JOURNAL OF INFORMATION SCIENCE AND ENGINEERING XX, XXX-XXX (2013)

Analysis of Reliable Broadcast of Data Buffer Window over Wireless Networks

MUSTAFA B. MUTLUOĞLU AND ŞEBNEM BAYDERE
Department of Computer Engineering, Yeditepe University
Kayışdağı, İstanbul 34755 - TURKEY
E-mail: {mmutlu, sbaydere}@cse.yeditepe.edu.tr

Reliable transfer of a data buffer window over lossy wireless channels may require many retransmissions. Due to high power consumption of radio hardware and delay constraints of real time applications, wireless link protocols operating at resource poor environments aim to minimize the number of these retransmissions. In this paper, we consider reliable broadcast of the full buffer in a window to multiple recipients over wireless channels. We present a generic sender initiated negative acknowledgement based joint retransmission scheme and its mathematical analysis in terms of the total number of packets transmitted in a reliable transfer session. In the analysis, we assume that the communication channels have independent packet loss probabilities. Hence, some packets may successfully be received by only a subset of the receivers. The analysis concludes with a generalized formula derived as a function of **number of receivers** (k) and **buffer size** (N). The formula, which is also verified with simulations, can be useful as a basis for the theoretical analysis of joint retransmission based cooperative Automatic-Request-Reply (ARQ) protocols.

Keywords: Broadcast, reliability, wireless networks, ARQ, mathematical analysis.

1. INTRODUCTION

In wireless networks, power consumption due to the radio hardware is considerably high. For the applications which require bulk data transfer from sink to multiple receivers, Automatic-Request-Reply (ARQ) error recovery algorithms aim to decrease the number of retransmissions. There have been some studies on designing energy conserving variants of the classical Go-Back-N and Selective Repeat Automatic-Request-Reply mechanisms for wireless communication [1]. Among these protocols, Selective Repeat is a better choice, especially when used with loss indication, i.e. negative acknowledgments (NACKs) for resource poor environments [2]. However, in the analysis of these schemes it has been assumed that the protocols operate at peer-to-peer links. Ouyang et.al. [3] studied single sender multiple receivers case for end-to-end reliable multicast in multi hop tree networks. Koutsonikolas compared ARQ and Forward-Error-Correction (FEC) performance in [4]. Further, Koutsonikolas and Hu analyzed the design space of reliable multicast for wireless mesh networks in [5]. Kuri et.al. analyzed reliable multicast over hybrid networks [6]. The analysis of cooperative retransmission schemes for resource poor environments have been

.

studied in [7] and [8]. Predojev et.al. gives an analysis of the energy efficiency of cooperative ARQ strategies in low-power networks in [9]. Analysis of these strategies in the presence of hidden and exposed terminals is given in [10] by Alonso-Zarate et.al. To this end, theoretical models, based on the assumptions presented in this paper, utilizing the sender protocol for multiple receivers in a broadcast medium have not been studied before.

In the scope of this study, we present a sender initiated joint retransmission protocol with its mathematical analysis for reliable transfer of *N* packets to *k* distinct receivers. The analysis exploits the characteristics of the wireless broadcast medium to determine the number of retransmissions required in the case of mutual losses at the receivers. As a result of the analysis, we present a general formula which gives the expected number of packets to be transmitted in a transfer session. The formula is also validated against simulation results.

We show that the presented method significantly reduces the total number of messages that have to be sent during a reliable transfer session; which result in a significant decrease in the power consumption. The derived formula can be used to determine the topology of the networks to be constructed; like determining the number of nodes on each hop of a network.

The structure of the paper is organized as follows: In Section 2, a generic retransmission scheme is explained. Section 3 contains the mathematical analysis of the scheme, followed by the simulation results in Section 4. We conclude and state the future work in Section 5.

2. RETRANSMISSION SCHEME

In the generic protocol, the sender broadcasts a stream of *N* data packets to *k* receivers in its transmission range. The retransmission scheme is receiver-initiated i.e., sender retransmits a packet upon request from one or more receivers. The state transition diagram of the transmission model is illustrated in **Fig. 1**.

