# On the Interaction between Layered Protocols: the Case of Window Flow Control and ARQ

Chunmei Liu and Eytan Modiano
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology
email: mayliu@mit.edu, modiano@mit.edu

*Abstract* — **This paper studies the interactions between protocols at different network layers. In particular, the system considered consists of two end nodes communicating over a multi-hop network using a higher layer (HL) protocol and a lower layer (LL) protocol. The HL at the two end nodes implements an Additive-Increase-Multiplicative-Decrease (AIMD) congestion control mechanism, similar to the operation of TCP; and the LL protocol implements a link layer retransmission mechanism (ARQ) over the error prone bottleneck link. Both Go-Back-N (GBN) and Selective Repeat (SRP) ARQ protocols are considered. We examine the impact of packet losses due to transmission errors as well as packet losses due to congestion (i.e., buffer overflow). The throughput of the system is derived using the theory of Markov chain with rewards. Simple queuing models are developed for the LL ARQ. The numerical solutions for the throughput as a function of different protocol and packet loss parameters are also presented.**

## I. Introduction

Data networks employ a layered architecture where functions to be performed by the network are divided among protocols at the different layers. This division of responsibility simplifies network implementation and allows for interoperability among different networks. However, this division also introduces inefficiency into the network both because some functions are inevitably duplicated and because protocols at different layers may be incompatible. The goal of this research is to explore the interaction between protocols at different layers; focusing in particular on the transport and the link layers.

Typically in a data network, the link layer is responsible for reliable transmission of packets across a single link in the network. When operating over links with a high probability of packet errors (e.g., satellite or wireless links), link layers employ a packet retransmission protocol (ARQ) for recovering from packet errors. Transport layers, on the other hand, are responsible for transmission of messages end-to-end across the network. Hence, they too often employ an error-recovery protocol for recovering from end-to-end packet errors that are not otherwise removed at the link layer (e.g., packet loss due to buffer overflow).

Furthermore, transport layer protocols, such as TCP, also employ congestion control mechanisms for reacting to congestion in the network. For example, TCP's congestion control mechanism is triggered in reaction to packet losses that are presumably due to congestion. TCP detects a packet loss using a timeout signal; in addition to retransmitting the lost packet, TCP assumes that this loss is due to a buffer overflow and reduces its window size (thereby reducing the transmission rate). It has been known that this behavior of TCP is not appropriate over satellite or wireless links where packet losses are more likely to be due to transmission errors than to congestion. This problem can be overcome with an ARQ protocol at the link layer. However, the presence of an ARQ protocol can lead to other more subtle problems. In particular, the presence of a link layer ARQ protocol may introduce a large variability in the round trip delay for sending a packet. This variability, again, may result in a false time-out at the higher layer leading to both unnecessary retransmissions and unnecessarily triggering TCP's congestion mechanism.

This phenomenon, in general, is not limited to the interaction between TCP at the transport layer and some ARQ mechanisms at the link layer. It can occur at various layers that may use retransmission mechanisms. For example, often application protocols may use a timeout mechanism to retransmit a file or reload a web page. There again, retransmissions may be falsely triggered due to the lower layer ARQ. Hence in exploring this problem we refer to the two layers as the higher layer (HL) and lower layer (LL). Our goal is to understand this delicate interaction so that protocols at different layers can be designed to work effectively together.

Recently a number of papers have examined system performance with multi-layer protocols [1, 3, 5, 8, 10]. In [10] several simple higher layer protocols are investigated when the link layer implements ARQ. Also, [1] examines alternative schemes designed to improve the TCP performance with lossy links via simulation and suggests the use of a TCP-aware link layer protocols. In addition, [3] also examines the TCP performance with link layer FEC/ARQ via simulation, assuming instant feedback. There are many papers on the analysis of ARQ as well. In [7] queuing models are developed for the GBN protocol, and [6] provides queuing models for both the GBN and SRP protocols. Generally queuing models are used as a tool to analyze the ARQ protocols for different channels and network structures [9, 11].

Different from earlier work on the multi-layer protocols, this paper provides an analytical framework to examines the interactions between protocols of window flow control and ARQ at different layers. The system investigated consists of two end nodes communicating over a multi-hop network. The bottleneck link of the system is also the error prone link. This architecture corresponds to a hybrid network that includes a satellite link and some terrestrial links, where usually the satellite link is the bottleneck link as well as the error prone link. The LL over the error prone bottleneck link implements the Go-Back-N (GBN) or Selective Repeat (SRP) retransmission mechanisms to recover the transmission errors. The HL at the end nodes implements an Additive-Increase-

Multiplicative-Decrease (AIMD) window flow control mechanism, similar to that of TCP. The losses considered include both the transmission losses and the congestion losses. The HL sends packets in batches (windows), and uses timeouts as indications of packet losses. The batch size changes according to the AIMD rule, which will be described in detail later.

