Markov-Based Analysis of End-to-End Batch Transmission in a Multi-Hop Wireless Network

Teerawat Issariyakul, Ekram Hossain, and Attahiru Sule Alfa
University of Manitoba and TRLabs
Winnipeg, MB, Canada.
Email: {teerawat, ekram, alfa}@ee.umanitoba.ca.

Abstract—We present a novel model for analyzing end-to-end transmission of a batch of packets in a multi-hop wireless network with Automatic Repeat reQuest (ARQ)-based error control mechanism implemented at each node. For a batch of packets, we derive complete statistics (in terms of probability mass function) for end-to-end latency and the number of packets successfully delivered to the destination node. The analytical model is validated by means of simulation. Typical numerical results obtained from the model reveal the trade-off between end-to-end latency and reliability which would be an important issue in design and engineering of multi-hop wireless networks. The presented analytical model would be useful in analyzing and optimizing flow control and congestion control protocols in multi-hop wireless networks such as sensor networks.

Index Terms—Multi-hop wireless networks, automatic repeat request (ARQ), discrete time Markov chain (DTMC).

I. INTRODUCTION

A successful end-to-end transmission in a multi-hop wireless network is achieved only when transmission in every hop along a communicating path is successful. If a transmitted data packet is corrupted due to non-zero packet error rate, an Automatic Repeat reQuest (ARQ)-based error recovery can be performed. Therefore, end-to-end performance (e.g., reliability, latency, energy efficiency) depends on the ARQ-policies used at the hop level.

Recently, there have been a few research works on end-to-end performance modeling and analysis in multi-hop wireless networks. An end-to-end performance model for single-packet transmission with different hop-level ARQ policies in a multi-hop wireless network was presented in [1]. Since spatial reuse of the wireless channels would allow multiple concurrent transmissions in a multi-hop scenario, an end-to-end flow/error control protocol (e.g., TCP (Transmission Control Protocol)) would presumably use a transmission window larger than one. For a multi-hop wireless network with chain topology, TCP exhibits both instability and throughput degradation when the window size is large [2], and the optimal window size was observed to be \( H/4 \), where \( H \) is the number of hops in a network path [3]. The effects of number of hops and channel error on energy efficiency and throughput performance of TCP were studied in [4]. In [5], a model for analyzing TCP performance over a two-hop network with chain topology was presented assuming error-free wireless channels and a collision-free medium access control protocol and that only one mobile can transmit at any instant. With the same topology, [6] modeled end-to-end latency under multi-rate transmission only for infinite-persistence hop-level ARQ. However, to the best of our knowledge, a general analytical framework to study the inter-relationship among link level error probability, hop-level ARQ policy, and end-to-end performance has not been reported in the literature.

In this paper, we present an analytical model to evaluate an end-to-end flow control mechanism in a multi-hop wireless network. Different levels of end-to-end reliability are achieved through different types of hop-level ARQs. For a batch of packets, we determine the probability mass function (pmf) of the end-to-end latency and the number of packets successfully delivered to the destination, and reveal the trade-off between reliability and latency for a batch transmission.

II. BACKGROUND

A. Absorbing Markov Chain

An absorbing Markov chain is a Markov process which finally stops at one of absorbing states \([7]\). Consider a discrete time Markov chain (DTMC) with first \( s_0 \) states and subsequent \( s \) states being absorbing and transient states, respectively. A general form of the corresponding transition probability matrix (TPM) \( P \), can be written as in (1) below

\[
P = \begin{pmatrix}
I & 0 \\
R & Q
\end{pmatrix}
\]  

where the matrices \( R \) and \( Q \) are called absorbing and transient TPM, respectively. Throughout this paper, we denote all-zero, all-one, and identity matrices by 0, e, and I, respectively. The pmf that the DTMC is absorbed at step \( k \) (\( f_k \)), the absorbing probability vector (\( f \)) and the expected time to absorption corresponding to each absorbing state (\( E[k] \)) can be calculated from (2)-(4), where \( \alpha_0 \) and \( \alpha \) are the probability vectors that the DTMC starts at the absorbing and transient states,

\(^1\)Throughout this paper, we use regular and boldface letters to represent scalar values and matrices, respectively.
respectively [8].