Before explaining the sender and receiver algorithms, we present the legend in the form of vectors and matrices as follows:

s is the bit-vector of *to-be-sent* packets maintained by the sender, denoted as $\mathbf{s} = \langle s_0 \quad s_1 \quad \ldots \quad s_k \rangle$ where initially $\forall i \left(s_i = 1 \right)$. A "*1*" in this vector means there are still receivers who haven't successfully received the corresponding packet.

The bit-vectors of *received* packets maintained by the receivers will be named as $\mathbf{r_i}$ for i = *1* to *k* (for *k* receivers). A "*1*" in vector $\mathbf{r_i}$ means that the corresponding packet is not yet received by the receiver i. Receiver vectors are denoted as $\mathbf{r}_i = \langle r_{i_0} \quad r_{i_1} \quad \ldots \quad r_{i_n} \rangle$ where initially $\forall ij \left(r_{ij} = 1 \right) where\ i = 1, k;\ j = 1, n$.

The retransmission matrix $\mathbf{R}$ which is denoted as,

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \cdots & & & \\ r_{k1} & r_{k2} & \cdots & r_{kn} \end{bmatrix}$$

is the matrix formed by the sender upon receiving the request vectors of the receivers at the end of each retransmission phase. In the following subsection we explain the sender and receiver algorithms.

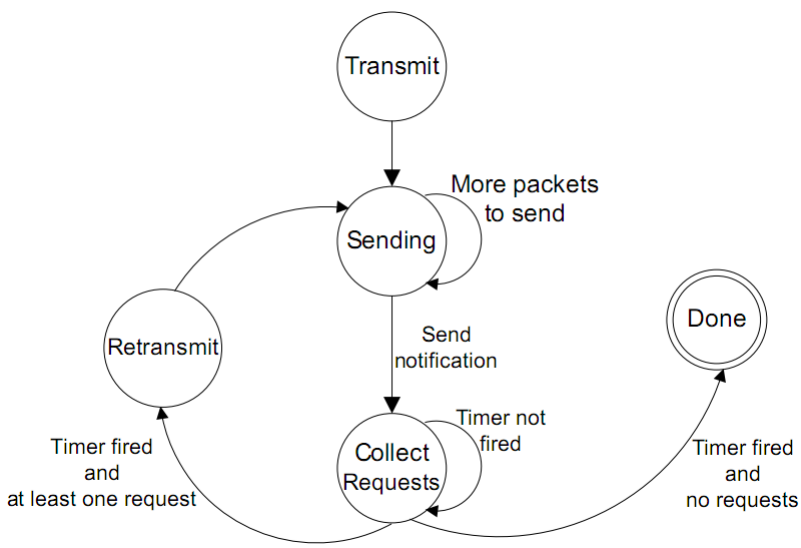

Fig. 1. State diagram of the sender algorithm

2.1 Sender and Receiver Algorithms

The sender initiates a transfer session (t_i) and moves to **transmit** state. It first sends all data packets in its buffer which is maintained by the vector $\mathbf{s}$. Each packet contains the id of the session, sequence number in the buffer and the size of the buffer to be transmitted. After transmitting the last packet, the sender switches to **collect_requests** state and broadcasts a request notification to the receivers.

Upon arrival of the first data packet, the receiver node records the sender, allocates a receive buffer and the bit vector $\mathbf{r_i}$ by using the information in the data packet and places the data packet into the buffer. Receivers maintain the bit vector to identify successfully received and lost packets. When a packet is successfully received, the corresponding value in the vector is changed from “1” to “0”. They do not respond to data packets until the notification is received from the sender. After the notification, they generate a request

message for the missing packets in the arriving stream. Methods like "random time-slot-assignment" can be used to reduce the number of collisions during this state.

