INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. 2003; **16**:735–757 (DOI: 10.1002/dac.615)

Analysis of resequencing in downloads

Yoav Nebat† and Moshe Sidi*,‡

Electrical Engineering Department Technion, Haifa 32000, Israel

SUMMARY

Recent studies indicate that out-of-order arrival of data packets during downloads of resources is not pathological network behaviour (*ACM/IEEE Trans. Networking* 1999; **7**(6):789). Though this situation is most intuitive when packets of the same resource arrive in parallel from several sources, it turns out that this phenomenon may also occur in the single source scenario. Knowledge regarding the expected reordering needed is important both for being able to decide on the size of the resequencing buffer needed, and to estimate the burstiness in arrival of data to the application. In this study we present a method to calculate the resequencing buffer occupancy probabilities for the single source scenario, and a study of the resequencing buffer occupancy for the two source scenario, where arrival from each of the sources is in order. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: parallel download; out-of-order arrivals; resequencing buffer

1. INTRODUCTION

Text files, pictures, web pages and similar resources are often available at several sites in the network. A user that needs such a resource, say a file, may open several connections to some of the sites available and receive all the required data via the connections opened.

The simple, most common approach to obtain a resource in today's networks is to open a single connection to one of the available sources and receive all the required data via this single connection. When this method is used, the common practice is that the web site fragments the file into data packets, and starts transmitting the packets of the resource, after assigning consecutive integers to them. When a packet arrives at the destination, it is decoded and delivered to the application only if all the preceding packets have been received. Otherwise, it is stored in a resequencing buffer, waiting for all its predecessors to arrive. Though a simple model for the single connection situation may be a simple FIFO link, resulting in in-order arrivals only, this model turns out to be unrealistic. Indeed, out-of-order arrivals occur, even when a simple, single TCP connection is used [1]. This is mainly due to local parallelism in the network (i.e. multi-input switch), as well as packet losses and more global parallelism (i.e. packets being routed via different links).

*Correspondence to: Moshe Sidi, Electrical Engineering Department, Technion, Haifa 32000, Israel.
† E-mail: nyoav@tx.technion.ac.il
‡ E-mail: moshe@ee.technion.ac.il

Received 22 October 2002
Revised 12 May 2003
Accepted 14 July 2003

When connections are opened to several sources, out-of-order arrivals are much more intuitive. Regardless of the scheme used to decide on the data portions to be transmitted by every source and the transmission timing, out-of-order arrival of data is unavoidable, due to the random delay packets experience on their route to the destination.

Out-of-order arrival of data packets to the destination result in many unfavourable consequences including large resequencing buffer demands and re-ordering, resulting in an unregulated flow of data to the application at the destination [2]. When unicast TCP is used, the occurring of reordering may result in unfavourable consequences, including self-reinforcingly poor performance [1].

This paper analyses the resequencing buffer occupancy probabilities for several download scenarios. Having the resequencing buffer occupancy probabilities may allow us to determine the buffer size that may be needed in specific download scenarios. Moreover, the resequencing buffer occupancy is highly correlated to the amount of out-of-order arrivals. Therefore, it can be used to indicate how severe is the burstiness in arrival of data to the application, and the expected degradation in TCP performance due to the out-of-order arrivals.

In this paper we consider two different download models. The first is the single source case, in which a new packet is transmitted every time unit, where each packet is delayed randomly and the delay of packets correspond to an arbitrary distribution function. The second model considered is a two source scenario, where the inter-arrival time of packets from each of the sources is exponentially distributed. Other models for analysing the resequencing buffer occupancy can be found in Reference [3].

The rest of this paper is organized as follows. In Section 2 we present an analysis of the resequencing buffer occupancy, receiving data from a single source. A recursive method to derive the resequencing buffer occupancy probabilities and a closed form solution for the average buffer occupancy are introduced. Numerical results for several delay distributions are also presented. Section 3 contains the analysis of the resequencing buffer occupancy for a two data source scenario. A closed form solution for the probability not to reach an overflow, while receiving data packets of a resource, having limited resequencing buffer space is introduced. Numerical examples are also presented. Finally, Section 4 contains our summary and future work.

2. RESEQUENCING BUFFER OCCUPANCY, RECEIVING DATA FROM A SINGLE SOURCE

2.1. The model

In many networking situations data packets sent by a single source are assigned consecutive integers before being sent to their destination. The receiver at the destination delivers packets to the application only in sequence, thus, a received packet is delivered only if all its preceding packets have been received. Otherwise, the received packet is stored in a resequencing buffer, waiting for all its preceding packets to arrive, and only then it is delivered to the application. Whenever a packet is delivered, the next packet in sequence is delivered immediately if it is already stored in the resequencing buffer.

The system considered consists of a source, sending a new packet every time unit. Packets are assigned consecutive integers. Every sent packet experiences a random delay. We assume that for each packet sent arrival is possible every Δ time units and the maximal possible delay is $K \cdot \Delta$ time units, i.e. the delay of each packet is $d \cdot \Delta$, $(1 \leqslant d \leqslant K)$. The probability distribution function of d is $F_d(i)$, i.e. $p(d \leqslant i) = F_d(i)$, independent of the delay of other packets.

Due to the random delay, packets may arrive at the destination in a different order than they were sent. Consequently, the occupancy of the resequencing buffer will be random. Our goal is to determine the occupancy probabilities for this buffer. Having the probability distribution of the buffer occupancy enables provision of the necessary buffer space at the receiver to keep potential overflows under a specific threshold.

2.2. *Resequencing buffer occupancy for* $\Delta = 1$

To facilitate the presentation, we begin the analysis with $\Delta = 1$. Without loss of generality we can assume that the integer assigned to every sent packet is the time it was sent, since a new packet is sent every time unit.

We denote the probability of having m packets in the resequencing buffer at time t by $P^t(m)$, and the steady-state probability of having m packets in the resequencing buffer by $P(m)$.

Definition
The minimal valued packet at time t (mvp^t) is the packet that was assigned the lowest integer among the packets that have not arrived at the destination's receiver till time t [4].

Definition
D_m is the elapsed time since the mvp at time t (mvp^t) was sent, i.e. $D_m = t - \text{mvp}^t$, where D_m is updated every time unit.

From the definition of mvp^t we notice that none of the packets sent before mvp^t are stored in the resequencing buffer. Noticing that, since the maximum delay any packet may experience is K, mvp^t is one of the $K-1$ most recently sent packets or t, if all packets sent before time t arrive till time t.

From the above we conclude:

(1) The maximum possible occupancy of the resequencing buffer is $K-2$, since all packets stored in the resequencing buffer were sent after the mvp.
(2) The occupancy of the resequencing buffer at time t depends only on the delay of the $K-1$ most recently sent packets, and since these are independent identically distributed (i.i.d.), its probability distribution is the same for any t given that there were at least $K-1$ packets sent before time t. Therefore, if we assume, without loss of generality, that the first packet is sent at time 0, the buffer reaches its steady-state at time $K-1$ and the steady state probability to have m packets in the resequencing buffer, $P(m)$, is equal to the probability to have buffer occupancy m at time t, $P^t(m)$, for any $t \geqslant K-1$.
(3) Since $\text{mvp}^t \geqslant t - K + 1$ and $D_m = t - \text{mvp}^t$, we have: $D_m \leqslant K-1$.

The probability that $D_m = k$ is the same for any $t \geqslant K-1$ since D_m is determined by mvp^t which depends only on the delays of the $K-1$ most recently sent packets, which are i.i.d. Therefore, we can define Q_k to be the probability that $D_m = k$ in steady-state, i.e. Q_k is the probability that $D_m = k$ for $t \geqslant K-1$. The probability that $D_m = k$ for $k \leqslant K-1$, $t \geqslant K-1$ is the probability that the mvp at time t is $t-k$, since $\text{mvp}^t = t - D_m$. Therefore, $D_m = k$ if and only if all packets sent before time $t-k$ arrive at the destination's receiver till time t (this occurs with probability $\prod_{j=k+1}^{K} F_d(j)$), and the packet sent at time $t-k$ does not arrive till time t (this occurs with probability $R_d(k) \equiv 1 - F_d(k)$). Therefore, for $0 \leqslant k \leqslant K-1, t \geqslant K-1$

$$Q_k = Pr(D_m = k) = R_d(k) \prod_{j=k+1}^{K} F_d(j) \tag{1}$$

In order to calculate $P(m)$ for $m = 0, 1, 2, \ldots, K-2$ we shall calculate $P^t(m \mid D_m = k)$ for $t \geqslant K-1$ and use the *total probability theorem* for unconditioning.

