_w^{P,F}$ is the previous defined $Q_k^{P,F}$ with $k = w$. By the definition of $Q_k^{P,F}$, $Q_w^{P,F}$ is the probability of no timeouts and no random losses on a window with size w. Also notice that M_w^{TO} is the previous defined M_k^{TO} with $k = w$.

Similarly, $\overline{T_w^{W,RL}}$ can be shown to be

$$\begin{aligned}
\overline{T_w^{W,RL}} &= d - 1 + t_d\\
&\quad + \sum_{k=1}^{w} \sum_{m=0}^{M_k^{RL}} q^{k-1} p (md + k)\\
&\quad \times \binom{m+k-1}{m} p_l^m q_l^k / P_w^{W,RL}.
\end{aligned} \tag{14}$$

For brevity, we omit the details.

Eqs. (11)–(14) give us the four quantities needed for solving the system Markov chain when the SLL employs the GBN protocol.

4.2. Queuing model for the SRP

As with the GBN protocol, the SRP protocol also delivers packets to the corresponding higher layer in order and Inequality (9b) still holds. Therefore, the first two equalities for $Q_k^{P,F}$ and $P_k^{P,RL}$ in Eq. (11) and (12) still hold. However in SRP, at the SLL the packets are not acknowledged in order anymore, and a packet is delivered to the corresponding higher layer only when all the previous packets and itself are correctly received by the receiver. Thus:

$$RTTH_k = \max_{l=1,2,\ldots,k} RTTL_l, \tag{15a}$$

$$RTTH_l \leqslant RTTH_k \text{ for all } l \leqslant k. \tag{15b}$$

For convenience, define the function $f(k,z)$ to be the probability that the TL round trip time of the kth packet is less than or equal to a variable z, that is

$$f(k,z) \triangleq \Pr(RTTH_k \leqslant z). \tag{16}$$

Then from the first two equalities in Eqs. (11) and (12), $Q_k^{P,F}$ and $P_k^{P,RL}$ can be expressed in terms of $f(k,z)$ as follows

$$Q_k^{P,F} = f(k, TO-1), \tag{17}$$

$$P_k^{P,RL} = q^{k-1} p f(k, TO-t_d). \tag{18}$$

Now express $\overline{T_w^W}$ in terms of the function $f(k,z)$. Again as in the GBN protocol, in the SRP protocol the in-order delivery of packets from SLL to the corresponding higher layer means that $T_w^W = RTTH_w$. Moreover, T_w^W is a non-negative integer valued random variable. Together with its definition, its expected value can thus be expressed as

$$\begin{aligned}\overline{T_w^W} &= \sum_{z=0}^{\infty} \Pr(T_w^W > z \mid \text{no timeout, no random loss}) \\ &= \sum_{z=0}^{\infty} \Pr(RTTH_w > z \mid \text{no timeout, no random loss}).\end{aligned}$$

By straightforward derivations and noting that $\min(RTTH_w) = w + d - 1$, it can be shown that

$$\overline{T_w^W} = TO - 1 - \sum_{z=w+d-1}^{TO-2} f(w,z)/f(w, TO-1). \tag{19}$$

For brevity, we omit the details.

$\overline{T_w^{W,RL}}$ can be obtained in the similar way as that for $\overline{T_w^W}$. We omit the details and the result is

$$\overline{T_w^{W,RL}} = TO - \sum_{k=1}^{\min(w_T,w)} \frac{\sum_{z=k+d-1}^{TO-1-t_d} f(k,z)}{f(k, TO-t_d)} P_k^{P,RL} / P_w^{W,RL}, \tag{20}$$

where $w_T \triangleq TO - t_d - d + 1$ is a constant.

Eqs. (17)–(20) show that once we know the function $f(k,z)$, the four quantities Q_k^{PF}, $P_k^{P,RL}$, $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$ can be obtained. For the two cases when $k \leqslant d$ and $k > d$, the following paragraphs derive queuing models for SRP to obtain the function $f(k,z)$. Note that by equality (15a) and the definition of $f(k,z)$ (Eq. (16)), we have

$$f(k,z) = \Pr\left(\bigcap_{l=1}^{k} RTTL_l \leqslant z\right). \tag{21}$$

For convenience, starting from the time the SLL sender receives a window of packets, we index the packets and time slots in order, that is, packet k is the kth packet and slot k is the kth slot.