\[
\begin{align*}
f_k &= \begin{cases} 
\alpha_0, & k = 0 \\
\alpha q^{k-1} R, & k \geq 1 
\end{cases} \\
\beta &= \alpha (I - Q)^{-1} R \\
E[k] &= \alpha (I - Q)^{-1} e.
\end{align*}
\]

(2) (3) (4)

\section*{B. Methodology for the Analysis}

For a batch transmission of size \( N \) packets, we derive complete statistics in terms of pmf, cumulative probability distribution function (cdf), and expectation of the two following performance metrics. One is the number of data packets successfully delivered to the destination (\( M \)) and the other is the end-to-end latency (\( D \)) defined as the number of transmission intervals\(^2\) required for all \( N \) packets to reach the destination or to be dropped out of the network due to limited-persistence ARQ. The end-to-end latency is the duration after which there is no packet (from the tagged batch) left in the network.

We model a batch transmission process by using an absorbing DTMC, which starts with \( N \) packets at the source node and finishes when all \( N \) packets leave (either delivered or dropped) the network path. The process finishes in an absorbing state \( M \in \{0, 1, \ldots, N\} \), where \( M \) packets have been successfully delivered to the destination node. The pmf \( f_D(d) \) and the expected value \( E[D] \) of end-to-end latency and the pmf \( f_M(m) \) of the number of packets delivered to the destination can be calculated as follows:

\[
\begin{align*}
f_D(d) &= f_a \cdot e \\
f_M(m) &= f(1, m) \\
E[D] &= E[d] \cdot e
\end{align*}
\]

where \( f_a \), \( f \), and \( E[d] \) can be calculated from (2)-(4), and \( x(i, j) \) denotes the entry \( (i, j) \) of a matrix \( x \). At this point, we need to find the TPM of an absorbing DTMC representing a batch transmission process.

\section*{III. System Model and Assumptions}

Consider a static two-hop chain topology\(^3\) where the transmission range of each node is just enough to reach its neighboring nodes (Fig. 1). We assume that data transmission from node \( i \) is successful with probability \( p_i \). If the transmission fails (i.e., with probability \( 1 - p_i \)), the lost packet can be retransmitted according to one of the following ARQ policies.

- With infinite retransmission ARQ (ARQ\(^\infty\)), a packet is retransmitted repeatedly until it is successfully received by the next node in the path.
- With zero retransmission ARQ (ARQ\(^0\)), the node reports a route failure to the upper layer immediately after the transmission fails (i.e., there is no retransmission at all).
- With probabilistic retransmission (ARQ\(^P\)), a route failure is reported with probability \( \xi \) for each transmission failure.

Note that, ARQ\(^0\) and ARQ\(^\infty\) can be considered as special cases of ARQ\(^P\) (e.g., \( \xi = 1 \) for ARQ\(^0\) and \( \xi = 0 \) for ARQ\(^\infty\) [1]). After reporting the route failure, the node flushes all the packets of the corresponding batch in its transmitting buffer and stops receiving packets (corresponding to the tagged batch) from the upstream node.

We assume that the node 1 and node 3 are the source and destination nodes, respectively. We also assume that node 1 and node 2 use different channels for transmission. Therefore, transmissions from these two nodes can occur simultaneously. A batch transmission is initiated with \( N \) packets at node 1. Due to the limited transmission range, node 1 forwards these packets to node 2, which in turn will forward them to node 3. The process is considered to be complete, when no packets corresponding to the tagged batch are left in the buffers of node 1 and node 2.

\section*{IV. A Markov-Based Analytical Model}

At each transmission interval \( t \), we keep track of the number of packets (from a tagged batch) in the buffer of node \( i \) by using \( X_i^{(t)} \), \( i \in \{1, 2, 3\} \), and use an absorbing DTMC, \( Y^{(t)} = (X_1^{(t)}, X_2^{(t)}, X_3^{(t)})^T \), to model the above transmission process. The process starts with \( Y^{(0)} = (N, 0, 0) \), and finishes at interval \( t \) where \( X_1^{(t)} + X_2^{(t)} = 0 \) (absorbing states).