This paper models the system as a finite state Markov chain with reward functions associated with the number of successfully transmitted packets and the time taken for the transmissions. Queuing models for the LL that implements the GBN and SRP are also developed. The results are used to obtain the transition probabilities and the reward functions of the Markov chain, and the throughput of the system is derived by the theory of Markov chain with reward functions.

The paper is organized as follows: Section 2 describes in detail the system under consideration. Section 3 presents our modeling process for the system and gives the expressions for the throughput of the system. Section 4 provides the simple queuing models for the ARQ protocols, whose results are used to solve the system model. Section 5 discusses the numerical results for different protocol and packet loss parameters. Finally, Section 6 gives our conclusion and directions for future work.

## II. System Description

The system we considered consists of two end nodes communicating over a multi-hop network, as shown in Figure 1. The two end nodes communicate using layered protocols where the higher layer (HL) protocol is responsible for the functions of transport protocols such as TCP, the medium layer (ML) is responsible for the functions of routing protocols such as IP, and the lower layer (LL) protocol is responsible for the functions of a link layer. The sender has unlimited packets to be transmitted to the receiver, and these packets have fixed lengths. The bottleneck link between the two end nodes is also error prone. As mentioned before, this system corresponds to a hybrid network with one satellite link and some terrestrial links. The LL for the error prone bottleneck link implements an ARQ protocol, where both the GBN and SRP are considered. The other LLs do not employ ARQ. We call the LL that employs the ARQ protocol the ARQ LL, and the corresponding link the ARQ link. The time for the ARQ link to transmit one packet is defined to be one time unit. In this way, the time is divided into time slots, and the transmission of each packet over the ARQ link takes one time slot. The round trip time of one transmission over the ARQ link, defined to be the interval between the time the ARQ link sender sends out a packet and the time the 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.

We consider two types of packet losses: link losses and random losses. The link losses refer to the losses caused by the erroneous transmissions over the ARQ link, and the random losses refer to losses caused by all other reasons, for example, buffer overflow due to congestion and erroneous transmissions over the other links that do not employ a LL ARQ. The link losses can be recovered by the LL ARQ, and the random losses can be recovered by the end-to-end HL retransmissions. Packets incur losses independently of each other. Each packet incurs a random loss with probability $p_l$, and each transmission over the ARQ link incurs a link loss with probability $p$.

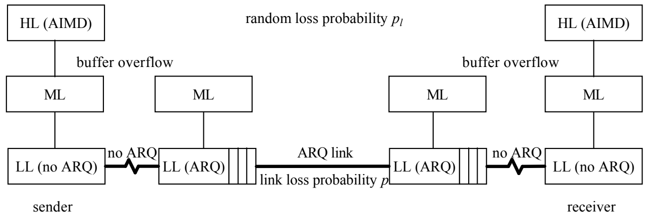


Fig. 1: System with Two Nodes Communicating over a Multi-hop Network

The HL sends packets in batches. The size of the batches is the current window size of the HL. Once the HL receives all the acknowledgments of its previous window of packets, it sends out the next window of packets. The HL uses a timeout value $TO$ for each packet. When the age of an unacknowledged packet exceeds the timeout value $TO$, a HL timeout occurs. At this moment, the HL changes the window size according to the AIMD algorithm, and restarts the transmission from the packet that incurs the timeout.

The window-update-algorithm of the HL has two parameters: current window size $W$ and a threshold $W_t$. The 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 $W \geq W_t$, the window size is increased by 1.
- When a packet incurs a timeout, the window size is set to be 1, and $W_t$ is set to be half of the window size when the timeout occurs.

One can see that this algorithm is an additive-increase-multiplicative-decrease algorithm. It is similar to the TCP window-update algorithm, except that there the window size is updated upon each acknowledgment, while here the window size is updated when a batch of acknowledgments is received. Our batch assumption is used to make the analysis of the protocols tractable.

The LL ARQ window size is no less than the round trip delay $d$, so that its link capacity can be fully utilized and the actual transmission rate over it is limited only by the HL window size. Furthermore, negative acknowledgement (NAK) signals are assumed to be used. (An equivalent protocol is one where the LL does not use NAKs, but its timeout value is $d$). That is, after $d$ time slots, the sender knows whether the transmission is successful or not. The order of the packet transmissions follows the standard GBN or SRP rules [2].

Our goal is to develop a model for the system, obtain its throughput as a function of the protocol and loss parameters, and explore the interaction between the two protocols at the two layers.

## III. System Model

In order to simplify the analysis, we assume that when the HL timeout occurs, the system clears out all the packets currently in the system. When the HL timeout occurs, the future behavior of the system is completely determined by the HL window size, or equivalently, by the current window threshold $W_t$, and is independent of its past. Thus, the system can be modeled by a Markov chain, with the ending window size (i.e., the window size when a TO occurs) as the states, as shown in Figure 2. Moreover, since the HL timeout value is fixed, there exists a maximum value of the window size, denoted by $W_m$, such that at steady state, once the window size reaches this value, a HL timeout must occur. Therefore, the window size will never exceed the maximum value $W_m$, and the system chain is a finite state Markov chain.