We begin our calculation by calculating $P^t(0 \mid D_m = k),\ 0 \leqslant k \leqslant K-1$.

For $k = 0$ or $k = 1$ there are no packets that were sent after mvp^t. Therefore,

$$P^t(0 \mid D_m = 0) = P^t(0 \mid D_m = 1) = 1 \tag{2}$$

for $2 \leqslant k \leqslant K-1$ we have

$$P^t(0 \mid D_m = k) = \prod_{j=1}^{k-1} R_d(j) \tag{3}$$

Only packets sent after mvp^t may be stored in the resequencing buffer, consequently

$$P^t(m \mid D_m = k) = 0 \quad k \leqslant m \leqslant K-2 \tag{4}$$

Using the probabilities above in the following theorem we can derive $P^t(m \mid D_m = k)$ for $0 < m \leqslant K-2$, $t \geqslant K-1$.

Theorem 2.1

The probability of having m packets in the resequencing buffer at time t, $t \geqslant K-1$, given that $D_m = k$, can be calculated recursively for $1 \leqslant m < k$ using

$$P^t(m \mid D_m = k) = F_d(k-1)P^t(m-1 \mid D_m = k-1) + R_d(k-1)P^t(m \mid D_m = k-1) \tag{5}$$

Proof

The probability of having m packets in the resequencing buffer at time $t, t \geqslant K-1$, given that D_m is k is the probability that exactly m of packets $t-k+1, t-k+2, \ldots, t-1$ arrive at the destination's receiver before time t. Therefore $P^t(m \mid D_m = k)$ is simply the probability of receiving exactly m of the $k-1$ most recently sent packets.

The probability of receiving exactly m of the $k-1$ most recently sent packets is equal to the probability of receiving exactly $m-1$ of the $k-2$ most recently sent packets (i.e. $P^t(m-1 \mid D_m = k-1)$), and receiving packet $t-k+1$ before time t (occurs with probability $F_d(k-1)$) plus the probability of receiving exactly m of the $k-2$ most recently sent packets (i.e. $P^t(m \mid D_m = k-1)$), and not receiving packet $t-k+1$ before time t (occurs with probability $R_d(k-1)$). From the above, (5) results immediately, and the proof is completed. □

In order to calculate $P(0)$ we use the *total probability theorem* to uncondition $P^t(0 \mid D_m = k)$. Using (1)–(3) for all possible values of k $(0 \leqslant k \leqslant K-1)$ we have

$$P(0) = \sum_{k=0}^{K-1} P^t(0 \mid D_m = k)Q_k = Q_0 + Q_1 + \sum_{k=2}^{K-1} \left[\left[\prod_{j=k+1}^{K} F_d(j) \right] \prod_{j=1}^{k} R_d(j) \right]$$

Similarly, we calculate $P(m)$, $1 \leqslant m \leqslant K-2$ using (1), (4) and (5) for all $m+1 \leqslant k \leqslant K-1$ (since $m \leqslant k-1$) we have

$$P(m) = \sum_{k=m+1}^{K-1} P^t(m \mid D_m = k) Q_k = \sum_{k=m+1}^{K-1} \Big[[F_d(k-1) P^t(m-1 \mid D_m = k-1) + R_d(k-1) P^t(m \mid D_m = k-1)] R_d(k) \prod_{j=k+1}^{K} F_d(j) \Big] \quad (6)$$

2.3. Resequencing buffer occupancy for $\Delta > 1$

It may be useful to have a solution for the resequencing buffer occupancy probability when $\Delta > 1$. This situation may represent a model of a single source, sending packets with a constant round-trip time and a positive probability of being erroneous. If the probability of error is independent of the number of times the packet was sent, d is geometrically distributed and can be approximated by a distribution with a finite support, though the exact distribution is infinite. Anyway, in case special measures are taken upon retransmitting, the probability distribution of d may be of any general type.

A similar recursion can be used to calculate the resequencing buffer occupancy for $\Delta \geqslant 2$.

Given the probability distribution function of d, $F_d(i)$, and Δ, we could use the same method we used to derive the resequencing buffer occupancy probability for $\Delta = 1$ if we substitute $F_d(i)$ by $G_d(i)$ where

$$G_d(i) = F_d\left(\left\lfloor \frac{i}{\Delta} \right\rfloor \right) \quad 0 \leqslant i \leqslant K \cdot \Delta$$

Yet, due to the different nature of the problem, we bring the direct calculation for the probability distribution of the buffer occupancy for this situation.

Now, the probability that a packet sent k time units before t will not arrive at the destination's receiver till time t is $R_d(\lfloor k/\Delta \rfloor)$. Thus, for $\Delta - 1 \leqslant k \leqslant K \cdot \Delta - 1$, (1) becomes

$$Q_k = P(D_m = k) = R_d\left(\left\lfloor \frac{k}{\Delta} \right\rfloor \right) \prod_{j=k+1}^{\Delta \cdot K - 1} F_d\left(\left\lfloor \frac{j}{\Delta} \right\rfloor \right) \quad (7)$$

The maximum buffer occupancy, given that $D_m = k$ is $k - \Delta$ since the packets that may be received till time t are: $t-k+1, t-k+2, \ldots, t-\Delta$. The maximum for D_m is $K \cdot \Delta - 1$, thus, the maximum possible buffer occupancy is $\Delta(K-1) - 1$. For similar reasons to those mentioned for the case $\Delta = 1$, steady-state is reached at time $t = K \cdot \Delta - 1$.

Now, $P^t(0 \mid D_m = k) = 1$ for $k = \Delta - 1$ and for $k = \Delta$ since no packets that were sent after the mvp could possibly arrive at the destination's receiver till time t. For $m = 0$, $\Delta < k \leqslant K \cdot \Delta - 1$ we now have

$$P^t(0 \mid D_m = k) = \prod_{j=\Delta}^{k-1} R_d\left(\left\lfloor \frac{j}{\Delta} \right\rfloor \right) \quad (8)$$

We also have $P^t(m \mid D_m = k) = 0$ for $m > k - \Delta$, since only packets sent after mvp^t and before time $t - \Delta + 1$ may be stored in the resequencing buffer. For $0 < m \leqslant k - \Delta$, (5) becomes

$$P^t(m \mid D_m = k) = F_d\left(\left\lfloor \frac{k-1}{\Delta} \right\rfloor \right) P^t(m-1 \mid D_m = k-1) + R_d\left(\left\lfloor \frac{k-1}{\Delta} \right\rfloor \right) P^t(m \mid D_m = k-1) \quad (9)$$

Using the probabilities above in (9) we can derive $P^t(m \mid D_m = k)$ for $1 \leqslant m \leqslant \Delta(K-1) - 1$, $t \geqslant K \cdot \Delta - 1$. In order to calculate $P(m)$ we calculate $P(0)$ using (7) and (8) in the *total probability theorem* where $\Delta - 1 \leqslant k \leqslant K \cdot \Delta - 1$

$$P(0) = \sum_{k=\Delta-1}^{K\cdot\Delta-1} P^t(0 \mid D_m = k) Q_k$$

and then calculate $P(m)$ for $1 \leqslant m \leqslant \Delta(K-1) - 1$ recursively, using (7) and(9), where $m + \Delta \leqslant k \leqslant K \cdot \Delta - 1$ (since $m \leqslant k - \Delta$)

$$P(m) = \sum_{k=\Delta+m}^{K\cdot\Delta-1} P^t(m \mid D_m = k) Q_k$$

2.4. The average buffer occupancy

Using (5) we can derive the average buffer occupancy. Denoting $F_d(k-1)$ by q_k and $P^t(m \mid D_m = k)$ by $P(m \mid k)$, for $1 \leqslant m < k$ where $k \leqslant K - 1$, (5) becomes

$$P(m \mid k) = q_k P(m-1 \mid k-1) + (1 - q_k) P(m \mid k-1)$$

The conditional expected value of m given k, $2 \leqslant k \leqslant K - 1$, is

$$\begin{aligned} E(m \mid k) &= \sum_{m=1}^{k-1} m P(m \mid k) = \sum_{m=1}^{k-1} m[q_k P(m-1 \mid k-1) + (1 - q_k) P(m \mid k-1)] \\ &= q_k \sum_{m=1}^{k-1} (m-1) P(m-1 \mid k-1) + q_k \sum_{m=1}^{k-1} P(m-1 \mid k-1) \\ &\quad + \sum_{m=1}^{k-1} m P(m \mid k-1) - q_k \sum_{m=1}^{k-1} m P(m \mid k-1) \\ &= q_k E(m \mid k-1) + q_k + E(m \mid k-1) - q_k E(m \mid k-1) = q_k + E(m \mid k-1) \end{aligned}$$