1. The case of $k \leqslant d$: When the SLL employs the SRP protocol, the service times of packets, denoted by X_l for packet l, are geometrically distributed, independent of each other, and independent of the waiting times of the packets. The distribution is $\Pr(X_l = (m+1)d) = p_l^m q_l$, where m is the number of retransmissions before the successful transmission of packet l. Moreover, in the case of $k \leqslant d$, the waiting time of packet l is fixed to be $l-1$ for all $l = 1, \ldots, k$. The SLL round trip times of the packets are thus $RTTL_l = l - 1 + X_l$ for $l = 1, \ldots, k$, and therefore independent of each other. Together with Eq. (21), $f(k,z)$ can then be written in the following simple product form:

$$f(k,z) = \prod_{l=1}^{k} \Pr(RTTL_l \leqslant z) = \prod_{l=1}^{k} \left(1 - p_l^{M_{zl}^r + 1}\right), \tag{22}$$

where $M_{zl}^{r} \triangleq \lfloor (z-l+1)/d \rfloor - 1$ is the maximum allowed total number of retransmissions of packet l such that $RTTL_l \leqslant z$.

Eq. (22) gives the function $f(k,z)$ when $k \leqslant d$.

2. The case of $k > d$: When $k > d$, packet service times are still geometrically distributed and independent of each other. However, the waiting times of packets are no longer fixed and independent of each other. As a result, $RTTL_l$ for $l = 1,\ldots,k$ are no longer independent of each other, and $f(k,z)$ no longer has the simple product form as with the case $k \leqslant d$ in Eq. (22). Nevertheless, we'll show that given the waiting time of packet k, denoted by R_k, to be r, the conditional probability $\Pr(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r)$ still has a simple product form. Bayes rule then gives a simple sum-product form of $f(k,z)$. In the following we will first derive the distribution of R_k, then find the conditional probability $\Pr(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r)$ and thus $f(k,z)$.

First consider the distribution of R_k. The event $R_k = r$ means that the first transmission of packet k takes place at slot $r + 1$. See Fig. 3 for illustration. According to the SRP protocol, this happens if and only if the transmission at slot $r - d + 1$ was successful and at slot $r + 1$, packet k is the head-of-line packet that has never been transmitted yet. The latter happens if and only if the number of successfully transmitted packets before and including slot $r - d + 1$ is $k - d$. Since each transmission incurs a loss independently with probability p_l, the joint probability that this number is $k - d$ and the transmission at slot $r - d + 1$ is successful is $\binom{r-d}{k-d-1} p_l^{r-k+1} q_l^{k-d}$. We therefore have

$$\Pr(R_k = r) = \binom{r-d}{k-d-1} p_l^{r-k+1} q_l^{k-d}. \tag{23}$$

Next consider the conditional probability $\Pr(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r)$. Since the service time of packet k is at least d, given $R_k = r$ we have $RTTL_k \geqslant r + 1 + d$. Therefore, $\Pr(\bigcap_{l=1}^{k} RTTL_l \leqslant z) = 0$ for $z < r + 1 + d$. We henceforth only consider the case $z \geqslant r + 1 + d$.

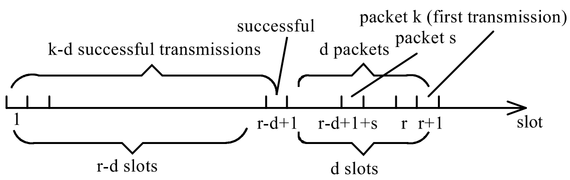

Fig. 3. Transmission pattern for SRP when $k > d$.

Moreover, for packets that were successfully transmitted before and including slot $r - d + 1$ (see Fig. 3), their round trip times satisfy $RTTL_l \leqslant r - d + 1 + d = r + 1 < z$. We hence have $f(k,z) = \Pr(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r) = \Pr(\bigcap_{l \in S} RTTL_l \leqslant z | R_k = r)$, where S is defined to be the set of packets transmitted between slot $r - d + 2$ and $r + 1$. Note that the size of S is d.