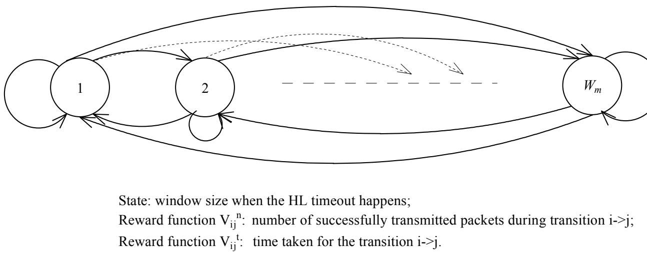


Fig. 2: System Markov Chain

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 $i$ are:

$$v_i^n = \sum_{j=1}^{W_m} V_{ij}^n P_{ij}^T \ and \ v_i^t = \sum_{j=1}^{W_m} V_{ij}^t P_{ij}^T, \tag{1}$$

where $P_{ij}^T$ is the transition probability from state $i$ to state $j$. The superscript $T$ denotes transition.

The steady state expected rewards per transition are thus:

$$v^n = \sum_{i=1}^{W_m} \pi_i v_i^n \ and \ v^t = \sum_{i=1}^{W_m} \pi_i v_i^t, \tag{2}$$

where $\pi_i$ is the steady state distribution of the Markov chain.

By the theory of Markov chain with reward functions, given the steady state expected rewards $v^n$ and $v^t$, the throughput of the system can be shown to be the ratio of the two [4]:

$$\lambda = \frac{v^n}{v^t}. \tag{3}$$

We can now derive the transition probabilities of the Markov chain as well as the reward functions. Then the throughput of the system can be obtained using equation 3.

## A Transition Probabilities

This subsection derives the transition probability of the Markov chain. Consider the transition from state $i$ to state $j$. By the definition of the state of the system Markov chain, $i$ is the window size when the last HL timeout happens, and $j$ is the window size when the next HL timeout happens. Moreover, according to the HL window update algorithm, we know exactly the number of successfully transmitted windows between these two timeouts. For example, when $i = 27$, the window threshold is set to $W_t = \lfloor i/2 \rfloor = 13$. The window size then evolves as: 1, 2, 4, 8, 16, 17, 18, .... If $j = 17$, then the number of successfully transmitted windows during this transition is 5, and the timeout that ends the transition happens on the sixth window. Let $N$ be the window number such that when starting from state $i$, the Nth window has size $j$. Then the problem of finding the transition probability $P_{ij}^T$ becomes finding the probability that the next HL timeout happens on the Nth window when the window threshold is $W_t = \lfloor i/2 \rfloor$.

Furthermore, the HL sends out a new window of packets only after it receives all the acknowledgements of the packets of the previous window. Therefore, the HL timeout probability of each window is independent of each other. Starting from state $i$, let $Q_{in}^W$ be the probability that the HL timeout does not happen on the nth window and $P_{in}^W$ be the probability that the HL timeout happens on the nth window. Here the superscript $W$ denotes that the probability is related to windows. Then, the transition probability $P_{ij}^T$ has a product form as follows:

$$P_{ij}^T = (\prod_{n=1}^{N-1} Q_{in}^W) P_{iN}^W. \tag{4}$$

and

$$P_{in}^W = 1 - Q_{in}^W, \tag{5}$$

There are two causes of the HL timeouts. One cause is the random losses, which cannot be recovered by the LL ARQ. Another cause is the possible long time taken by the retransimissions of the LL ARQ due to transmission errors. Since the HL timeout signal is designed to recover the losses that cannot be recovered by the LL, we call the HL timeout caused by the second cause a false timeout. Note that in our system these two causes are independent of each other.

Let $Q_{nk}^P$ be the probability of no timeout on the first $k$ packets of the nth window, and let $Q_{nk}^{PF}$ be the probability of no false timeout on the first $k$ packets of the nth window. Here the superscript $P$ denotes that this probability is related to packets. Let $W_n$ be the window size of the nth window. Then, since each packet incurs losses independent of each other, we have:

$$Q_{nk}^P = Q_{nk}^{PF}(1 - p_l)^k, \tag{6}$$

$$Q_{in}^W = Q_{nW_n}^P. \tag{7}$$

Let $P_{nk}^P$ be the probability that the HL timeout happens on the kth packet of the nth window. Then $P_{nk}^P$ can be expressed using $Q_{nk}^P$ as follows:

$$P_{nk}^P = Q_{n(k-1)}^P - Q_{nk}^P. \tag{8}$$

where the initial value $Q_{n0}^P = 1$.