Using the above recursively, starting with $E(m \mid 1) = 0$ we have

$$E(m \mid 2) = F_d(1)$$

$$E(m \mid 3) = F_d(1) + F_d(2)$$

thus, for k, $2 \leqslant k \leqslant K - 1$

$$E(m \mid k) = \sum_{i=1}^{k-1} F_d(i) \tag{10}$$

Consequently, the average resequencing buffer occupancy is given by

$$E(m) = E(E(m \mid k)) = \sum_{k=0}^{K-1} Q_k E(m \mid k) = \sum_{k=2}^{K-1} R_d(k) \left[\prod_{j=k+1}^{K} F_d(j) \right] \sum_{i=1}^{k-1} F_d(i) \tag{11}$$

For instance, if we assume a bi-modal delay model where

$$F_d(i) = \begin{cases} p_1 & 1 \leqslant i \leqslant K - 1 \\ 1 & i = K \end{cases} \tag{12}$$

then using (11), after some algebra, we obtain (see Appendix A for details)

$$E(m) = p_1(K-1) - \frac{p_1 - p_1^K}{1 - p_1}$$

Similarly, for a uniform delay model

$$F_d(i) = \frac{i}{K} \quad 0 \leqslant i \leqslant K$$

we obtain (see Appendix B for details)

$$E(m) = \frac{K!}{2 \cdot K^{K+2}} \sum_{k=2}^{K-1} \frac{K^k(K-k)}{(k-2)!}$$

2.5. Numerical results

Figures 1–3 illustrate the resequencing buffer occupancy probabilities for several delay models.

Consider a transmission scenario where every transmitted packet has a probability p_1 to arrive within a time unit, and if the packet does not arrive in a single time unit, it is guaranteed to arrive after exactly K time units (e.g. QoS guarantee). This scenario is modelled by a bi-modal delay distribution as in (12).

Figure 1 shows the resequencing buffer occupancy probabilities for bi-modal delay distributions, with different values of p_1 (in all cases $p_{100} = 1 - p_1$). As p_1 gets closer to 1 the probability of having no packets stored in the resequencing buffer increases, and the probability distribution, given that the occupancy is not 0, becomes close to uniform. We can explain this phenomenon

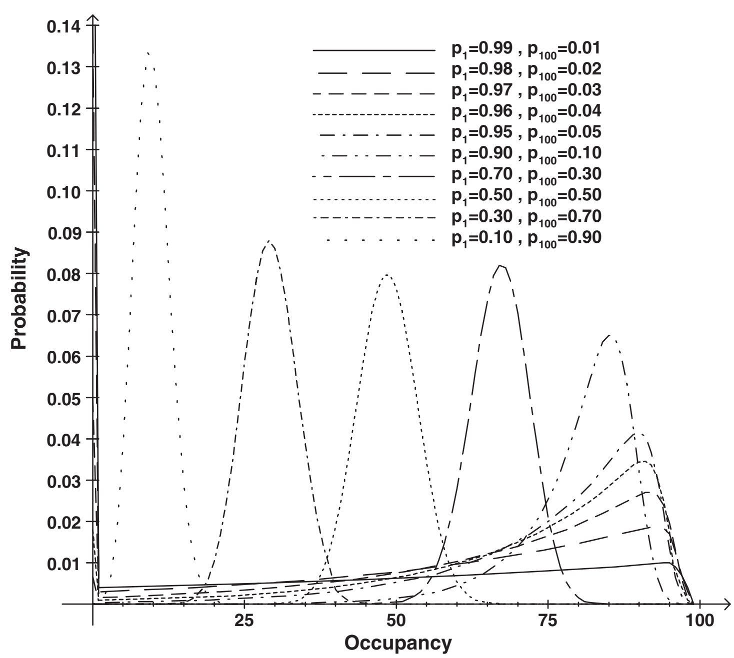


Figure 1. Resequencing buffer occupancy distributions for bi-modal delay models.

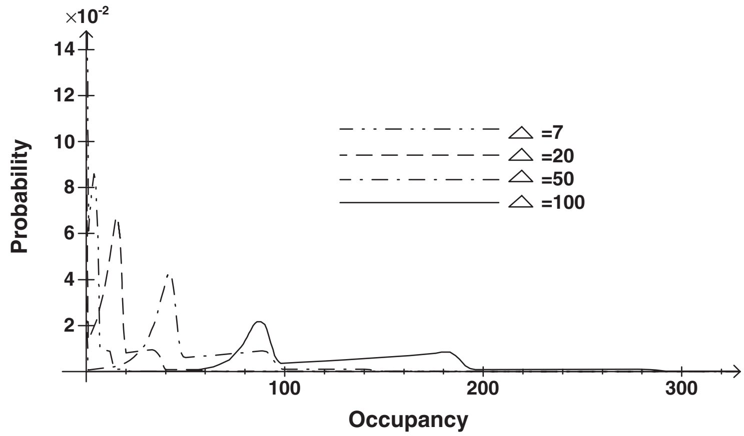


Figure 2. Resequencing buffer occupancy distributions for geometrical delay models, for several values of Δ.

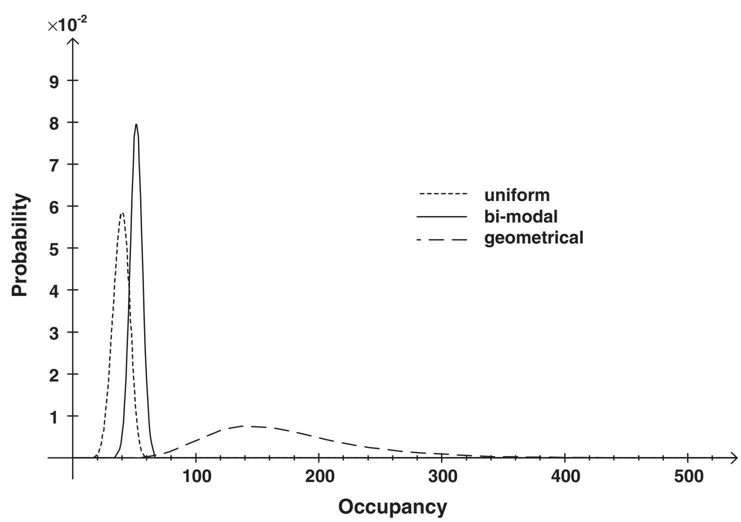


Figure 3. Resequencing buffer occupancy distributions for uniform, bi-modal and geometrical delay models.

through investigating the delay of the mvp. The delay of the mvp in all cases is 100 time units, unless all 100 last transmitted packets experience a delay of 1 time unit. When p_1 is not very close to 1 the probability that all 100 last transmitted packets experience a delay of 1 time unit is negligible, thus, the delay of the mvp is 100 time units almost surely. Recalling that packets stored in the resequencing buffer are the packets that were transmitted after the mvp, and have already arrived, we get that for the bi-modal case packets stored in the resequencing buffer are the packets that were transmitted after the mvp that experienced a delay of 1 time unit. The number of packets stored in the resequencing buffer, thus, increases as p_1 increases, as Figure 1 indicates for $p_1 = 0.1, 0.3, 0.5, 0.7, 0.9$.

However, as p_1 gets closer to 1, the probability that all 100 last transmitted packets experienced a delay of 1 time unit increases, which results in an increase in the probability of having no packets stored in the resequencing buffer. Moreover, as p_1 gets closer to 1, if not all 100 last transmitted packets experienced a delay of 1 time unit, it becomes more and more likely that only one of the 100 last transmitted packets experiences a delay of 100 time units. The placing of this packet among the 100 last transmitted packets is equi-probable, and since all packets transmitted after this packet experienced a delay of 1 time unit, they are the packets stored in the resequencing buffer, and the occupancy, given only one of the 100 last transmitted packets experiences a delay of 100 time units is uniform.

Now consider a download scenario where the round-trip time is approximately constant and packets have a certain probability of drop and error. Packets may be retransmitted upon reaching a time-out in the sending side. This scenario may be modeled by a delay distribution where arrival of any sent packet is possible every Δ time units. If no special measures are taken upon retransmit, each transmission has the same probability of being successful, and the delay of packets will be geometrically distributed.

Note that the buffer occupancy for the geometrically distributed delay could not be calculated directly from the analysis, since the analysis assumes a finite support. In order to calculate the occupancy probabilities for a geometrically distributed delay, we approximated the delay distribution by a distribution with a large finite support.