For an arbitrary packet in S, let slot $r - d + 1 + s$ be the slot when it is transmitted and call it packet s (see Fig. 3 for illustration). Due to the memoryless property of geometric distribution, the remaining service time of packet s is also geometrically distributed with $\Pr(X_s = (m+1)d) = p_l^m q_l$, where X_s denote the remaining service time and m is the remaining retransmission times. Notice that X_s is independent of each other for all $s = 1,\ldots,d$. We hence have $RTTL_s = r - d + 1 + s + X_s$ and given $R_k = r$, $RTTL_s$ is independent of each other for all $s = 1,\ldots,d$ as well. The conditional probability then becomes

$$\begin{aligned}
\Pr\left(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r\right)
&= \Pr\left(\bigcap_{s \in S} RTTL_s \leqslant z | R_k = r\right) \\
&= \prod_{s \in S} \Pr(RTTL_s \leqslant z | R_k = r) \\
&= \prod_{s=1}^{d} \Pr(r - d + 1 + s + X_s \leqslant z) \\
&= \prod_{s=1}^{d} \Pr(r - d + 1 + s + (m+1)d \leqslant z) \\
&= \prod_{s=1}^{d} \sum_{m=0}^{M_{zs}^{r}} p_l^m q_l \\
&= \prod_{s=1}^{d} \left(1 - p_l^{M_{zs}^{r}+1}\right),
\end{aligned} \tag{24}$$

where $M_{zs}^{r} \triangleq \lfloor (z - r - s)/d \rfloor$.

The sum-product form of the function $f(k,z)$ thus follows:

$$f(k,z) = \Pr\left(\bigcap_{l=1}^{k} RTTL_l \leqslant z\right)$$
$$= \sum_r \Pr\left(\bigcap_{l=1}^{k} RTTL_l \leqslant z | R_k = r\right) \Pr(R_k = r)$$
$$= \sum_{r=k-1}^{z-d} \prod_{s=1}^{d} (1 - p_l^{M_{zs}^r + 1}) \Pr(R_k = r), \quad (25)$$

where $\Pr(R_k = r)$ is given in Eq. (23). Here in the third equality, we use the fact that the minimum waiting time for packet k is $k - 1$ and the maximum waiting time of packet k such that its round trip time is less than or equal to z is $z - d$. Equation (25) together with Eq. (23) gives us the function $f(k,z)$ when $k > d$.

We conclude this section by giving the outline of how to numerically obtain the throughput of the system:

Given the system, protocol and loss parameters p, p_l, d and TO:

1. For $w \in [1, W_M]$ and $k \leqslant w$, use the SLL ARQ queuing model to obtain the four quantities $Q_k^{P,F}$, $P_k^{P,RL}$, $\overline{T_w^W}$ and $\overline{T_w^{W,RL}}$ as follows:
   For GBN, plug the parameter numbers into Eqs. (11)–(14);
   For SRP, first plug the numbers into Eq. (22) or (25) to get $f(k,z)$, then Eqs. (17)–(20) give the four quantities.
2. From the two quantities $Q_k^{P,F}$ and $P_k^{P,RL}$ and relationship (8), solve for the five probabilities, Q_w^W, $P_k^{P,TO}$ and $P_k^{P,RL}$, $P_w^{W,TO}$ and $P_w^{W,RL}$.
3. From Eqs. (4)–(6), compute the transition probabilities and reward functions.
4. Solve the system Markov chain for the steady state distribution π_i. Then Eqs. (1) and (2) gives us the steady state expected rewards per transition v^n and v^t.
5. The throughput of the system is $\lambda = v^n/v^t$ (Theorem 1).

5. Other systems

5.1. System with Tahoe algorithm

The Tahoe algorithm differs from the Reno algorithm only in that it does not use the random loss signal. All the arguments in Section 3 for systems with the Reno algorithm are also applicable for systems with the Tahoe algorithm except the parts related to random loss signals, and the model for systems with the Tahoe algorithm can be obtained from the model in Section 3 with corresponding modifications. Specifically, the system with the Tahoe algorithm can also be modelled as a finite state Markov chain, with the window size upon a timeout signal as the state. Different from the chain for the system with the Reno algorithm, the chain for the system with the Tahoe algorithm has only the states with $S = TO$.