\subsection*{A. Infinite Retransmission ARQ (ARQ\(^\infty\))}

With ARQ\(^\infty\), data packets will never be dropped. Therefore,

\[
\sum_{i=1}^{3} X_i^{(t)} = N, \quad \forall t
\]

and the process always finishes at interval \( t \) where \( Y^{(t)} = (0, 0, N) \).

When \( X_1^{(t)} \neq 0 \) and \( X_2^{(t)} \neq 0 \), all the possible transitions from \( Y^{(t)} \) to \( Y^{(t+1)} \) and the corresponding conditional probabilities \( (q) \) are shown in Table I, where \( SS = p_1 p_2 \).

\begin{table}
\begin{tabular}{|c|c|c|}
\hline
\( q \) & \( Y^{(t+1)} \) \\
\hline
SS & \( (X_1^{(t)} - 1, X_2^{(t)}, X_3^{(t)} + 1) \) \\
SF & \( (X_1^{(t)} - 1, X_2^{(t)} + 1, X_3^{(t)}) \) \\
FS & \( (X_1^{(t)}, X_2^{(t)} - 1, X_3^{(t)} + 1) \) \\
FF & \( (X_1^{(t)}, X_2^{(t)}, X_3^{(t)}) \) \\
\hline
\end{tabular}
\caption{Possible transitions for ARQ\(^\infty\) when \( X_1^{(t)} \neq 0 \) and \( X_2^{(t)} \neq 0 \).}
\end{table}

\(^2\)A transmission interval is defined as a period required for each node to transmit a packet to its neighboring node and to determine whether the transmission has been successful or not.

\(^3\)Although we consider a two-hop network here, the following analyses can be done for a general \( H \)-hop network in a similar way.
\[ SF = p_1(1 - p_2), \quad FS = (1 - p_1)p_2, \quad FF = (1 - p_1)(1 - p_2), \]
and \( p_i \) is the successful transmission probability at hop \( i \). On the other hand, when either \( X_1^{(t)} = 0 \) or \( X_2^{(t)} = 0 \), \( Y^{(t+1)} \) can be obtained from (9) below, where \( S_1 = p_1, \quad F_1 = 1 - p_1, \quad S_2 = p_2, \) and \( F_2 = 1 - p_2 \).

\[
Y^{(t+1)} = \begin{cases} 
(0, X_2^{(t)} - 1, X_3^{(t)} + 1), & X_1^{(t)} = 0; q = S_2 \\
(0, X_2^{(t)}, X_3^{(t)}), & X_1^{(t)} = 0; q = F_2 \\
(X_1^{(t)} - 1, 1, X_3^{(t)}), & X_2^{(t)} = 0; q = S_1 \\
(X_1^{(t)}, 0, X_3^{(t)}), & X_2^{(t)} = 0; q = F_1
\end{cases}
\]

Due to (8), \( X_1 \in \{0, 1, \ldots, N\} \) and \( X_2 \in \{0, 1, \ldots, N - X_1\} \), while \( X_3(= N - X_1 - X_2) \) is absolutely dependent upon \( X_1 \) and \( X_2 \). The number of states in the DTMC process, \( Y' = (X_1, X_2, X_3) \), is therefore, \( \sum_{i=0}^{N} (N+1-i) = (N+1)(N+2)/2 \).

With ARQ\(^∞\), the general form of the TPM of the DTMC representing a batch transmission process with \( N \) packets (\( \mathbf{P}_N \)) is given in (10)-(15), where the operator \( (x, y) \) is to concatenate matrix \( x \) to the left of matrix \( y \).

\[
\mathbf{P}_N = \begin{pmatrix} 1 & 0 \\ \mathbf{R}_N & \mathbf{Q}_N \end{pmatrix}
\]

\[
\mathbf{R}_N(i, j) = \begin{cases} S_2, & (i, j) = (1, 1) \\
0, & \text{otherwise}
\end{cases}
\]

\[
\mathbf{Q}_N(i, j) = \begin{cases} \mathbf{A}'_N, & (i, j) = (1, 1) \\
\mathbf{B}_N, & (i, j) = (2, 1) \\
(0, \mathbf{B}_N^{-i+2}), & i = j + 1; i \geq 2 \\
\mathbf{A}_N^{-i+2}, & (i, j) = (2, 1)
\end{cases}
\]

\[
\mathbf{A}'_N(i, j) = \begin{cases} F_2, & i = j \\
S_2, & i = j + 1; i > 1 \\
0, & \text{otherwise}
\end{cases}
\]

\[
\mathbf{A}_N(i, j) = \begin{cases} F_1, & (i, j) = (1, 1) \\
F_2, & i = j; i > 1 \\
F_S, & i = j + 1; i > 1 \\
0, & \text{otherwise}
\end{cases}
\]

\[
\mathbf{B}_N(i, j) = \begin{cases} S_1, & (i, j) = (1, 1) \\
S_S, & i = j; i > 1 \\
0, & \text{otherwise}
\end{cases}
\]