Figure 2 shows the resequencing buffer occupancy probabilities for several Δ and geometrical delay distributions (in all cases the probability of receiving a packet successfully is $p = 0.9$). We notice that the resequencing buffer occupancy probabilities are periodically decreasing, with period Δ. This can be explained by the fact that in order to have the occupancy exceed Δ, at least one of the packets not received must fail in arriving twice, which occurs with probability $(1 - p)^2 = 0.01$ for each packet. In order to have the occupancy exceed $2 \cdot \Delta$, at least one of the packets not received must fail in arriving three times, which occurs with probability $(1 - p)^3 = 0.001$ for each packet.

We also notice that the resequencing buffer occupancy probabilities in the first period increase and then decrease close to the end of the period, while this phenomenon slowly disappears in the next periods. We shall focus on the case $\Delta = 100$, although a similar explanation may be used for any other Δ. If the resequencing buffer occupancy is in the range $[0, 100)$, it is most likely that the mvp is a packet that failed to arrive exactly once, therefore it is a packet transmitted between 199 and 100 time units ago. The average number of packets among these 100 potential packets that fail to arrive is 10, therefore, it is most likely that the mvp is one of the first transmitted among these 100 potential packets. The packets stored in the resequencing buffer are the packets that are among the packets transmitted after the mvp, and till 100 time units ago, that did not fail to arrive. The average number of these packets is 0.9 of the packets transmitted in the mentioned time interval, therefore, the occupancy is most likely to be slightly under 90.

On the other hand if we consider, for instance, the case where the resequencing buffer occupancy is in the range $[200, 300)$, the mvp is most likely a packet that failed to arrive exactly three times, and since this event is rare, the average number of packets that failed in arriving exactly three times, given that there exists at least one, is not much greater than 1, therefore the placing of the mvp among the 100 potential packets is almost equi-probable, resulting in an almost uniform conditional distribution for the resequencing buffer occupancy.

Figure 3 shows the resequencing buffer occupancy probabilities for several cases of delay distributions (in all cases the average delay used is 50 time units).

In the figure we notice that the resequencing buffer occupancy when packets' delay is geometrically distributed is very likely to be much greater than when packets experience bi-modal or uniformly distributed delays. This occurs since the delay of every packet, when the delay is geometrically distributed, is not limited therefore the delay of the mvp is very likely to be large, and since all packets transmitted after the mvp may potentially be stored in the resequencing buffer, the buffer-occupancy is likely to be large. When the delay of packets is limited, as in bi-modal and uniform delay distributions, the delay of the mvp is limited, and so is the resequencing buffer occupancy, since only packets transmitted after the mvp may be stored in the resequencing buffer.

While investigating the resequencing buffer occupancy for different delay distributions, we checked if there may be any correlation between the variance of the delay and the average resequencing buffer occupancy. When we checked the resequencing buffer occupancy for a heavy-tailed delay distribution, where $p(d = i) \propto 1/i^3$, where there exists a finite average delay, but the variance is infinite, we found that the resequencing buffer has an infinite average. This correlation between the variance of the delay and the average resequencing buffer occupancy can explain the difference in the resequencing buffer occupancy distributions for the bi-modal and uniform delay models, since the variance of the bi-modal distribution is far greater than the variance for the uniform distribution, therefore it is reasonable to expect the average resequencing buffer occupancy to be greater.

3. RESEQUENCING BUFFER OCCUPANCY: TWO DATA SOURCES

3.1. The model

One of the main problems dealing with the situation of receiving data segments of the same file from different sources simultaneously is that in order to forward the data to the application in sequence, the receiver at the destination must store packets received out of sequence in a resequencing buffer until all their preceding packets are received. When data is received from several sources, packets may be received out of sequence, even if packets received from any single source are received in the same sequence they were sent.

Our goal is to determine the probability distribution of the number of packets received before the resequencing buffer reaches a certain threshold.

The system considered consists of two sources, sending packets towards a destination. One source sends the odd indexed packets (i.e. $1, 3, 5, \ldots$), the other source sends the even indexed packets (i.e. $2, 4, 6, \ldots$). We assume that packets from each source are received in order and the inter-arrival time of any two packets received from the same source is exponentially distributed (note that orderly arrival of packets from each source implies that the actual delay cannot be independent and identically distributed). The odd and the even indexed packets are sent with rates λ_1 and λ_2 packets per second, respectively. Therefore, the mean inter-arrival time of odd and even indexed packets is $1/\lambda_1$ and $1/\lambda_2$ seconds, respectively. Each packet received is delivered to the application if all its preceding packets were received. Otherwise, it is stored in a resequencing buffer.

Due to the random delay, the resequencing buffer occupancy is random. We wish to determine the probability distribution of the number of received packets until the resequencing buffer reaches occupancy K. Having this probability distribution enables the provision of the necessary buffer size for receiving files of a specified magnitude, keeping potential overflows under a specified threshold.

3.2. *The average number of packets received till the buffer reaches a threshold*

Packets are stored in the resequencing buffer only if one or more of their preceding packets were not received. In order to examine the buffer occupancy we shall use the definition of the *minimal valued packet* at time t, mvp^t, which is the packet that was assigned the lowest index among the packets that have not arrived at the destination's receiver till time t. We notice that by definition all the packets preceding mvp^t were received by time t, and none of the packets sent by the source of mvp^t after mvp^t were received till time t, since packets sent by each source are received in the same order they were sent. Consequently, none of the packets sent by the sender of mvp^t are stored in the resequencing buffer. In other words, if mvp^t is an odd indexed packet, no odd indexed packet is stored in the resequencing buffer, and if mvp^t is an even indexed packet, no even indexed packet is stored in the resequencing buffer.

Lemma 3.1
The couple: Resequencing buffer occupancy at time t and the source of mvp^t establish a continuous time Markov-chain.

Proof
We denote the state of having x packets in the resequencing buffer and mvp^t being odd indexed by (x, odd), and the state of having x packets in the resequencing buffer and mvp^t being even indexed by (x, even).

We shall show first that the knowledge of the current state and the source of the next packet to be received is all that is needed to determine the next state of the system. To conclude the proof we will show that the source of the next packet to be received is independent of the system's past.

Without loss of generality, we assume that mvp^{t_1} is odd indexed, and its index is i. Let the next arrival time be t_2. There are four possible situations:

(a) If the resequencing buffer occupancy is 0 and the next packet received is odd indexed then the packet received is indexed i, and the index of mvp^{t_2} is $i+1$, thus the new state will be $(0, \text{even})$.
(b) If the resequencing buffer occupancy is k, $k \neq 0$ and the next packet received is odd indexed then the packet received is indexed i, and the index of mvp^{t_2} is $i+2$, since the packet indexed $i+1$ is surely available in the resequencing buffer, thus, the new state will be $(k-1, \text{odd})$.
(c) If the resequencing buffer occupancy is 0 and the next packet received is even indexed then the packet received is indexed $i+1$, and the index of mvp^{t_2} is i, thus the new state will be $(1, \text{odd})$.
(d) If the resequencing buffer occupancy is $k, k \neq 0$ and the next packet received is even indexed then the packet received is indexed $i+2k+1$, and the index of mvp^{t_2} is i, thus the new state will be $(k+1, \text{odd})$.

All the above will still hold if we substitute 'odd' and 'even'.

Remembering that transitions occur only upon arrivals, the probabilities of arrival of an odd and an even indexed packet are $\lambda_1/(\lambda_1+\lambda_2)$ and $\lambda_2/(\lambda_1+\lambda_2)$, respectively, independent of the current state and the system's past, since the inter-arrival times are exponentially distributed. Consequently, The couple: resequencing buffer occupancy at time t and the source of mvp^t establish a continuous time Markov-chain. □

Therefore, our system can be represented by the Markov-chain in Figure 4. The probability p represents an arrival of an odd indexed packet (this occurs with probability $\lambda_1/(\lambda_1+\lambda_2)$). The probability q represents an arrival of an even indexed packet (this occurs with probability $\lambda_2/(\lambda_1+\lambda_2)$).

In order to calculate the average number of packets received till the buffer occupancy reaches K we assume the initial state is $(0, \text{odd})$, since there are no packets stored in the resequencing buffer, and the index of mvp^0 is 1. Since every state change in the Markov-chain in Figure 4 represents a packet arrival and *vice versa*, the average number of packets received till the buffer occupancy reaches K is the average number of state transitions in the Markov-chain of Figure 4, beginning at state $(0, \text{odd})$ until reaching either state (K, odd) or state (K, even).