The transition probabilities also has a simple window-based product form, and the reward functions have simple window-based sum forms as well. Specifically,

$$P_{ij}^T = \left(\prod_{n=1}^{N_{ij}-1} Q_{w_{in}}^W\right) P_{w_{iN_{ij}}}^W, \quad (26)$$

$$V_{ij}^n = \sum_{n=1}^{N_{ij}-1} w_{in} + \frac{\sum_{k=1}^{w_{iN_{ij}}} (k-1) P_k^P}{P_{w_{iN_{ij}}}^W}, \quad (27)$$

$$V_{ij}^t = \sum_{n=1}^{N_{ij}-1} \overline{T_{w_{in}}^W} + TO. \quad (28)$$

For brevity, we omit the details.

The SLL queuing models given in Section 4 are applicable to systems with the Tahoe algorithm as well. They give the quantities needed for the transition probability P_{ij}^T and the reward function V_{ij}^n and V_{ij}^t, and Eqs. (1) and (2) and Theorem 1 give the throughput.

5.2. System with multiple transmission ARQ protocol

When the link error probability is high, multiple transmission ARQ protocol had been proposed to improve the throughput of the link [22]. Initially the purpose of this protocol was to reduce the probability of receiver buffer overflow and thus increase the throughput of the link. Here in our system, multiple transmission attempts can reduce the effective SLL error probability of each packet, thus reduce the number of SLL retransmissions and TL false timeouts. As a result, an improved throughput is expected when the link loss probability is high. The protocol considered works as follows:

Each packet is transmitted r times by the satellite link ARQ sender. After receiving all the r transmissions, if at least one of these transmissions is successful, the satellite link receiver sends an ACK back to the sender. Otherwise the receiver sends back a NAK. The retransmission mechanism in case of NAKs works in the same way as the standard ARQ protocol.

The analysis of the system with multiple transmission ARQ protocol follows the same approach as in the previous sections. For brevity, we omit the details. The numerical solutions for systems under this protocol will be presented to provide some insights into the system operations.

6. Numerical results and discussions

Based on the model derived in the previous sections, we present, for different protocol and packet loss parameters, the numerical solutions for the throughput of the system. These results tell us whether implementing SLL ARQ will improve the system performance, and give us some insights into the system performance as a function of the protocol and loss parameters.

In numerically solving the system, we use data that is typical in hybrid space-terrestrial networks and satellite links. Usually for a satellite link, the round trip time of one transmission is around 1 s; the bit error rate BER is between 10^{-4} and 10^{-5}; packet size L is about 2000 bits; and transmission rate is around 10^5–10^6 bits/s. Converting these into the parameters in our system, we obtain that the link error probability p_l is between 0.02 and 0.2 and the round trip delay d is about 50–500 time slots.

In our model, the error prone bottleneck satellite link can either employ no ARQ or one of the two ARQ protocols (GBN and SRP), and the TL can either employ the Tahoe algorithm or the Reno algorithm. Therefore, we investigate the following six systems: SRP/Reno, SRP/Tahoe, GBN/Reno, GBN/Tahoe, noARQ/Reno and noARQ/Tahoe. Specifically, the SRP/Reno system refers to the system whose SLL employs the SRP and TL employs the Reno protocol. Other systems are similarly defined. Note that when the random loss probability $p = 0$ and SLL employs an ARQ protocol to recover link losses, there is no random loss signal generated. Consequently, the Tahoe and Reno algorithm are essentially the same. That is, in this case, the SRP/Reno system is equivalent to the SRP/Tahoe system, and the GBN/Reno system is equivalent to the GBN/Tahoe system.

6.1. Effects of TL timeout value

Fig. 4 plots the throughput versus TO for different systems and parameters. In particular, Fig. 4(a) plots the throughput versus TO for all the six systems considered, with $p_l = 0.01$, $p = 0$ and $d = 100$. That is, there are no random losses and all losses can be recovered by the SLL ARQ. Recall that with $p = 0$, the SRP/Tahoe and SRP/Reno systems are the same, and so are the GBN/Tahoe and GBN/Reno systems. The figure shows that for systems with SLL ARQ (upper two curves in Fig. 4(a)), either GBN or SRP, the throughput increases monotonically with TO. This is because higher TO allows more time for the SLL ARQ to recover satellite link losses and reduces the false timeout probability. Whereas for systems without ARQ (lower two curves in Fig. 4(a)), the throughput decreases monotonically with TO. This is because in this case, the only way for the system to detect the losses is via timeout signals. Higher TO makes the system take longer time to detect losses and recover them, which results in lower throughput. Notice that in most cases except when TO is very low ($TO \leqslant 2d$), the systems with SLL ARQ have higher throughput than the systems without SLL ARQ by a factor of two to eight.