After collecting the request vectors of the receivers the sender forms the $\mathbf{R}$ matrix and generates its new vector $\mathbf{s}$ using the formula $\forall i s_i = \bigcup_{j=1}^{n} r_{ji}$. So, if any of the receivers has "1" for a packet, the corresponding value in the send vector will be "1"; which means it is still to be retransmitted. Then the sender scans the vector from the beginning to the end and broadcasts the packets which have "1"s in the vector.

The receiver switches back to the **receive** state whenever it receives the first retransmitted data packet. If it has no missing packets left or no packets to send, it switches to **done** state and stays silent.

These retransmission phases are repeated until the termination condition is reached which can be formulated as $\forall i \mathbf{s}_i = 0$. If all the values in the sender vector are zero, this means all the receivers have received all the packets.

3. ANALYSIS

We aim to derive a generalized formula for the total number of packets sent in a reliable buffer transfer session. Hence in this section, we analyze the steps of the protocol and present the derivation of the general formula.

We analyze the retransmission scheme in terms of total number of packets sent in a data transfer session. The symbols used in the equations are: N, which represents data buffer size, i.e. the number of packets to be send; p, is the independent packet loss probability of the wireless links for each receiver; and k, is the number of the receivers.

The following analysis is carried out for a network as given in **Fig. 2**. The network consists of a number of cells all of which include node 0 and some other nodes. Node 0 hears all the others but the other nodes need not hear all. So, there is a one hop communication between node 0 and the others.

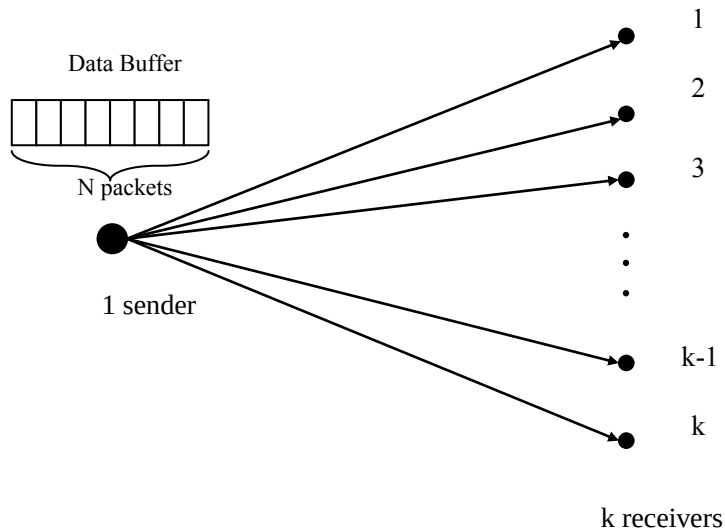

Fig. 2. General view

Node 0 is the sender and the other *k* nodes are the receivers of the buffer. Each communication link between the sender and a receiver is modeled as an independent Bernoulli process; i.e., a data packet sent to any receiver is lost with independent probability *p*. We assume that this probability is equal for all channels. The sender has a buffer of *N* packets to be transmitted to all receivers. Hence, *1-to-k* reliable transmission is required.

We start the analysis by defining the following function:

Definition 1: $f(k,N)$ is a function which gives the expected number of messages that the sender should send in total, in order to have *k* receivers to get *N* packets reliably. So, for example $f(3,100)$ gives us the number of messages the sender should send when 3 receivers should get a buffer of 100 data packets.

When a node sends a message to *k* receivers, *k+1* cases can be differentiated: The message may be received by all receivers successfully, or it may be lost by *i* receivers and received by the remaining *k-i* receivers; where *i* = 1,2,...,*k*. The probabilities for all these cases are given in Equation 1.

$$P(k,i) = \binom{k}{i} p^i (1-p)^{k-i} \quad for \quad i = 0..k. \tag{1}$$

We will first find $f(k,N)$ values for specific *k* values. Then we will rewrite these values using partial fraction decomposition and show that they fit to the resulting generalized formula.