We obtain $Q_{nk}^{PF}$ from the LL queuing models, and equation 6, 7, 5 and 4 give us the transition probability $P_{ij}^T$.

## B Reward Function $V_{ij}^n$

This subsection presents the derivation of the reward function $V_{ij}^n$. Since the HL window size is updated according to a fixed rule, the number of successfully transmitted packets during one transition is simply the sum of the window sizes before the timeout happens and the number of successfully transmitted packets of the window that incurs the HL timeout, that is, $\sum_{n=1}^{N-1} W_n + k - 1$, where $k$ is the packet within window $N$ that incurs the HL timeout.

Moreover, by the definition of $Q_{in}^W$ and $P_{Nk}^P$, the probability that the HL timeout happens on packet $k$ of the Nth window is $\prod_{n=1}^{N-1} Q_{in}^W \cdot P_{Nk}^P$. Therefore, the expected reward $V_{ij}^n$ is:

$$\begin{aligned} V_{ij}^n &= \frac{\sum_{k=1}^{W_N}[(\sum_{n=1}^{N-1} W_n + k - 1)(\prod_{n=1}^{N-1} Q_{in}^W \cdot P_{Nk}^P)]}{\sum_{k=1}^{W_N}(\prod_{n=1}^{N-1} Q_{in}^W \cdot P_{Nk}^P)} \\ &= \sum_{n=1}^{N-1} W_n + \frac{\sum_{k=1}^{W_N}(k-1)P_{Nk}^P}{P_{iN}^W}. \end{aligned} \quad (9)$$

Here $P_{Nk}^P$ and $P_{in}^W$ can be obtained from equation 8 and 5, respectively. Equation 9 thus gives us the reward function $V_{ij}^n$.

## C Reward Function $V_{ij}^t$

We derive the reward function $V_{ij}^t$ in this subsection. Let $T_{ij}$ be the time taken by one transition. Then $T_{ij}$ is the sum of the time taken by the $N-1$ windows that did not incur timeout and the timeout value TO, that is:

$$T_{ij} = \sum_{n=1}^{N-1} T_{in}^W + TO,$$

where $T_{in}^W$ is the time taken by the nth window starting from state $i$.

Furthermore, the expected reward $V_{ij}^t$ is the expected value of $T_{ij}$, that is:

$$V_{ij}^t = \overline{T_{ij}} = \sum_{n=1}^{N-1} \overline{T_{in}^W} + TO, \quad (10)$$

where $\overline{T_{ij}}$ is the expected value of $T_{ij}$ and $\overline{T_{in}^W}$ is the expected value of $T_{in}^W$ given no timeout happens on the nth window. Here for simplicity, we did not indicate the condition on the symbol.

The LL queuing models will give us $\overline{T_{in}^W}$, which will be derived in Section 4. Equation 10 thus gives us the reward function $V_{ij}^t$.

# IV. LL Queuing Models

This section develops simple queuing models for the LL ARQ protocol with batch arrivals. The two quantities $Q_{nk}^{PF}$ and $\overline{T_{in}^W}$, which are needed for solving the system Markov chain, are obtained from these queuing models.

## A Queuing Model for the GBN

The GBN protocol in our model can be modeled as a queuing system with independent geometrically distributed service time [2] as follows:

$$Pr(X_k = md + 1) = p^m q,$$

where $X_k$ is the service time of packet $k$ and $m$ is the number of retransmissions of packet $k$ before it is successfully acknowledged.

Let's first find the round trip time of one window, then the two quantities $Q_{nk}^{PF}$ and $\overline{T_{in}^W}$.

Since packets arrive at the LL in batches, the round trip time of packet $k$ (defined to be the interval between the time packet $k$ arrives at the LL queue and the time packet $k$ leaves the queue) is the sum of its service time and the service times of the $k-1$ previous packets within the same window. Since the service times of packets are independent of each other, the round trip time of packet $k$ can be shown to be binomially distributed and have the following distribution:

$$Pr(RTT_k = md + k + d - 1) = \binom{m+k-1}{m} p^m q^k, \quad (11)$$

where $RTT_k$ is the the round trip time of packet $k$ and $m$ is the total number of retransmissions of the first $k$ packets in the batch. Note that extra $d-1$ slots are added for allowing the ACK to come back to the sender after the packet leaves that LL queue.

Moreover, since in GBN the packets are acknowledged in order, the event of no timeout on the first $k$ packets is equivalent to the event that the round trip time of packet $k$ is less than the timeout value $TO$. Thus:

$$Q_{nk}^{PF} = Pr(RTT_k < TO) = \sum_{m=0}^{M_k} \binom{m+k-1}{m} p^m q^k, \quad (12)$$

where $M_k \equiv \lceil \frac{TO-k-d+1}{d} \rceil - 1$ is the maximum number of retransmissions of the first $k$ packets without causing a timeout.