Fig. 4(b) plots the throughput against TO for the SRP/Reno and SRP/Tahoe systems, with $p_l = 0.01$, $p = 0.01$ and $d = 100$. The curves for the GBN/Reno and GBN/Tahoe systems are similar and have the same shape. For clarity, they are not shown here. The figure shows that when both losses exist, the throughput of the SRP/Tahoe system first increases with TO, then decreases, while the throughput of the SRP/Reno system first

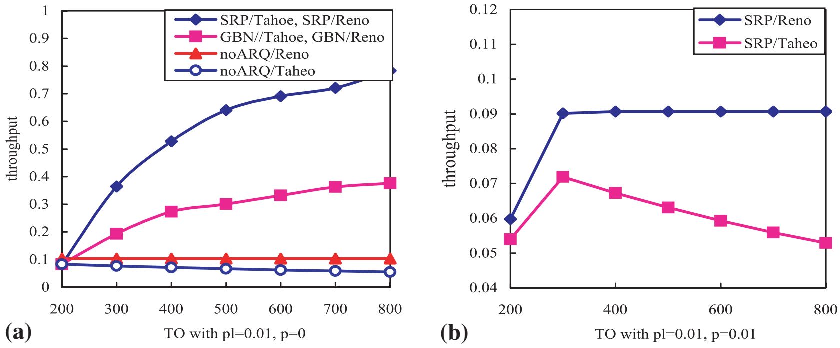

Fig. 4. Throughput versus TL timeout value *TO*.

increases with *TO*, then saturates. The increase part of both systems is because of the lower false timeout probability with increasing *TO*. After *TO* reaches a certain point, further increasing *TO* will not reduce the probability of false timeouts significantly, and the increase of the throughput for this reason should not be significant.

In addition, after *TO* is already large, for the SRP/Tahoe system, the time for the system to detect and recover random losses becomes longer. Consequently, the reduction of the throughput due to this longer time dominates, and the throughput goes down. Together with the increasing part explained above, for the SRP/Tahoe system, the curve is concave, and there exists an optimal value of *TO* that gives the best system performance. The optimal value depends on the link and loss parameters. On the contrary, when the *TO* is already large, for the SRP/Reno system, the losses will more likely be detected by random loss signals rather than by timeout signals. In this sense, higher *TO* should not affect the detection of losses, thus the throughput, significantly. Together with the increasing part when *TO* is not large and the insignificant effect on the false timeout probability when *TO* is large explained above, the throughput of the SRP/Reno system first increases, then saturates.

6.2. Effects of loss probabilities and round trip delay

Fig. 5(a) plots the throughput against the link loss probability p_l for $p = 0$, $d = 100$ and $TO = 500$. The figure shows that the throughput of the systems decreases when p_l increases, regardless which TL and SLL protocol are used. This result is consistent with our intuition on the effects of p_l on the system performance. The reason behind it is that larger loss probability means more retransmissions, longer delays and more window reductions by the TL window flow control. In addition, the figure also shows significant improvement of the system performance by implementing SLL ARQ: an order of magnitude at the maximum.

Our numerical results for the throughput versus the round trip time d (which are not shown here) give us similar plots as those in Fig. 5(a). That is, the throughput of the system decreases as d increases and significant improvement of the system performance can be achieved by implementing SLL ARQ. This result is also consistent with the system operation, since longer round trip time for one transmission means that longer time is needed for both the TL sender and the ARQ SLL sender to receive the loss signals and start retransmissions, which leads to a lower throughput.