For ***k*=1** (depicted in **Fig. 3**)***;*** the expected number of messages the sender should send in total is *f(1, N).* This value can be found as follows: first the sender sends *N* messages each with independent *p* loss probability; this reveals that the receiver looses *Np* packets

and receives *N(1-p)* packets. Then the receiver sends back a request message for the lost packets. In the next phase, the sender retransmits *Np* messages and Np^2 of them get lost by the receiver. Then it sends Np^2 messages; Np^3 of them get lost and so on. This process continues until all the messages are received by the receiver. Hence, the total numbers of messages sent by the sender can be found as,

$$f(1,N) = N\sum_{i=0}^{\infty} p^i = N(1+p+p^2+\cdots+\infty) = N\frac{1}{1-p}. \tag{2}$$

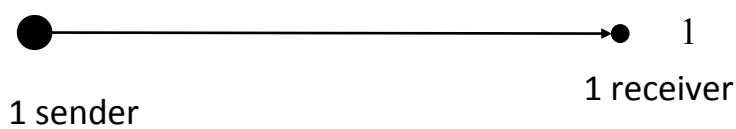

Fig. 3. 1 receiver

For ***k*=2** (depicted in **Fig. 4**)***;*** *N* messages sent by the sender will be divided into *three* groups in the first phase. Some messages will be received by both receivers; some of them will be lost by either one of the receivers and some of them will be lost by both. For the first group of messages, no retransmission is required. The expected number of messages in the second group, i.e. the messages lost only by one receiver, can be found from Equation 1 as $N\,P(2,1)$ = *2Np(1 - p)*. This group reduces to the *k*=1 case; i.e. these messages can be thought of being sent to 1 receiver. So, the number of messages that the sender should send for these messages is equal to *f(1, 2Np(1 - p))*. The last group, i.e. the messages lost by both receivers, consists of Np^2 messages. These messages will be retransmitted by the sender and again as in the first phase, they will be divided into *three* groups. The only difference is the number of messages; Np^2 instead of *N*. Again, *f(1,* $2Np^3(1 - p)$*)* messages will be sent for the messages which are lost by 1 receiver; and Np^4 messages will be sent for the messages lost by both.

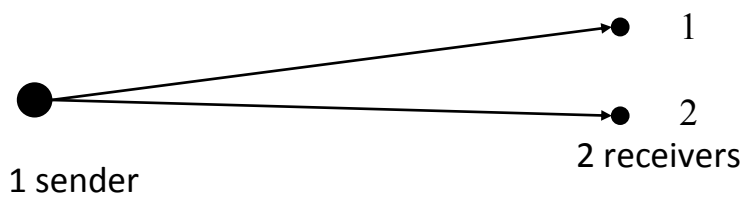

Fig. 4. 2 receivers

This process continues recursively until all the messages are received by both receivers. If we add up the total number of messages sent by the sender we get:

$$\begin{aligned} f(2,N) &= \sum_{i=0}^{\infty} f(1, 2Np^{2i+1}(1-p)) + N\sum_{i=0}^{\infty} p^{2i} \\ &= 2Np\sum_{i=0}^{\infty} p^{2i} + N\sum_{i=0}^{\infty} p^{2i} \qquad . \qquad (3) \\ &= \frac{2Np}{1-p^2} + \frac{N}{1-p^2} = N\frac{1+2p}{1-p^2} \end{aligned}$$

For ***k*=3** (depicted in **Fig. 5***;* the first *N* messages sent by the sender will be divided into *four* groups. From Equation 1, $N(1-p)^3$ messages will be received by all receivers. The number of the messages lost by only 1 receiver will be $3Np(1-p)^2$. This group reduces to *k*=1 case; hence the number of messages sent for this group will be $f(1, 3Np(1-p)^2)$. $3Np^2(1-p)$ of the first *N* messages will be lost by 2 receivers, reducing to *k*=2 case. The number of messages sent for this group will be $f(2, 3Np^2(1-p))$. Np^3 of the first *N* messages will be lost by all the receivers. The sender will retransmit these messages to all receivers. Again, these messages will be divided into *four* groups just like in the first phase.