For the same reason, the time taken by each window is the round trip time of its last packet, i.e., $T_{in}^W = RTT_{W_n}$. From equation 11, which gives us the distribution of $RTT_{W_n}$, we have:

$$\begin{aligned} \overline{T_{in}^W} &= \overline{RTT_{W_n}} \triangleq \overline{RTT_{W_n}|no\,timeout} \\ &= \frac{\sum_{m=0}^{M_{W_n}} (md+W_n+d-1)\binom{m+W_n-1}{m} p^m q^{W_n}}{Q_{in}^W}, \end{aligned} \quad (13)$$

where $M_{W_n} \equiv \lceil \frac{TO-W_n-d+1}{d} \rceil - 1$ is the maximum allowed total number of retransmissions of all the packets in the window so that no timeout occurs in this window.

Equation 12 and 13 give us the two quantities needed for solving the system Markov chain for GBN.

## B Queuing Model for the SRP

By definition, $Q_{nk}^{PF}$ is the probability that the round trip times of the first $k$ packets are less than the timeout value $TO$, i.e.:

$$Q_{nk}^{PF} = Pr(\bigcap_{l=1}^{k} RTT_l < TO). \quad (14)$$

For $\overline{T_{in}^W}$, note that $T_{in}^W$ is a non-negative integer valued random variable. Thus, it can be shown that its expected value is as follows:

$$\begin{aligned} \overline{T_{in}^W} &= \sum_{z=0}^{\infty} Pr(T_{in}^W > z|no\,timeout) \\ &= TO - \sum_{z=0}^{TO-1} \frac{Pr(T_{in}^W \le z)}{Q_{nW_n}^{PF}} \end{aligned}$$

$$= TO - \sum_{z=0}^{TO-1} \frac{Pr(\bigcap_{l=1}^{W_n} RTT_l \le z)}{Q_{nW_n}^{PF}} \quad (15)$$

Equation 14 and 15 show that once we know the two probabilities $Pr(\bigcap_{l=1}^{k} RTT_l < TO)$ and $Pr(\bigcap_{l=1}^{W_n} RTT_l \le z)$, the two quantities, $Q_{nk}^{PF}$ and $\overline{T_{in}^{W}}$, can be obtained. For the two cases when $k \le d$ and $k > d$, the following subsections use queuing models for SRP to obtain these two probabilities.

## B.1 The Case of $k \le d$

When the LL employs the SRP protocol, the service times of packets are geometrically distributed, independent of each other, and independent of the waiting times of the packets. Moreover, in the case of $k \le d$, the waiting time for each of the first $k$ packets is fixed and thus independent of each other as well. Since the round trip time of each packet is the sum of its waiting time and service time, this means that the round trip times are also independent of each other. Thus in this case, both of the two probabilities, $Pr(\bigcap_{l=1}^{k} RTT_l < TO)$ and $Pr(\bigcap_{l=1}^{W_n} RTT_l \le z)$, have product forms. The details are shown below.

Let $R_l$ be the waiting time of packet $l$ in the batch. Then, in the first $k$ slots, according to the SRP protocol, the LL sends out the $k$ packets in order and $R_l = l-1$ for $l = 1, 2, \cdots, k$, $k \le d$. Let $X_l$ be the service time of packet $l$ and $RTT_l$ be the round trip time of packet $l$, then $RTT_l = R_l + X_l$ for $l = 1, 2, \cdots, k$, and $X_l$ has the distribution of $Pr(X_l = (m+1)d) = p^m q$, where $m$ is the number of retransmissions before the successful transmission of packet $l$.

Since $RTT_l$ are independent of each other,

$$Pr(\bigcap_{l=1}^{k} RTT_l < TO) = \prod_{l=1}^{k} Pr(RTT_l < TO) = \prod_{l=1}^{k} (1 - p^{M_l+1}) \quad (16)$$

where $M_l \equiv \lceil \frac{TO-l+1}{d} \rceil - 2$ is the maximum allowed total number of retransmissions of the packet $l$ such that packet $l$ will not incur a timeout. Similarly,

$$Pr(\bigcap_{l=1}^{W_n} RTT_l \le z) = \prod_{l=1}^{W_n} (1 - p^{M_{zl}+1}), \quad (17)$$

where $M_{zl} \equiv \lfloor \frac{z-l+1}{d} \rfloor$;

Equation 16 and 17 give us the two desired probabilities when $k \le d$.