To facilitate the presentation of the calculation we shall use the finite Markov-chain in Figure 5. State (i), $0 \leqslant i \leqslant K$ in the Markov-chain in Figure 5 represents state $(K-i, \text{odd})$ in the Markov-chain in Figure 4; State (i), $K+1 \leqslant i \leqslant 2K+1$ in the Markov-chain in Figure 5 represents state $(i-(K+1), \text{even})$ in the Markov-chain in Figure 4. Therefore, in order to calculate the average number of packets received till the resequencing buffer occupancy reaches K packets we shall calculate the average number of state transitions in the Markov-chain in Figure 5, beginning at state (K), until reaching either state (0) or state $(2K+1)$.

We denote the average number of state transitions in the Markov-chain in Figure 5, beginning with state (i), till reaching state (0) or state $(2K+1)$ by n_i. Consequently,

$$n_i = \begin{cases} 1 + p \cdot n_{i+1} + q \cdot n_{i-1} & 1 \leqslant i \leqslant 2K \\ 0 & i = 0,\ 2K+1 \end{cases} \tag{13}$$

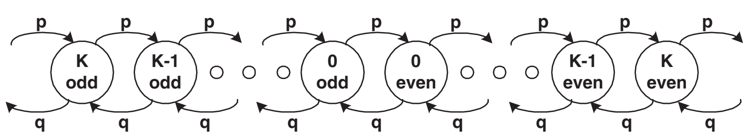


Figure 4. The Markov-chain representing the system.

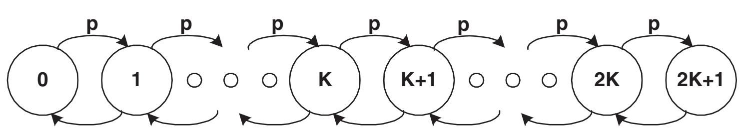


Figure 5. The finite, modified Markov-chain representing the system.

The generating function of n_i is given by

$$N(z) = \sum_{i=0}^{2K+1} n_i z^i = \sum_{i=1}^{2K} z^i + p\sum_{i=1}^{2K} n_{i+1} z^i + q\sum_{i=1}^{2K} n_{i-1} z^i \tag{14}$$

After some algebra (see Appendix C for details), (14) becomes

$$\begin{aligned} N(z) &= \frac{A(z)}{q(1-z)^2(z-p/q)} \\ &= \frac{z[z(1-z^{2K}) - (1-z)(qn_{2K}z^{2K+1} + pn_1)]}{q(1-z)^2(z-p/q)} \end{aligned} \tag{15}$$

In (15) the quantities n_1 and n_{2K} are still unknown. To determine them we exploit the fact that $N(z)$ is an analytic function for all z. Through examination of the poles of $N(z)$ we conclude that

$$A'(z)|_{z=1} = 0 \tag{16}$$

and, for $p \neq q$

$$A(z)|_{z=\frac{p}{q}} = 0 \tag{17}$$

for $p = q$

$$A''(z)|_{z=1} = 0 \tag{18}$$

If we calculate the first derivative of $A(z)$ for $z = 1$, using (16) we get

$$n_{2K} = \frac{2K - pn_1}{q} \tag{19}$$

And, for $p \neq q$, substituting (19) into (17) we get

$$n_1 = \frac{p(1-p/q)/q - 2K(p/q)^{2K+1}(1-p/q)}{p(1-p/q)(1-(p/q)^{2K+1})} \tag{20}$$

n_K, which is equal to the average number of packets received till the resequencing buffer occupancy reaches K, can now be calculated recursively using (13). For $K \gg 1$ and $q > p$ (20) becomes

$$n_1 \simeq \frac{1}{q-p} \tag{21}$$

and, by inserting (21) into (13), and calculating n_K recursively we get

$$n_K \simeq \frac{K}{q-p} \quad K \gg 1,\ q > p \tag{22}$$

Similarly, for $K \gg 1$, $q < p$ we get

$$n_K \simeq \frac{K+1}{p-q} \quad K \gg 1,\ q < p \tag{23}$$

If we calculate the second derivative of $A(z)$ for $z = 1$, using (18), for $p = q$ we get

$$n_1 = 2K \tag{24}$$

and, by substituting (24) into (13), and calculating n_K recursively we get

$$n_K = K(K+1) \quad q = p \tag{25}$$

From the above we conclude that if the two sources send their packets at the same rate, the average number of packets that will be received till a resequencing-buffer that can accommodate K packets is full is $K(K+1)$. Consequently, the average number of packets that will be received till a packet is lost (this occurs when a packet that should be stored in the resequencing buffer is received and there are K packets already stored in the resequencing buffer) is $(K+1)(K+2)$, which corresponds to the average number of packets that will be received till a resequencing buffer that can accommodate $K+1$ packets is full.

3.3. The probability distribution of the number of packets received before reaching buffer occupancy K

We denote the probability of receiving n packets without reaching an occupancy of K packets in the resequencing buffer by $P_K(n)$. Also, we denote the number of odd and even indexed packets among the first i packets received by the destination's receiver by o_i and e_i, respectively, where $i = o_i + e_i$.

The probability of receiving n packets before reaching an occupancy of K packets in the resequencing buffer, given that exactly e of the n packets received are even indexed is given by

$$P_K(n \mid e_n = e) = prob(-K < o_m - e_m < K+1 \;\; for \;\; m = 1, 2, \ldots, n \mid e_n = e) \tag{26}$$

The equality in (26) holds since the resequencing buffer occupancy will reach K packets during the receipt of n packets if and only if there exists m, $m \leqslant n$, such that the number of the odd indexed packets among the first m packets received exceeds the number of the even indexed packets by $K+1$ (this corresponds to moving from state (K) to state $(2K+1)$ in the Markov-chain of Figure 5), or if the number of even indexed packets among the first m packets received exceeds the number of the odd indexed packets by K (this corresponds to moving from state (K) to state (0) in the Markov-chain of figure 5).

In order to calculate the probability in (26) we use a known solution for a ballot problem, shown in Reference [5]:

In a ballot, candidate A scores a votes and candidate B scores b votes, and all possible voting records are equally probable. Let $c - d < b - a < c$ where $0 < c < d$ where a, b, c and d are integers. If we denote by α_r and β_r the number of votes registered for A and B, respectively, among the first r votes recorded, the equality below holds

$$\text{prob}(c - d < \beta_r - \alpha_r < c \;\; for \;\; r = 1, 2, \ldots, a+b \mid \alpha_{a+b} = a, \beta_{a+b} = b)$$

$$= \frac{1}{\binom{a+b}{a}} \sum_{k=0,\pm1,\pm2,\ldots} \left[\binom{a+b}{a-kd} - \binom{a+b}{a+c+kd} \right] \tag{27}$$

Since our system is Markov, all packet receiving records are equally probable, and thus if we substitute a, b, c and d in (27) by $e, n-e, K+1$ and $2K+1$, respectively, (26) becomes

$$P_K(n \mid e_n = e) = \frac{1}{\binom{n}{e}} \cdot \sum_{j=0,\pm1,\pm2,\ldots} \left[\binom{n}{e - j(2K+1)} - \binom{n}{e + K + 1 + j(2K+1)} \right] \tag{28}$$

Now, in order to calculate $P_K(n)$, we use the *total probability theorem* for unconditioning. Therefore

$$P_K(n) = \sum_{e \in A} P_K(n \mid e_n = e) \cdot prob(e_n = e) \tag{29}$$

where A is found through substituting a, b, c and d in the condition $c - d < b - a < c$ by $e, n - e, K + 1$ and $2K + 1$, respectively. Therefore, A is given by

$$A = \{e\colon\ -K < n - 2e < K + 1,\ e \text{ integer}\}$$

from the above we conclude that

$$A = \left\{e\colon \frac{n - K - 1}{2} < e < \frac{n + K}{2},\ e \text{ integer}\right\}$$

The probability of receiving exactly e even indexed packets among the first n packets received is simply

$$\text{prob}(e_n = e) = \binom{n}{e} p^{n-e} q^e \tag{30}$$

Consequently, substituting (30) and (28) into (29) we conclude that

$$P_K(n) = \sum_{e=\left\lceil \frac{n-K-1}{2} \right\rceil}^{\left\lfloor \frac{n+K}{2} \right\rfloor} p^{n-e} q^e \cdot \sum_{j=0,\pm 1,\pm 2,\ldots} \left[\binom{n}{e - j(2K+1)} - \binom{n}{e + K + 1 + j(2K+1)}\right] \tag{31}$$

which can be exploited, according to the following lemma, to determine the required buffer size to keep overflows under a specified threshold, while receiving packets of a resource of a specific magnitude.