By arranging the states in the above DTMC in an ascending order, the transmission process starts from the last row in \( \mathbf{P}_N \), where \( Y = (N, 0, 0) \), and finishes at the first column (the absorbing state), where \( Y = (0, 0, N) \). Each row/column \( i \) of \( \mathbf{Q}_N \) stands for a set of states where \( X_k = i - 1 \). If the transmission at node 1 is unsuccessful, the process will stay in the same state \( i-1 \) with TPM \( \mathbf{A}_N^{-i+2} \) or \( \mathbf{A}'_N \). On the other hand, the process will propagate towards the left with TPM \( \mathbf{B}_N^{-i+2} \), if the transmission at node 1 is successful. Due to the fact that the buffer size of node 1 monotonically decreases with increasing \( t \), \( \mathbf{Q}_N \) in (12) is a lower triangular matrix. Furthermore, the matrix form of \( \mathbf{Q}_N \) is diagonal dominant (with two non-zero diagonal rows), because the buffer size of node 1 can decrease by at most one during each transmission interval.

B. Probabilistic Retransmission ARQ (ARQ\(^P\))

With ARQ\(^P\), after each transmission failure, node \( i \) assumes that the receiver node has been turned off or moved away and a route failure is reported to the upper layer with probability \( \xi_i \). If route failure occurs at node 1, the buffer of node 1 (for a tagged batch) will be flushed and the process will continue with \( Y = (0, X_2, X_3) \). If route failure occurs at node 2, packets from node 1 and node 2 will not be able to reach node 3, and the process will finish with \( Y = (0, 0, X_3) \).

Let \( S_N \) be a state space containing all reachable states in the DTMC for ARQ\(^∞\) with \( N \) initial packets, where (8) is satisfied. The DTMC for ARQ\(^P\) starts off at the last state of \( S_N(N, 0, 0) \), and stays in this state until route failure occurs. Assuming that \( k \) packets are dropped, the process moves to \( S_{N-k} \), where \( \sum_{i=1}^{k} X_i = N - k \). Therefore, the entire state space for ARQ\(^P\) is \( \mathcal{S}_N = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \mathcal{S}_2 \cup \cdots \cup \mathcal{S}_N \).

Again, with ARQ\(^P\) the batch transmission process finishes when the buffers of node 1 and node 2 are empty, or when \( Y = (0, 0, X_3) \), \( X_2 \in \{0, 1, \ldots, N\} \). We model ARQ\(^P\) using a DTMC with \( N + 1 \) absorbing states, each of which corresponds to a certain value of \( X_3 \) when the process finishes. We arrange the states by grouping all absorbing states as first \( N + 1 \) states and concatenate them with \( \mathcal{S}'_1, \mathcal{S}'_2, \ldots, \mathcal{S}'_N \), where \( \mathcal{S}'_i = \mathcal{S}_i - (0, 0, l) \). The general form of the TPM for ARQ\(^P\) can be obtained as \( \mathbf{T}_N \) given by (16)-(21) where \( R_1 = (1 - p_1) \cdot \xi_1, R_2 = (1 - p_2) \cdot \xi_2, RS = (1 - p_1) \cdot \xi_1 \cdot p_2 \), and \( RF = (1 - p_1) \cdot \xi_1 \cdot (1 - p_2)(1 - \xi_2) \).