## B.2 The Case of $k > d$

When $k > d$, the service times of packets are still geometrically distributed and independent of each other. However the waiting times are no longer fixed and independent of each other. Nevertheless, the SRP protocol gives the following three facts. First, given that the waiting time of packet $k$ is $R_k$, before slot $R_k + 1$, the LL was transmitting one of the first $k-1$ packets in each slot. Second, the transmission in slot $R_k - d + 1$ must have been successful. Third, since the round trip time of one transmission is $d$, the packets that the LL was transmitting in the $d-1$ slots before slot $R_k + 1$ are different from each other. These packets were either transmitted for the first time, or their previous transmissions are erroneous. These facts give us the conclusion that during the first $R_k - d + 1$ slot, the number of successfully transmitted packets is $k - d$. Since each transmission incurs error independently with probability $p$, the waiting time distribution is thus given by:

$$Pr(R_k = r_k) = \binom{r_k - d}{k - d - 1} p^{r_k - k + 1} q^{k-d}. \quad (18)$$

Another less insightful and more complicated, but more straightforward approach to obtain $Pr(R_k = r_k)$ is by using the Markov property of the waiting times of the packets, that is, given the waiting time of packet $l$, the waiting time of packet $l+1$ is independent of the waiting times of those packets ahead of packet $l$. This approach will give us the same distribution as shown in equation 18. Here we omit the details of this approach.

Given the waiting time of each packet, let's now find $Pr(\bigcap_{l=1}^{k} RTT_l < TO)$. For those packets that have already been successfully transmitted before the first transmission of packet $k$, $RTT_k < TO$ guarantees that their round trip times are less than TO. For the $d$ packets (including packet $k$) that are transmitted between slot $R_k - d + 2$ and slot $R_k + 1$, their round trip times are the sum of the time slots between slot $R_k - d + 2$ and slot $R_k + 1$ when they are transmitted and the residual times. Note that the residual times are also geometrically distributed. Therefore, calculating the round trip times of these packets is equivalent to calculating the round trip times of packets with geometrically distributed service times and the following waiting times:

$$R_l^e = R_k - d + l, \; for\, l = 1, 2, \cdots, d.$$

where $l$ means the lth packet of these $d$ packets and $R_l^e$ is its equivalent waiting time.

Note that the above equivalence is consistent with the memoriless property of the geometric distribution.

Similar to the case when $k \le d$, the service time $X_l$ of packet $l$ is independent of each other and the waiting times. Thus given $R_k$, the round trip time of packet $l$ of these packets, denoted by $RTT_l^e$ for packet $l$, is independent of each other. Moreover,

$$Pr(RTT_l^e < TO | R_k = r_k) = \sum_{m=0}^{M_l} p^m q = 1 - p^{M_l+1}, \quad (19)$$

where $M_l \equiv \lceil \frac{TO-r_k-l}{d} \rceil - 1$ is the maximum allowed total number of retransmissions of the packet $l$ without incurring a timeout.

Therefore,

$$\begin{aligned} & Pr(\textstyle\bigcap_{l=1}^{k} RTT_l < TO) \\ = & Pr(\textstyle\bigcap_{l=1}^{d} RTT_l^e < TO) \\ = & \textstyle\sum_{r_k=k-1}^{TO-d-1} (\prod_{l=1}^{d} Pr(RTT_l^e < TO | R_k = r_k)) Pr(R_k = r_k) \\ = & \textstyle\sum_{r_k=k-1}^{TO-d-1} \prod_{l=1}^{d} (1 - p^{M_l+1}) Pr(R_k = r_k) \quad (20) \end{aligned}$$

Here in the third equality, we use the fact that the minimum waiting time for packet $k$ is $k-1$, which happens when the first $k-1$ packets were successfully transmitted on their first transmissions, and the maximum waiting time of packet $k$ in order for its round trip time to be less than $TO$ is $TO - d - 1$.

Similarly,

$$Pr(\bigcap_{l=1}^{W_n} RTT_l \leq z)$$
$$= \sum_{r_k=W_n-1}^{TO-1} \prod_{l=1}^{d} (1 - p^{M_{zl}+1}) Pr(R_{W_n} = r_k), \quad (21)$$

where $M_{zl} \equiv \lfloor \frac{z-r_k-l}{d} \rfloor$.

Equation 20 and 21 give us the two probabilities needed when $k > d$.

In summary, the LL queuing models give us the probability of no false timeout on the first $k$ packets of the $n$th window, $Q_{nk}^{PF}$, and the expected window cycle given no timeout, $\overline{T_{in}^{W}}$ (using equation 12 to 15). The transition probabilities and the reward functions of the system chain can be obtained from equation 4, 9 and 10. After having all the variables $P_{ij}^{T}$, $V_{ij}^{n}$ and $V_{ij}^{t}$, we can solve for the steady state distribution of the Markov chain, and the throughput of the system can be obtained using equations 1, 2 and 3.

## V. Numerical Results and Discussions

Based on the model derived in the previous section, here we present, for different protocol and packet loss parameters, the numerical solutions for the throughput of the system. For comparison purpose, when there are no random losses (i.e., $p_l = 0$) and no HL protocol is employed, the throughput of the system is also given. In this case the throughput of the system is the efficiency of the ARQ employed. For ideal GBN, this throughput is given by $\frac{1}{1+pd/q}$, and for ideal SRP, it is given by $1 - p$.