Lemma 3.2

The probability of receiving all the packets of a file that consists of N_t packets, using a buffer that can accommodate K packets and the transmission scheme described in Section 3.1, is equal to $P_K(N_t - K - 1)$.

The proof of the Lemma appears in Appendix D.

3.4. Numerical results

Figures 6–8 illustrate the relationship between the resequencing buffer size, the number of packets to be received and the overflow probability.

Figure 6 shows the probability of not reaching overflows while receiving up to 2000 packets from two sources that transmit packets at the same rate (i.e. $\lambda_1 = \lambda_2$) for resequencing buffer sizes 20, 50 and 100. Recall from (25) that the expected value for the number of packets to be received till overflowing a buffer that can accommodate K packets is $(K + 1)(K + 2)$. For $K = 20$, 50 and 100 that is 462, 2652 and 10 302 packets on the average, respectively. As

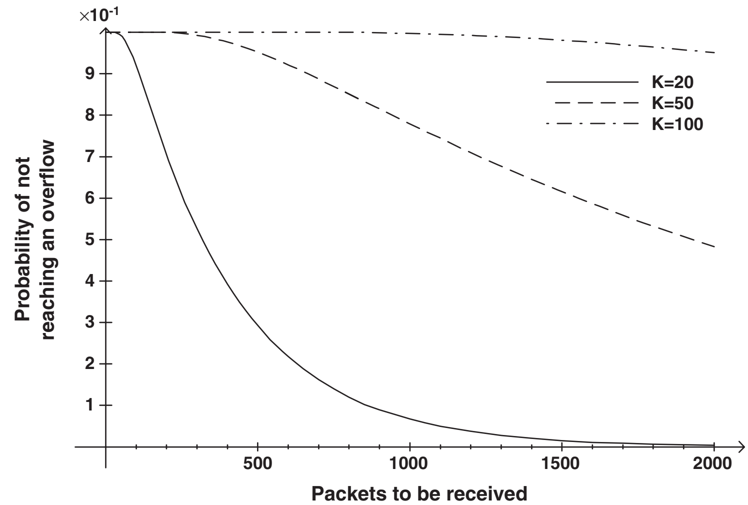


Figure 6. Probability of not reaching a buffer overflow *vs* number of packets, for several buffer sizes.

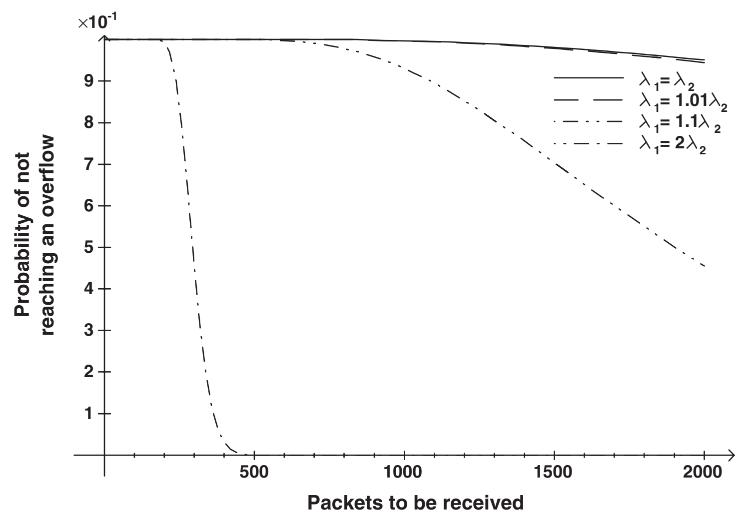


Figure 7. Probability of not reaching a buffer overflow *vs* number of packets, for several transmission rates; buffer size 100.

expected, we can see that provisioning a 20 packet buffer is likely to result in many packet drops while provisioning a 100 packet buffer is very likely to be adequate.

Figure 7 shows the probability of not reaching overflows while receiving up to 2000 packets from two sources that transmit packets with different rates for a resequencing buffer size of 100. As expected when the rates differ significantly, buffer overflows are expected after receipt of a relatively small number of packets. Note that even when the rates differ significantly, the

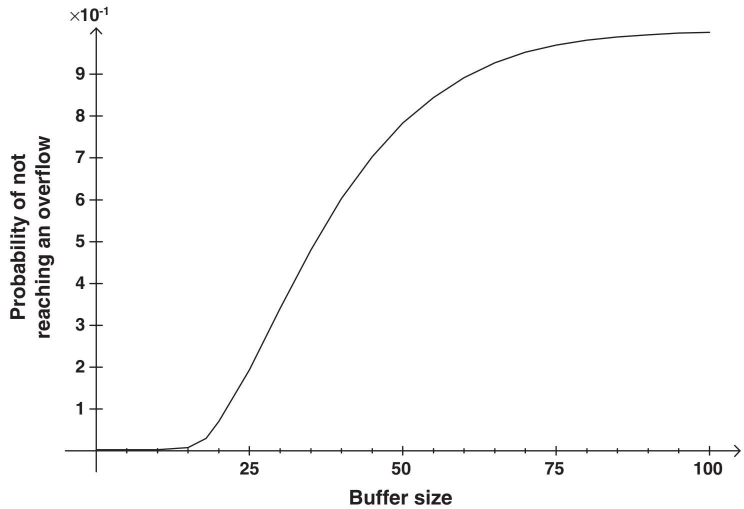


Figure 8. Probability of not reaching a buffer overflow while receiving 1000 packets *vs* resequencing buffer size.

probability of not reaching an overflow is similar to the case where the transmission rates are equal till some point, where a threshold effect occurs. This threshold occurs at different file sizes, depending on the different rate ratios.

Figure 8 shows the probability of not reaching an overflow while receiving 1000 packets from two sources that transmit packets at the same rate (i.e. $\lambda_1 = \lambda_2$) for resequencing buffer sizes up to 100. If we receive files that are not over 1000 packets long, the required buffer size for any desired threshold can be obtained from this figure.

4. SUMMARY AND FUTURE WORK

In this paper the resequencing buffer occupancy was analysed for two different downloading models.

In the first a single source is assumed and a new packet is transmitted every time unit, where each packet is delayed randomly and the delay of packets is i.i.d. and correspond to an arbitrary distribution function. A closed form expression for calculating the average buffer occupancy and a set of recursive equations for calculating the resequencing buffer occupancy distribution were introduced. This model and the derived solutions may help predict the severity of out-of-order when the round-trip time is constant and packet losses occur, as well as when the delay is simply random and corresponds to an arbitrary delay function.

The second model considered is a two source scenario, where the inter-arrival time of packets from each of the sources is exponentially distributed. This model assumes no control is sent to the servers. In this scenario the buffer demands grow as the downloaded resource gets larger. Closed form expressions for calculating the distribution and average of the number of packets to be received before a resequencing buffer of a specific magnitude reaches a threshold were introduced.

For the single source scenario it may be interesting to define a model that does not assume the delay of packets is i.i.d. and analyse the expected out-of-order. It may also be useful to have an analysis with the constant packet rate assumption omitted.

It may be useful to have an analysis of the out-of-order when several servers are used for downloading different parts of a resource in parallel. In Reference [6] the delay of the packets from each source was modelled and the resequencing buffer occupancy distribution when more than one server is used to download different parts of the same resource in parallel was analysed. Algorithms that assign each packet to a source responsible for its transmission, to achieve better performance were defined and analysed as well.

APPENDIX A

The reliability function for the given bi-modal distribution is

$$R_d(i) = \begin{cases} p_K & 1 \leqslant i \leqslant K-1 \\ 0 & i = K \end{cases}$$

where $p_K = 1 - p_1$. Using (11), the average resequencing buffer occupancy is given by

$$E(m) = \sum_{k-2}^{K-1} p_K p_1^{K-k-1}(k-1)p_1 = p_K p_1^K \sum_{k=2}^{K-1} (k-1)p_1^{-k}$$

If we denote p_1^{-1} by w this becomes

$$E(m) = p_K p_1^K w^2 \sum_{k=1}^{K-2} kw^{k-1} = (1-p_1)p_1^{K-2} \frac{\partial}{\partial w} \frac{w - w^{K-1}}{1-w}$$

$$= \frac{p_1(K-1)(1-p_1) + p_1^K - p_1}{1-p_1} = p_1(K-1) - \frac{p_1 - p_1^K}{1-p_1}$$

as in Section 2.4.