For the SRP/Reno, GBN/Reno and noARQ/Reno systems, Fig. 5(b) shows the throughput with respect to the random loss probability p when $p_l = 0.1$, $d = 100$, $TO = 500$ and $t_d = 50$. This figure also shows improvement of the system performance by implementing ARQ. The advantage of ARQ becomes small when p is large. In this case random losses come to dominate, which results in less efficiency of the SLL ARQ in increasing the throughput. Our results for the SRP/Tahoe,

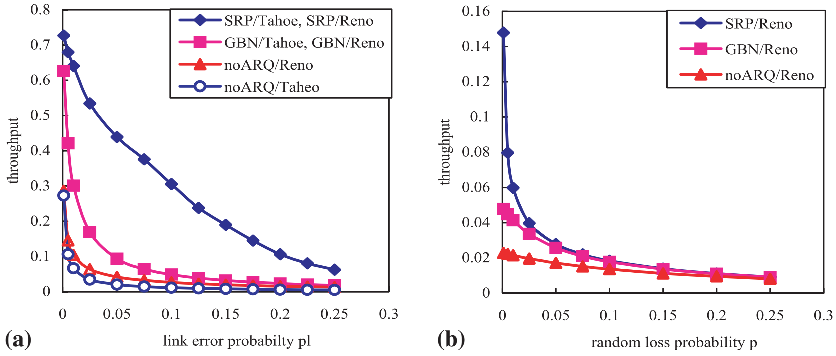

Fig. 5. Throughput versus error probability.

GBN/Tahoe and noARQ/Tahoe systems yield similar observations. For brevity, these plots are not shown here.

6.3. Effects of other parameters

Fig. 6(a) shows the throughput as a function of r, the number of transmission attempts per packet at SLL, when $p_l = 0.2$, $p = 0.01$, $d = 100$, $TO = 500$. It shows that for each system, the throughput first increases then decreases. Actually there are two opposite effects resulting from increasing r. On the one hand, compared with the standard ARQ protocol where $r = 1$, transmitting each packet r times with $r > 1$ greatly reduces the effective link loss probability seen by the SLL by a power of r, which results in significant reduction of the number of retransmissions at the SLL. Because of the large delay of the satellite link, this reduction of the retransmission times not only save a significant amount of time for one packet, but also leads to significant decrease of the variability of the packet round trip times, and thus the number of the TL false timeouts and the TL window reductions. In this sense, higher r may lead to a higher throughput. On the other hand, transmitting each packet r times also means that the effective link capacity is reduced by a factor of r, therefore greatly reducing the throughput. As a result, the overall system performance versus r depends on the tradeoff between these two effects. Fig. 6(a) shows that when r is relatively small, the benefit of increasing r dominates, while when r is relatively large, the drawback dominates. The curve is thus concave, and there exists an optimal value of r which gives the best throughput. Again, the optimal value depends on the link and protocol parameters. Notice that the figure shows a

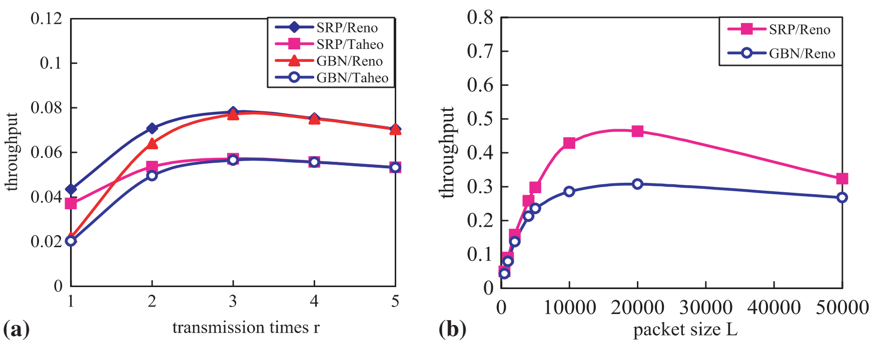

Fig. 6. Throughput versus SLL transmission times and packet size.

significant increase in throughput by using the optimum r (for example, a factor of 4 for the GBN/Reno system).

Fig. 6(b) plots the throughput as a function of the packet size L for the SRP/Reno and GBN/Reno systems, when the random loss probability $p = 0.01$, the round trip delay of the link is 1 s, BER=10^{-5} and the transmission rate is 10^5 bits/s. Here in order to illustrate the effects, we deliberately extend the range of packet lengths and make the length go beyond the practical values. The figure shows that the throughput first goes up then goes down with increasing L. The increase in throughput is because of the decrease in SLL round trip delay d (as measured in time slots) due to increasing L, and the decrease in throughput is due to the increase in packet error probability p_l (again, because of the larger packet size). There exists an optimal value of L that gives the best performance. The optimal value depends on the link parameters and the protocols employed, and can significantly improve the throughput as well (a factor of 6 to 9 with the parameters discussed). Our numerical results for the other four systems, the SRP/Tahoe and GBN/Tahoe systems and the noARQ/Reno and noARQ/Reno systems, give the same trends as those in Fig. 6(b). For brevity, they are not shown here.