When the ARQ LL implements the GBN, Figure 3 and Figure 4 show the throughput with the change of the loss parameter $p$, $p_l$ and the ARQ link round trip time $d$. When the LL employs SRP, the curves, which we did not show here, are similar except that the corresponding throughput for each case is higher. Figure 5 and Figure 6 show the throughput with the change of the HL timeout value $TO$, when the ARQ LL implements the GBN and the SRP, respectively. In Figure 3, $p > 0$ and $p_l = 0$ represents the case where there are no random losses and the x-axis denotes the change of the link loss probability $p$. Other lines can be explained similarly. Note that the case when $p = 0$ and $p_l > 0$ is equivalent to the case when no ARQ is employed. Thus the curve corresponding to this case serves as a comparison between two systems when the ARQ is employed and is not employed.

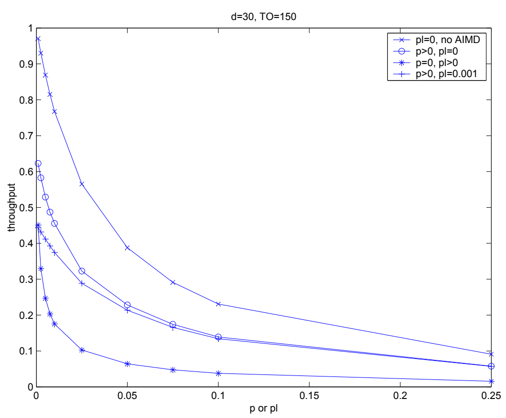


Fig. 3: Throughput vs Loss Parameter $p$ and $p_l$ - GBN

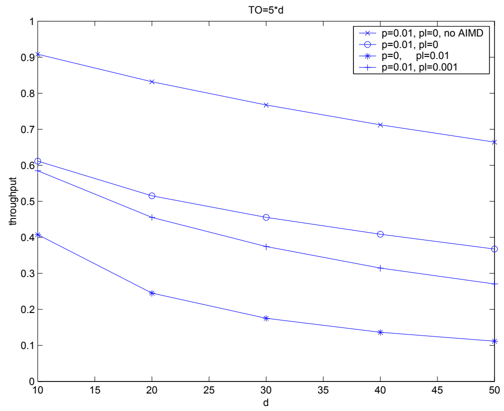


Fig. 4: Throughput vs Round Trip Time $d$ - GBN

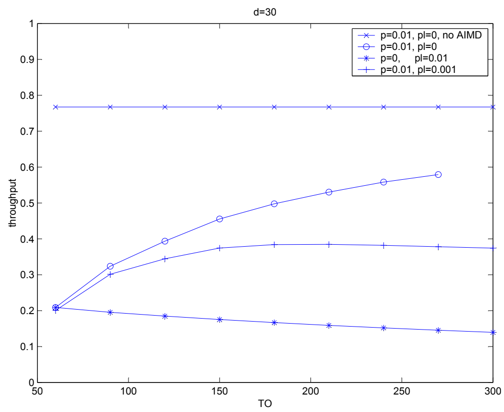


Fig. 5: Throughput vs the HL Timeout Value $TO$ - GBN

From figures 3 one can see that the throughput of the system decreases when the erroneous transmission probability $p$ or the random loss probability $p_l$ increases, regardless of whether the ARQ protocol is employed or not. This is consistent with the system operation, since larger loss probability means more retransmissions and longer delays. Notice that in this case, a system with ARQ performs much better than one without ARQ.

Figures 4 shows that the throughput of the system decreases as the round trip time $d$ increases. This is also consistent with the system operation, since longer round trip time for one transmission means that longer time is needed for both the end node HL sender and the ARQ LL sender to detect errors and start retransmissions, which leads to a lower throughput.

Figures 5 and 6 show the change of the throughput with the HL timeout value $TO$. It can be seen that when the LLs do not employ the ARQ protocol (the $p = 0$ and $p_l = 0.01$ case), the throughput of the system decreases when the timeout value of the HL increases. This is because all of the errors in this case are recovered by the HL retransmissions. The only way for the HL to detect the losses is its timeout signal. Higher timeout value makes the HL less responsive to the losses, thus gives lower throughput.

On the other hand, when the ARQ protocol is employed and there are no random losses (the $p = 0.01$ and $p_l = 0$ case),

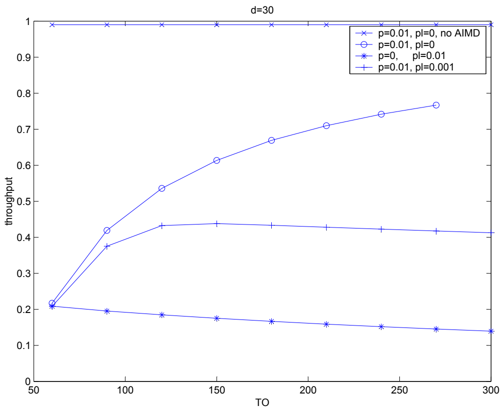


Fig. 6: Throughput vs the HL Timeout Value $TO$ - SRP

the throughput of the system increases when the timeout value of the HL increases. This is because the ARQ LL can recover all of the losses in this case, and higher HL timeout value allows more time for the LL to recover the losses and makes the unneccessary false HL timeout less likely to happen.