APPENDIX B

The reliability function for the given uniform distribution is

$$R_d(i) = 1 - \frac{i}{K} \quad 0 \leqslant i \leqslant K$$

Using (11), the average resequencing buffer occupancy is given by

$$E(m) = \sum_{k=2}^{K-1}\left(1-\frac{k}{K}\right)\left[\prod_{j=k+1}^{K}\frac{j}{K}\right]\sum_{i=1}^{k-1}\frac{i}{K}$$

$$= \sum_{k=2}^{K-1}\left(1-\frac{k}{K}\right)\frac{K!}{k!K^{K-k}}\frac{k(k-1)}{2K}$$

$$= \frac{K!}{2\cdot K^{K+2}}\sum_{k=2}^{K-1}\frac{K^k(K-k)}{(k-2)!}$$

as in Section 2.4

APPENDIX C

The generating function of n_i is given in (14):

$$N(z) = \sum_{i=1}^{2K} z^i + p\sum_{i=1}^{2K} n_{i+1}z^i + q\sum_{i=1}^{2K} n_{i-1}z^i$$

$$= \frac{z - z^{2K+1}}{1-z} + pz^{-1}\sum_{i=2}^{2K+1} n_i z^i + qz\sum_{i=1}^{2K-1} n_i z^i$$

$$= \frac{z - z^{2K+1}}{1-z} + pz^{-1}[N(z) - n_1 z] + qz[N(z) - n_{2K}z^{2K}]$$

Consequently, $N(z)$ is given by

$$N(z) = \frac{(z - z^{2K+1})/(1-z) - pn_1 - qn_{2K}z^{2K+1}}{1 - pz^{-1} - qz}$$

$$= \frac{z(1-z^{2K}) - (1-z)(pn_1 + qn_{2K}z^{2K+1})}{(1-z)(1-pz^{-1}-qz)}$$

$$= \frac{-\frac{z}{q}[z(1-z^{2K}) - (1-z)(pn_1 + qn_{2K}z^{2K+1})]}{(1-z)\left(z^2 - \frac{z}{q} + \frac{p}{q}\right)}$$

$$= \frac{-\frac{z}{q}[z(1-z^{2K}) - (1-z)(pn_1 + qn_{2K}z^{2K+1})]}{(1-z)(z-1)\left(z - \frac{p}{q}\right)}$$

$$= \frac{z[z(1-z^{2K}) - (1-z)(qn_{2K}z^{2K+1} + pn_1)]}{q(1-z)^2\left(z - \frac{p}{q}\right)}$$

as in (15).

APPENDIX D

Proof of Lemma 3.2

Denote the probability of receiving a packets of a file that consists of T packets, without overflowing a buffer that can accommodate K packets by $P_{K,T}(a)$. Lemma 3.2 now takes the form

$$P_{K,\,N_t}(N_t) = P_K(N_t - K - 1) \tag{D1}$$

$P_{K,\,N_t}(N_t)$ can be written as

$$P_{K,\,N_t}(N_t) = P_{K,\,N_t}(N_t \,|\, N_t - K - 1) P_{K,\,N_t}(N_t - K - 1) \tag{D2}$$

The proof consists of two parts:

Part I: $P_{K,\,N_t}(N_t \,|\, N_t - K - 1) = 1$
Part II: $P_{K,\,N_t}(N_t - K - 1) = P_K(N_t - K - 1)$

Denote the resequencing buffer occupancy upon the ith arrival by B_i. It is not difficult to verify that the following equality holds (proofs for similar equalities can be found in Reference [6])

$$B_i = \begin{cases} o_i - e_i - 1 & o_i > e_i \\ e_i - o_i & o_i \leqslant e_i \end{cases} \tag{D3}$$

Proof of Part I

Assume an overflow occurs upon the lth arrival, $l \geqslant N_t - K$. In this case either $o_l - e_l - 1 > K$ or $e_l - o_l > K$. Assume N_t is even. We clearly have $o_l \leqslant N_t/2$ and $e_l \leqslant N_t/2$. Assume $o_l - e_l - 1 > K$ holds we have

$$\frac{N_t}{2} \geqslant o_l > K + e_l + 1 = K + (l - o_l) + 1 \geqslant K + \left(l - \frac{N_t}{2}\right) + 1 \geqslant K + \left(N_t - K - \frac{N_t}{2}\right) + 1 = \frac{N_t}{2} + 1$$

which contradicts our assumption. If we assume $e_l - o_l > K$ holds we have

$$\frac{N_t}{2} \geqslant e_l > K + o_l = K + (l - e_l) \geqslant K + \left(l - \frac{N_t}{2}\right) \geqslant K + \left(N_t - K - \frac{N_t}{2}\right) = \frac{N_t}{2}$$

which, again, contradicts our assumption. Now, assume N_t is odd, in this case we clearly have $o_l \leqslant (N_t + 1)/2$ and $e_l \leqslant (N_t - 1)/2$. Assume $o_l - e_l - 1 > K$ holds we have

$$\begin{aligned} \frac{N_t + 1}{2} \geqslant o_l > K + e_l + 1 &= K + (l - o_l) + 1 \geqslant K + \left(l - \frac{N_t + 1}{2}\right) + 1 \\ &\geqslant K + \left(N_t - K - \frac{N_t + 1}{2}\right) + 1 = \frac{N_t + 1}{2} \end{aligned}$$

which contradicts our assumption. If we assume $e_l - o_l > K$ holds we have

$$\frac{N_t - 1}{2} \geqslant e_l > K + o_l = K + (l - e_l) \geqslant K + \left(l - \frac{N_t - 1}{2}\right) \geqslant K + \left(N_t - K - \frac{N_t - 1}{2}\right) = \frac{N_t + 1}{2}$$

which, again, contradicts our assumption. Therefore, given that no overflows where reached till $N_t - K - 1$ packets are received, no overflows may be reached till all packets are received, since an overflow may not occur for no $l, l \geqslant N_t - K$. This completes the proof of Part I. □

Proof of Part II:
Denote the index of the ith arriving packet by r_i. Let $\bar{r} = (r_1, r_2, \ldots, r_{N_t-K-1})$ be an arrival sequence for the first $N_t - K - 1$ packet arrivals. Let Z be the set of all arrival sequences that do not result in any buffer overflows and are feasible for the infinite arrival case. Let Y be the set of all arrival sequences that do not result in any buffer overflows and are feasible for the finite arrival case. $P_K(N_t - K - 1)$ is given by

$$P_K(N_t - K - 1) = \sum_{\bar{r} \in Z} p^{(i)}(\bar{r})$$

where $p^{(i)}(\bar{r})$ is the probability that the arrival sequence $\bar{r}$ will occur in the infinite arrival case. Similarly, $P_{K,N_t}(N_t - K - 1)$ is

$$P_{K,N_t}(N_t - K - 1) = \sum_{\bar{r} \in Y} p^{(f)}(\bar{r})$$

where $p^{(f)}(\bar{r})$ is the probability that the arrival sequence $\bar{r}$ will occur in the finite arrival case. We will first prove that $\bar{r} \in Y$ if and only if $\bar{r} \in Z$. Then we will prove that $p^{(f)}(\bar{r}) = p^{(i)}(\bar{r})$, for all $\bar{r} \in Z$. Clearly, every arrival sequence that is feasible for the finite case is also feasible for the infinite case, since all packet indexes are possible. We now show that every arrival sequence that is feasible for the infinite case, and does not result in any buffer overflows is also feasible for the finite case. Let $o_{\bar{r}}$ and $e_{\bar{r}}$ be the total number of odd and even indexed packets in $\bar{r}$, respectively. If N_t is even, $\bar{r}$ will be a feasible arrival sequence for the finite case for any feasible arrival sequence in the infinite case, as long as $o_{\bar{r}} \leqslant N_t/2$ and $e_{\bar{r}} \leqslant N_t/2$, since in the finite case only $N_t/2$ odd and only $N_t/2$ even indexed packets are sent. Assume $\bar{r}$ is not feasible for the finite case (i.e. either $o_{\bar{r}} \geqslant N_t/2 + 1$ or $e_{\bar{r}} \geqslant N_t/2 + 1$). Assume, first, $o_{\bar{r}} \geqslant N_t/2 + 1$. In this case the resequencing buffer occupancy will be

$$o_{\bar{r}} - e_{\bar{r}} - 1 \geqslant \frac{N_t}{2} + 1 - \left[N_t - K - 1 - \left(\frac{N_t}{2} + 1\right)\right] - 1 = K + 2$$

in contradiction to the fact that the arrival sequence $\bar{r}$ does not result in a buffer overflow.