6.4. Is SLL ARQ beneficial?

Now let us assess whether the SLL ARQ is beneficial or not. First consider the systems with the Tahoe protocol. In these systems, implementing the SLL ARQ can recover some of the link losses (which will otherwise lead to TL timeouts) and thus decrease the number of window reductions by the TL. Therefore the SLL ARQ should improve the throughput and be beneficial. This is evidenced by Figs. 4–6 where the SRP/Tahoe and GBN/Tahoe systems have better performance than the noARQ/Tahoe system.

Whereas when the TL employs the Reno protocol, the SLL ARQ has two opposite effects on the system performance. On one hand, the SLL ARQ can recover some of the link losses, and thus decrease the number of window reductions, which would lead the system back to the congestion avoidance phase if no ARQ were employed. As a result, the throughput should increase. On the other hand, the SLL ARQ introduces false timeouts. These false timeouts lead the system back to the slow start phase instead of the congestion avoidance phase if no ARQ were employed. Consequently, the throughput should decrease. Figs. 4–6 show that, in all cases except when the TL timeout value TO is very small ($TO \leqslant 2d$, see Fig. 4(a)), the benefit of employing ARQ overcomes the drawback and the SRP/Reno and GBN/Reno systems give better performance than the noARQ/Reno system.

Since in most cases except when TO is very low, employing ARQ yields significant improvement of the system throughput (for example, a factor of two to eight from Fig. 4(a) and an order of magnitude at the maximum in Fig. 5(a)), we thus suggest the implementation of ARQ over the satellite link.

7. Conclusion

In this paper we provide an exact model for hybrid space-terrestrial networks with AIMD transport protocols and satellite link layer ARQ protocols. Both GBN and SRP are considered, and the delay of ARQ ACKs is also taken into account. Numerical solutions for the throughput as a function of different protocol and packet loss parameters are presented. Our analysis shows that for most situations of interest, it is better for the satellite link to implement ARQ.

We were also able to obtain some insights on the system performance as a function of various protocol parameters, such as the transport layer time-out value; the link layer packet size and the number of transmission attempts per packet at the link layer. We show that by proper setting of these parameters, system throughput can be improved by nearly an order of magnitude.

There are two extensions to the work in this paper. One is to analyze TCP performance when there exist multiple connections. The other is to include non-persistent users, that is, users that have limited packets to sent. These two extensions will be our future work.

References

[1] M. Allman, S. Dawkins, D. Glover, J. Griner, D. Tran, T. Herderson, J. Heidemann, J. Touch, H. Kruse, S. Ostermann, K. Scott, J. Semke, Ongoing TCP Research Related to Satellites, RFC 2760, February 2000.

[2] M. Allman, C. Hayes, H. Kruse, S. Ostermann, TCP performance over satellite links, Proc. 5th Int. Conf. on Telecommunication Systems, March 1997.

[3] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, R.H. Katz, A comparison of mechanisms for improving TCP performance over wireless links, ACM/IEEE Trans. Network. 5 (6) (1997) 756–769.

[4] C. Barakat, E. Altman, W. Dabbous, On TCP performance in a heterogeneous network: a survey, IEEE Commun. Mag. 38 (1) (2000) 40–46.

[5] D. Bertsekas, R. Gallager, Data Networks, second ed., Prentice Hall, Upper Saddle River, NJ, 1992.

[6] A.F. Canton, T. Chahed, End-to-end reliability in UMTS: TCP over ARQ, Global Telecommunications Conference 2001, GLOBECOM '01, vol. 6, IEEE, 2001.

[7] H.M. Chaskar, T.V. Lakshman, U. Madhow, TCP over wireless with link level error control: analysis and design methodology, ACM/IEEE Trans. Network. 7 (5) (1999) 605–615.

[8] C.F. Chiasserini, M. Meo, Modeling interactions between link layer and transport layer in wireless networks, 12th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2001, vol. 1, 2001.

[9] C.F. Chiasserini, M. Meo, A reconfigurable protocol setting to improve TCP over wireless, IEEE Trans. Veh. Technol. 51 (6) (2002) 1608–1620.