![Fig. 2. A scalar form of the TPM for ARQ\(^P\) with \( N = 3 \).](image-url)

\[
\mathbf{T}_N = \begin{pmatrix} \mathbf{I} & 0 \\ \mathbf{w}_N & \mathbf{\Omega}_N \end{pmatrix}
\]

\[
= \begin{pmatrix} \mathbf{U}_1 & \mathbf{Q}_1 & \cdots \\ \mathbf{U}_2 & \mathbf{V}'_{21} & \mathbf{Q}_2 & \cdots \\ \vdots & \vdots & \ddots & \ddots \\ \mathbf{U}_N & \mathbf{V}'_{N1} & \mathbf{V}'_{N2} & \cdots & \mathbf{Q}_N \end{pmatrix}
\]

\( \text{Hereafter, we drop the superscript } (t) \text{ for brevity.} \)
\[
U_n = \begin{pmatrix}
    u_n'
    
    u_n
    
    (u_{n-1}, 0)
    
    \vdots
    
    (u_2, 0)
    
    (u_1, 0)
\end{pmatrix}
\begin{pmatrix}
    (R_n, 0)
\end{pmatrix}
\]  
(18)

\[
u_n(i, j) = \begin{cases}
    R_1, & (i, j) = (1, n) \\
    RS, & (i, j) = (2, n) \\
    R_2, & i = 2, \ldots, n; j = n + 1 - i \\
    0, & \text{otherwise}
\end{cases}
\]  
(19)

\[
u_n'(i, j) = \begin{cases}
    R_2, & i = 1, \ldots, n; j = n + 1 - i \\
    0, & \text{otherwise}
\end{cases}
\]  
(20)

\[
\begin{align*}
V_{mn}(i, j) &= \begin{cases}
    \Gamma_n, & (i, j) = (m - n + 1, 1) \\
    0, & \text{otherwise}
\end{cases} \\
\Gamma_n(i, j) &= \begin{cases}
    RF, & i = j + 1 \\
    RS, & i = j + 2 \\
    0, & \text{otherwise}
\end{cases}
\end{align*}
\]  
(21)

(22)

The matrices \(R_k\) and \(Q_k\) are the absorbing and the transient TPM for ARQ\(^\infty\) with \(k\) initial packets obtained from (11) and (12), respectively. The size of each of a column vector \(R_k\) and a square matrix \(Q_k\) is \(\frac{(k+1)(k+2)}{2} - 1\). The dimensions of \(U_k\) and \(V_{kj}\) are \(\frac{(k+1)(k+2)}{2} - 1\) × \((N + 1)\) and \(\frac{(k+1)(k+2)}{2} - 1\) × \((\frac{j(j+1)}{2} - 1)\), respectively. The dimensions of each of \(u_n', u_n\), and \(\Gamma_n\) is \(n^2\). Figs. 2-3 show the TPM for ARQ\(^P\) with \(N = 3\) in scalar and matrix forms. In Fig. 2, a three-digit codes represents each row and column in Fig. 3 stand for total number of packets \(k\) in the system \(S_k\) and the value of \(X_1\), respectively.

Each row/column \(i\) of \(\Omega_N\) represents states in sub-space \(S_i'\) within which each row/column is sub-divided based on the value of \(X_1\). Again, \(\Omega_N\) is a lower-triangular matrix because total packets in the system do not increase. The process stays in the sub-space \(S_i'\) with TPM \(Q_i\), where there are \(i\) packets in the system. If none of the packets is dropped, the process will finish with TPM \(R_i\). On the other hand, if \(k\) packets are dropped but \(i - k\) packets are still in the system, the process will move from \(S_i'\) to \(S_{i-k}'\) with TPM \(V_{i,i-k}\). These transitions occur due to the route failure at node 1, after which \(X_1 = 0\). Therefore, the only possible transitions are from \(S_i'\) to \(S_i'-X_1\), and the corresponding TPM \(\Gamma_{i-X_1}\) is located only in the first sub-column of \(S_i'\) where \(X_1 = 0\). When the route failure occurs at node 2 or the failure occurs at node 1 and all other packets reach node 3, the process finishes with TPM \(U_i\). Due to the special structure of the above matrices, the complexity for computing inverse and for multiplying \(\Omega_N\) can be reduced by the algorithm provided in [9].

V. NUMERICAL AND SIMULATION RESULTS

A. Expected End-to-End Latency and Throughput

Fig. 4 plots expected end-to-end latency \(E[D]\) and number of successfully delivered packets \(M\) for ARQ\(^0\), ARQ\(^\infty\), and ARQ\(^P\) with \(\xi = 0.5\), when \(p = 0.9\) and batch size \(N\) (the number of initial packets) \(N = \{1, 2, \ldots, 10\}\). Simulation results are also shown to validate the accuracy of our model.