If, on the other hand, we assume $e_{\bar{r}} \geqslant N_t/2 + 1$, the resequencing buffer occupancy will be

$$e_{\bar{r}} - o_{\bar{r}} \geqslant \frac{N_t}{2} + 1 - \left[N_t - K - 1 - \left(\frac{N_t}{2} + 1\right)\right] = K + 3$$

again, contradicting the fact that the arrival sequence $\bar{r}$ does not result in a buffer overflow. Now, If N_t is odd, $\bar{r}$ will be a feasible arrival sequence for the finite case for any feasible arrival sequence in the infinite case, as long as $o_{\bar{r}} \leqslant (N_t + 1)/2$ and $e_{\bar{r}} \leqslant (N_t - 1)/2$, since in the finite case only $(N_t + 1)/2$ odd and only $(N_t - 1)/2$ even indexed packets are sent. Assume $\bar{r}$ is not feasible for the finite case (i.e. either $o_{\bar{r}} \geqslant (N_t + 1)/2 + 1$ or $e_{\bar{r}} \geqslant (N_t - 1)/2 + 1$). Assume, first, $o_{\bar{r}} \geqslant (N_t + 1)/2 + 1$. In this case the resequencing buffer occupancy will be

$$o_{\bar{r}} - e_{\bar{r}} - 1 \geqslant \frac{N_t + 1}{2} + 1 - \left[N_t - K - 1 - \left(\frac{N_t + 1}{2} + 1\right)\right] - 1 = K + 3$$

in contradiction to the fact that the arrival sequence $\bar{r}$ does not result in a buffer overflow.

If, on the other hand, we assume $e_{\bar{r}} \geqslant (N_t - 1)/2 + 1$, the resequencing buffer occupancy will be

$$e_{\bar{r}} - o_{\bar{r}} \geqslant \frac{N_t - 1}{2} + 1 - \left[N_t - K - 1 - \left(\frac{N_t - 1}{2} + 1\right)\right] = K + 2$$

again, contradicting the fact that the arrival sequence $\bar{r}$ does not result in a buffer overflow. From the above we conclude that $\bar{r} \in Y$ if and only if $\bar{r} \in Z$. Note that since an exponential inter-arrival time is assumed for odd and for even indexed packets, as long as such packets are still expected to arrive, the probability of an odd arrival is $\lambda_1/(\lambda_1 + \lambda_2)$, until all the odd indexed packets arrive. Similarly, the probability of an even arrival is $\lambda_2/(\lambda_1 + \lambda_2)$, until all the even indexed packets arrive. Therefore the probability of an arrival sequence $\bar{r}$ in the finite case may differ from the probability of the same arrival sequence in the infinite case only if during the arrival process an odd or even arrival becomes impossible (i.e. all such packets have already arrived). Assume N_t is even. If during the arrival sequence $\bar{r}$ an odd arrival becomes impossible then, for this sequence, we will surely have $o_{N_t-K-2} = N_t/2$. Anyway, this is impossible since the resequencing buffer occupancy, right after arrival $N_t - K - 2$ would be $o_{N_t-K-2} - e_{N_t-K-2} - 1 = N_t/2 - (N_t/2 - K - 2) - 1 = K + 1$, contradicting the fact that $\bar{r}$ is an arrival sequence that does not result in a buffer overflow. Similarly, if during the arrival sequence $\bar{r}$ an even arrival becomes impossible, for this sequence we will surely have $e_{N_t-K-2} = N_t/2$. Anyway, this is impossible since the resequencing buffer occupancy, right after arrival $N_t - K - 2$ would be $e_{N_t-K-2} - o_{N_t-K-2} = N_t/2 - (N_t/2 - K - 2) = K + 2$, contradicting the fact that $\bar{r}$ is an arrival sequence that does not result in a buffer overflow. Now, assume N_t is odd. If during the arrival sequence $\bar{r}$ an odd arrival becomes impossible then, for this sequence, we will surely have $o_{N_t-K-2} = (N_t + 1)/2$. Anyway, this is impossible since the resequencing buffer occupancy, right after arrival $N_t - K - 2$ would be $o_{N_t-K-2} - e_{N_t-K-2} - 1 = (N_t + 1)/2 - ((N_t - 1)/2 - K - 2) - 1 = K + 2$, contradicting the fact that $\bar{r}$ is an arrival sequence that does not result in a buffer overflow. Similarly, if during the arrival sequence $\bar{r}$ an even arrival becomes impossible, for this sequence we will surely have $e_{N_t-K-2} = (N_t - 1)/2$. Anyway, this is impossible since the resequencing buffer occupancy, right after arrival $N_t - K - 2$ would be $e_{N_t-K-2} - o_{N_t-K-2} = (N_t - 1)/2 - ((N_t + 1)/2 - K - 2) = K + 1$, contradicting the fact that $\bar{r}$ is an arrival sequence that does not result in a buffer overflow. From the above we conclude that the probability of any arrival sequence $\bar{r} \in Z$ in the infinite case is equal to its probability in the finite case. Combining this with the fact that $\bar{r} \in Y$ if and only if $\bar{r} \in Z$, the proof of Part II is completed. □

From Parts I, II and (D2) the proof of Lemma 3.2 is completed instantly, since

$$\begin{aligned} P_{K,N_t}(N_t) &= P_{K,N_t}(N_t \mid N_t - K - 1) P_{K,N_t}(N_t - K - 1) = 1 \cdot P_K(N_t - K - 1) \\ &= P_K(N_t - K - 1) \end{aligned}$$

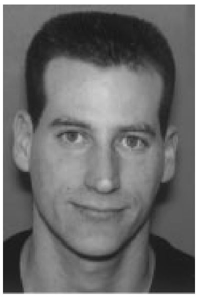
□

REFERENCES

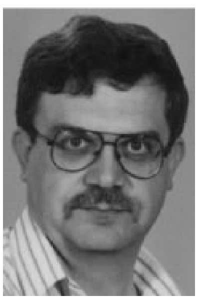
1. Bennet JCR, Partridge C, Shectman N. Packet reordering is not pathological network behavior. *ACM/IEEE Transactions on Networking* 1999; **7**(6):789–798.
2. Jean-Marie A, Tidball M, Escalante M, Leoni V, Ponce de Leon H. On the influence of resequencing on the regularity of service. *Performance Evaluation Journal* 1999; **36–37**:115–135.
3. Leung KC, Li VOK. A resequencing model for high speed networks. *IEEE ICC'99*. Vancouver, Canada, 1999; 1239–1243.

4. Rosberg Z, Shacham N. Resequencing delay and buffer occupancy under the selective-repeat ARQ. *IEEE Transactions on Information Theory* 1989; **35**(1):166–173.
5. Lajos Takacs. *Combinatorial Methods in the Theory of Stochastic Processes*. Wiley: New York, 1967.
6. Nebat Y, Sidi M. Resequencing considerations in parallel downloads. *Proceedings of Infocom 2002*. New York, 2002.

AUTHORS' BIOGRAPHIES



Yoav Nebat received the BSc and MSc (both summa cum laude) degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1995 and 2002, respectively, both in electrical engineering. Mr. Nebat is currently pursuing the PhD degree in electrical engineering at the University of California, San Diego. His research interests are in multi-user communication systems, signal processing and wireless communications.



Moshe Sidi received the BSc, MSc and the DSc degrees from the Technion-Israel Institute of Technology, Haifa, Israel, in 1975, 1979 and 1982, respectively, all in electrical engineering. In 1982 he joined the faculty of Electrical Engineering Department at the Technion where he is currently a Professor holding the Technion Chair for Electrical Engineering. During the academic year 1983–1984 he was a Post-Doctoral Associate at the Electrical Engineering and Computer Science Department at the Massachusetts Institute of Technology, Cambridge, MA. During 1986–1987 he was a visiting scientist at IBM, Thomas J. Watson Research Center, Yorktown Heights, NY. He has coauthored the book 'Multiple Access Protocols: Performance and Analysis,' Springer-Verlag, 1990.

He served as the Editor for Communication Networks in the IEEE Transactions on Communications from 1989 until 1993, as the Associate Editor for Communication Networks and Computer Networks in the IEEE Transactions on Information Theory from 1991 until 1994, as an Editor in the IEEE/ACM Transactions on Networking from 1993 until 1997, and as an Editor in the Wireless Journal 1993–2001. He also served as the General Chair for Infocom 2000.

His research interests are in wireless networks and multiple access protocols, traffic characterization and guaranteed grade of service in high-speed networks, queuing modeling and performance evaluation of computer communication networks.