The same analysis holds for this phase recursively; the only difference is the number of messages; Np^3 instead of *N*. As a result, for $f(3, N)$ we get:

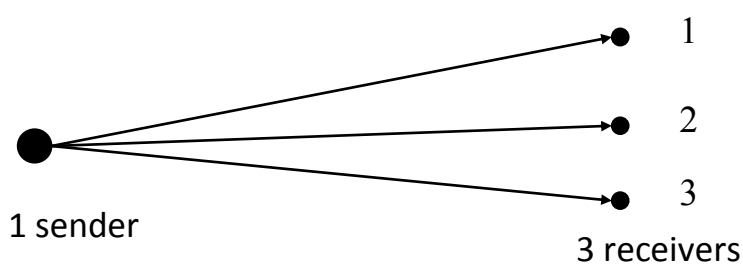

Fig. 5. 3 receivers

$$f(3,N) = \left[\begin{array}{c} \sum_{i=0}^{\infty} f(1,3Np^{3i+1}(1-p)^2) \\ +\sum_{i=0}^{\infty} f(2,3Np^{3i+2}(1-p)) + N\sum_{i=0}^{\infty} p^{3i} \end{array} \right] \tag{4}$$

$$= \frac{N}{(1-p^2)(1-p^3)}\left(1+3p-p^2-3p^4\right).$$

For ***k*=4** (depicted in **Fig. 6**); the first *N* messages will now be divided into *five* groups. From Equation 1; $N(1-p)^4$ messages will be received by all receivers. $4Np(1-p)^3$ messages will be lost by only 1 receiver (*k*=1 case). The number of messages sent for this group will be $f(1, 4Np(1-p)^3)$. $6Np^2(1-p)^2$ messages will be lost by 2 receivers; $f(2, 6Np^2(1-p)^2)$ messages will be sent for these messages. $4Np^3(1-p)$ of the first *N* messages will be lost by 3 receivers. $f(3, 4Np^3(1-p))$ messages will be sent for these messages. The number of the messages lost by all receivers in this case will be Np^4. The sender will retransmit these messages.

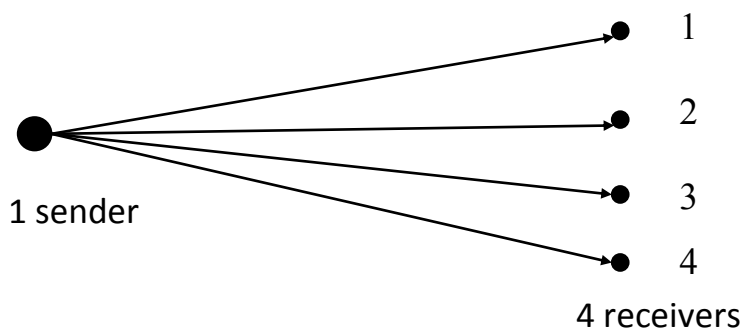

Fig. 6. 4 receivers

Again, the analysis for the first phase holds recursively for the following phases. Only the initial message quantity changes: N, Np^4, Np^8, ... Hence, for $f(4, N)$ we get:

$$f(4,N) = \left[\begin{array}{l} \sum_{i=0}^{\infty} f(1,4Np^{4i+1}(1-p)^3) \\ + \sum_{i=0}^{\infty} f(2,6Np^{4i+2}(1-p)^2) \\ + \sum_{i=0}^{\infty} f(3,4Np^{4i+3}(1-p)) + N\sum_{i=0}^{\infty} p^{4i} \end{array}\right] \tag{5}$$

$$= \frac{N}{(1-p^2)(1-p^3)(1-p^4)}\left(1+4p-3p^2+\cdots+4p^8\right).$$

Using the same analysis, we have obtained the results for 5 and 6 receivers. These values are not given here due to size limitations. However, they are used in the next subsection for partial fraction decomposition.

2.1 Partial Fraction Decomposition

In the previous section, we have found formulas for the total number of messages that should be sent for different number of receivers (k = 1, 2, 3, 4, 5 and 6). In this section, using the partial fraction decomposition method we will rewrite these formulas in fractions.