[10] A. Chockalingam, M. Zorzi, V. Tralli, Wireless TCP performance with link layer FEC/ARQ, Proc. IEEE ICC'99, vol. 2, 1999, pp. 1212–1216.

[11] A. DeSimone, M.C. Chuah, O.C. Yue, Throughput performance of transport-layer protocols over wireless LANs, Proc. IEEE GLOBECOM'93, December 1993.

[12] R.G. Gallager, Discrete Stochasic Processes, Kluwer Academic Publishers, Boston, 1995.

[13] T.R. Henderson, R.H. Katz, Transport protocols for Internet-compatible satellite networks, IEEE J. Sel. Areas Commun. 17 (2) (1999) 326–344.

[14] A.G. Konheim, A queuing analysis of two ARQ protocols, IEEE Trans. Commun. 28 (7) (1980) 1004–1014.

[15] F. Lefevre, G. Vivier, Optimizing UMTS link layer parameters for a TCP connection, Vehicular Technology Conference, 2001, VTC 2001 Spring, IEEE VTS 53rd, vol. 4, 2001.

[16] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, Modeling TCP throughput: a simple model and its empirical validation, Proc. of ACM SIGCOMM'98 Conf. on Applications, Technologies, Architectures, and Protocols for Computer Communication, 1998, pp. 303–314.

[17] J.S. Stadler, J. Gelman, Performance enhancement for TCP/IP on a satellite channel, MILCOM 98 Proceedings, vol. 1, IEEE, 1998.

[18] J.S. Stadler, A link layer protocol for efficient transmission of TCP/IP via satellite, MILCOM 97 Proceedings, IEEE, vol. 2, 1997.

[19] C. Partridge, T.J. Shepard, TCP/IP performance over satellite links, IEEE Network 11 (5) (1997) 44–49.

[20] W.R. Stevens, TCP/IP Illustrated, vol. 1, Addison-Wesley, New York, 1994.

[21] D. Towsley, J.K. Wolf, On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission schemes, IEEE Trans. Commun. 27 (4) (1979) 693–702.

[22] E.J. Weldon Jr., An improved selective-repeat ARQ strategy, IEEE Trans. Commun. 30 (3) (1982) 480–486.

[23] J.W.K. Wong, V.C.M. Leung, Improving end-to-end performance of TCP using link-layer retransmissions over mobile internetworks, Proc. IEEE ICC'99, vol. 1, 1999, pp. 324–328.

[24] M. Yoshimoto, T. Takine, Y. Takahashi, T. Hasegawa, Waiting time and queue length distributions for go-back-N and selective-repeat ARQ protocols, IEEE Trans. Commun. 41 (11) (1993) 1687–1693.

[25] L. Zhang, Why TCP timers don't work well, Proc. ACM SIGCOMM Conf. on Communications Architecture and Protocols, September 1986.

[26] M. Zorzi, R.R. Rao, L.B. Milstein, ARQ error control for fading mobile radio channels, IEEE Trans. Veh. Technol. 46 (2) (1997) 445–455.

Chunmei Liu received her B.S. and first M.S. from Harbin Institute of Technology (HIT), and second M.S. from Massachusetts Institute of Technology (MIT). She is currently working towards her PhD degree in the Laboratory for Information and Decision Systems (LIDS), MIT. Her research interests are in the area of heterogenous data networks with emphasis on cross-layer protocol interactions.

Eytan Modiano received his B.S. degree in Electrical Engineering and Computer Science from the University of Connecticut at Storrs in 1986 and his M.S. and PhD degrees, both in Electrical Engineering, from the University of Maryland, College Park, MD, in 1989 and 1992, respectively. He was a Naval Research Laboratory Fellow between 1987 and 1992 and a National Research Council Post Doctoral Fellow during 1992–1993 while he was conducting research on security and performance issues in distributed network protocols. Between 1993 and 1999 he was with the Communications Division at MIT Lincoln Laboratory where he designed communication protocols for satellite, wireless, and optical networks and was the project leader for MIT Lincoln Laboratory's

Next Generation Internet (NGI) project. He joined the MIT faculty in 1999, where he is presently an Associate Professor in the Department of Aeronautics and Astronautics and the Laboratory for Information and Decision Systems (LIDS). His research is on communication networks and protocols with emphasis on satellite, wireless, and optical networks.