For ARQ\(^\infty\), \(E[D]\) is attributed only to the case where all packets reach the destination. Therefore, the minimum latency in this case is \(N + 1\) transmission intervals. The process with all other ARQs, on the other hand, can finish much sooner due to possible route failure. For example, the process with ARQ\(^0\) finishes after only one interval, if the first transmission fails. ARQ\(^\infty\) and ARQ\(^0\) provide upper bound and lower bound, respectively, for end-to-end latency of all other ARQs obtained by adjusting \(\xi\) in ARQ\(^P\). As expected, the average number of successfully delivered packets is the largest for ARQ\(^\infty\) and the smallest for ARQ\(^0\).

Next, we set \(N = 10\) and plot \(E[D]\) as a function of the successful transmission probability in each link \(p\) in Fig. 5. With ARQ\(^\infty\), increasing \(p\) decreases \(E[D]\). However, for other ARQ variants, \(E[D]\) increases because the possibility for the batch transmission process to finish earlier (due to possible route failure) decreases with increasing \(p\). When \(p = 1\), all the hop-level transmissions are successful and exactly \(N + 1\) (= 11 in this case) intervals are required (for all the ARQ variants) to complete a batch transmission.

B. Probability Mass Function of End-to-End Latency and Number of Successfully Delivered Packets

Fig. 6 plots \(cdf\) of end-to-end latency, \(F_D(d) = \sum_{i=1}^{d} f_D(i)\), for \(N = 10\) and \(p = 0.9\) for the above three ARQ protocols. We observe that for ARQ\(^\infty\), it is impossible for the process to be completed before \(N + 1\) transmissions (i.e., \(f_D(d) = 0, d \leq N\)). Therefore, ARQ\(^\infty\) provides an upper-bound of the end-to-end latency for all ARQ mechanisms.
For both ARQ0 and ARQp, the cdf starts to increase when d ≥ 1. When d ≤ N, the process can finish only because all the packets are dropped, while for d > N the process can finish because all packets are dropped and/or delivered to the destination node. Due to the small packet error rate (i.e., 1 − p ≤ 0.3), the cdf for d > N converges to 1 at a much faster rate than that it does when d ≤ N.

The cdf of ARQp falls in between those for ARQ0 and ARQ∞. For end-to-end latency of [E[D]] intervals, the probability that the batch transmission process finishes is only 52.12%. This implies that some packets are still in the network path with significant chance. Therefore, the expected end-to-end latency may not be a good parameter for end-to-end flow control, and transmission control at the source node based on the expected latency may cause network congestion.

With ARQ0, due to route failure, the probability that m packets are successfully delivered to the destination node, fM(m) is p2m(1 − p2) for m < N, i.e., the transmissions of first m packets are successful in both hops and no more packets are successfully delivered to node 3. For m = N, the transmission of all packets must again be successful for both hops, i.e., fM(N) = p2N. In this case, the term (1 − p2) vanishes, since the buffers of both nodes are empty after N packets are successfully transmitted. Obviously, fM(m) is convex in m as can be observed in Fig. 7 where N = 5. For large p, the transmission of the entire batch tends to be successful and fM(m) is strictly convex (fM(N) > fM(N − 1)) because the term 1 − p2 scales down fM(m) for m < N. When p = 1, the pmf converges to that of ARQ∞ (upper-bound) where fM(N) = 1 and fM(m) = 0, m < N.

VI. Conclusions

We have modeled and analyzed end-to-end batch transmission in a multi-hop wireless network with different types of hop-level ARQs. The complete statistics (i.e., pmf) for end-to-end latency as well as the number of packets successfully delivered to the destination node have been derived. We have verified the accuracy of our model by means of simulation. Numerical results have revealed that the expected value of the end-to-end latency is not as useful as the cdf of the end-to-end latency. In some cases, some packets are still in the network path after E[D] intervals, where E[D] is the expected end-to-end latency, with the probability as high as 50%. Therefore, end-to-end flow control based on the cdf obtained from our model would be more useful.

REFERENCES