For each case, the formula can be written in the form of $f(k,N) = NF(k)$; where $F(k)$ is a polynomial fraction for k receivers including some constants and p. We will decompose $F(k)$ into fractions for each case. For k=1 and k=2, the results are already decomposed. For 3 receivers, using partial fraction decomposition with Equation 4 we get,

$$\frac{1+3p-p^2-3p^4}{(1-p^2)(1-p^3)} = \frac{a_0+a_1p}{1-p^2} + \frac{b_0+b_1p+b_2p^2}{1-p^3}. \tag{6}$$

Solving this equation gives us,

$$F(3) = \frac{-3}{1-p^2} + \frac{4+3p+3p^2}{1-p^3}. \tag{7}$$

For 4 receivers, using Equation 5 and the similar partial fraction decomposition we obtain,

$$F(4) = \frac{-6}{1-p^2} + \frac{4}{1-p^3} + \frac{3+4p+4p^2+4p^3}{1-p^4}. \tag{8}$$

For 5 and 6 receivers we have,

$$F(5)=\frac{-10}{1-p^2}+\frac{10}{1-p^3}+\frac{-5}{1-p^4}+\frac{6+5p+5p^2+5p^3+5p^4}{1-p^5}, \tag{9}$$

$$F(6)=\begin{array}{l}\dfrac{-15}{1-p^2}+\dfrac{20}{1-p^3}+\dfrac{-15}{1-p^4}+\dfrac{6}{1-p^5}\\ +\dfrac{5+6p+6p^2+6p^3+6p^4+6p^5}{1-p^6}.\end{array} \tag{10}$$

respectively. When carefully examined, it can be seen that there is a relation between these equations and the binomial triangle. In fact, the coefficients of the polynomials on the fractions are coming from this famous triangle. When we generalize these equations, for k receivers, we obtain the following formula:

$$\begin{array}{l}\dfrac{-\binom{k}{2}}{1-p^2}+\dfrac{+\binom{k}{3}}{1-p^3}+\dfrac{-\binom{k}{4}}{1-p^4}+\dfrac{+\binom{k}{5}}{1-p^5}+\cdots+\dfrac{\mp\binom{k}{k-1}}{1-p^{k-1}}\\ +\dfrac{\binom{k}{1}\mp 1+\binom{k}{1}p+\binom{k}{1}p^2+\binom{k}{1}p^3+\cdots+\binom{k}{1}p^{k-1}}{1-p^k},\end{array} \tag{11}$$

which can be rewritten as,

$$\sum_{i=2}^{k-1}\frac{(-1)^{i+1}\binom{k}{i}}{1-p^i}+\frac{\sum_{i=0}^{k-1}kp^i+(-1)^{k+1}}{1-p^k}. \tag{12}$$

We know that $\sum_{i=0}^{k-1}p^i=\dfrac{1-p^k}{1-p}$; so the formula becomes:

$$F(k) = \sum_{i=2}^{k-1} \frac{(-1)^{i+1}\binom{k}{i}}{1-p^i} + \frac{k\frac{1-p^k}{1-p} + (-1)^{k+1}}{1-p^k} = \sum_{i=2}^{k-1} \frac{(-1)^{i+1}\binom{k}{i}}{1-p^i} + \frac{k}{1-p} + \frac{(-1)^{k+1}}{1-p^k}. \tag{13}$$

Finally, rearranging and multiplying this polynomial fraction with N, we obtain the generalized formula as,

$$f(k,N) = N\sum_{i=1}^{k} \frac{(-1)^{i+1}\binom{k}{i}}{1-p^i}. \tag{14}$$

4. SIMULATION RESULTS

To verify the generalized formula given in Equation 14, we have carried out exhaustive number of Monte Carlo simulations for different loss probabilities (p) and number of receivers (k); namely, p values ranging from 0.1 to 0.9 and the number of receivers ranging from 1 to 40.