When both types of losses exist (the $p = 0.01$ and $p_l = 0.001$ case), the throughput of the system first increases when the timeout value of the HL increases, then decreases. This is because when the timeout value is low, although it increases, the benefit of allowing the ARQ LL to recover the losses overcome the drawback of less responsiveness of the HL, while when the timeout value is high, the opposite happens. The turning point changes with the relative value of the loss probabilities $p$ and $p_l$.

Thus, when the system includes a lossy link and/or long round trip time, if there are no random losses and all the losses can be recovered be the LL ARQ, or the random loss probability is negligible, it is better to set a higher HL timeout value to improve the throughput. But when there are random losses, one has to be cautious with the choice for the timeout value.

In the absence of random losses, all the figures show that the system not employing the AIMD HL protocol has a much higher throughput than the system with the HL protocol. This is because in this case all the losses can be recovered by the ARQ LL. But the retransmissions of the ARQ LL lead to the variability of the round trip time seen by the HL. This variability results in false timeouts at the HL, which makes the HL unnecessarily reset its window size. As a result, the throughput is decreased. This would lead one to think that when there are no random losses from which the LL ARQ cannot recover, it is better not to employ the HL protocol; however, the purpose of the HL AIMD protocol is not to recover from transmission losses, but rather to control congestion. Hence, although it appears that the AIMD protocol only serves to decrease performance, it has an important congestion control function that cannot be understood in the context of a single connection.

## VI. Conclusion

In this paper we provide an analytical framework to study the interactions between the protocols of window flow control and ARQ at different network layers. Simple queuing models are also developed for ARQ with batch arrivals. We analyze a system with two end nodes communicating over a multi-hop network whose packets incur both transmission errors and random losses. The HL of the two end nodes implements the AIMD congestion control mechanism and the LL over the error prone bottleneck link implements an ARQ protocol, where both the GBN and SRP are considered. This paper provides a Markov model for this system. Simple queuing models are used for the LL ARQ to obtain the transition probabilities and the reward functions, and the throughput of the system is derived by the theory of Markov chain with reward functions.

Numerical results for the throughput of the system show that when the loss probabilities increases, the throughput always decreases regardless of whether the ARQ protocol is employed or not. The throughput also decreases when the round trip time of the transmission increases. Moreover, increasing the HL timeout value is beneficial when there are no losses from which the LL cannot recover, that is, no random losses. But when there are random losses from which the HL must recover, increasing the timeout value beyond a certain value lowers the system throughput.

Our work so far only considered a higher layer that implements a window congestion control mechanism (AIMD). Naturally, higher layers that must also recover from random losses typically also implement some form of ARQ mechanism. A natural extension of this work, on which we hope to report in the near future, is to consider the interaction between the retransmission mechanisms at the different layers.

## References

[1] H. Balakrishnan, V.N. Padmanabhan, S.Seshan and R.H. Katz,"A comparison of mechanisms for improving TCPperformance over wireless links," ACM/IEEE Trans. on Networking, December 1997.

[2] D. Bertsekas and R. Gallager, *Data networks*, 2nd ed., Upper Saddle River, NJ, Prentice Hall, 1992.

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

[4] R.G. Gallager, *Discrete stochasic processes*, Boston/Dordrecht/London, Kluwer Academic Publishers, 1995.

[5] D. Huang and J.J. Shi, "Performance of TCP over radio link with adaptive channel coding and ARQ,"Proc. IEEE 49th Vehicular Technology Conference, vol. 3, pp. 2084 -2088, 1999.

[6] A.G. Konheim, "A queuing analysis of two ARQ protocols,"IEEE Trans. on communications, July 1980.

[7] D. Towsley and J.K. Wolf, "On the statistical analysis of queue lengths and waiting times for statistical multiplexers with ARQ retransmission schemes," IEEE Trans. on Communications, April 1979.

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

[9] M. Yoshimoto, T. Takine, Y. Takahashi and T. Hasegawa, "Waiting time and queue length distributions for go-back-N and selective-repeat ARQ protocols," IEEE Trans. on Communications, November 1993.

[10] M. Zorzi and R.R. Rao, "Error control in multi-layered stacks," Proc. IEEE GLOBECOM '97, vol. 3, pp. 1413-1418, 1997.

[11] M. Zorzi, R.R. Rao and L.B. Milstein, "ARQ error control for fading mobile radio channels," IEEE Trans. on Vehicular Technology, May 1997.