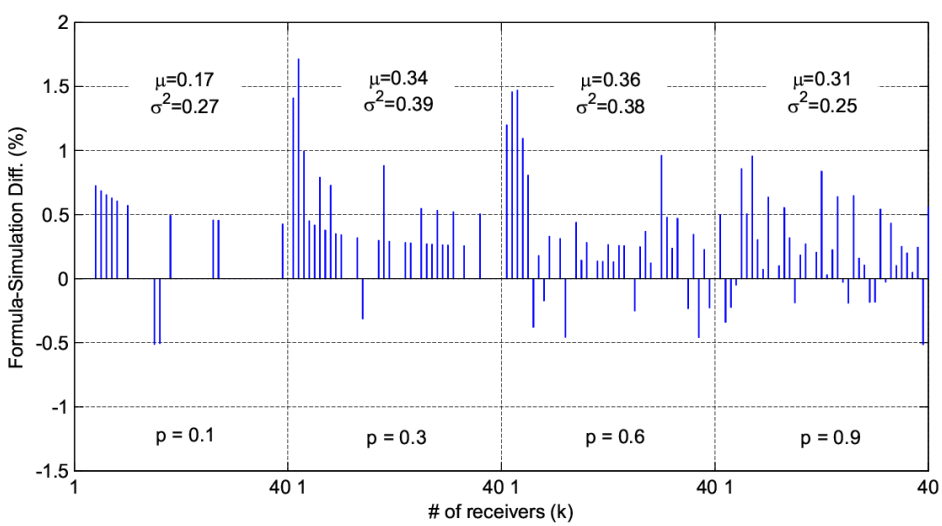

Fig. 7. Simulation results for 100 runs

The setting of the simulation environment is as follows: For each simulation there are one sender and *k* receivers. The sender sends a buffer of 100 packets to all the receivers reliably. For each simulation, *p* is the packet loss probability value and is assumed that all the receivers' communication channels have the same and independent loss probability.

For every *p* and *k* pair, the simulations have been run for 100 different random seed values and the arithmetic average of the results coming from these simulations have been compared with the results coming from the formula. The percentage difference between the simulations and the formula are illustrated in **Fig. 7**. As can be seen, deviation in the results are ranging from -0.51% to 1.71% which is considerably small. Then, the same simulations have been repeated for 1000 runs. As the results illustrated in **Fig. 8** suggest, the deviation has decreased; namely between -0.53% to 0.72.

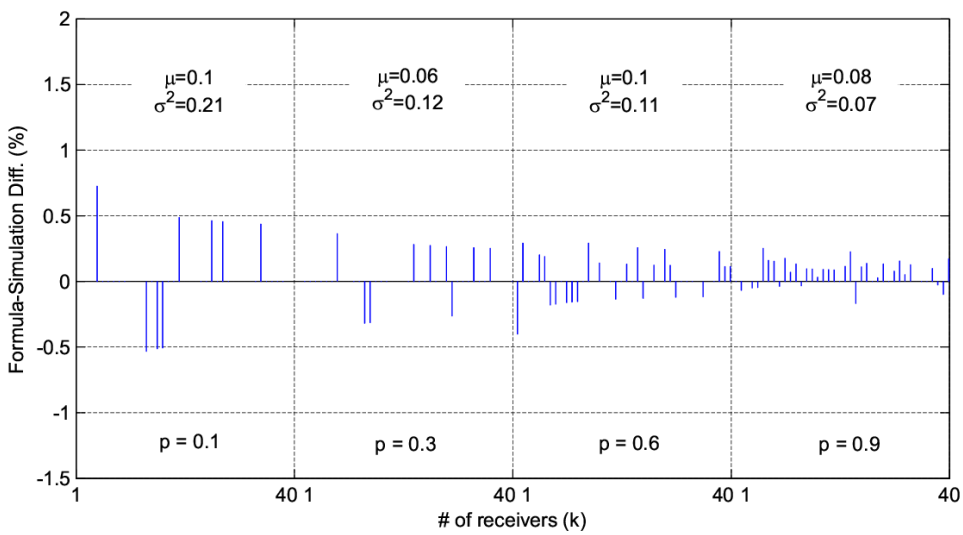

Fig. 8. Simulation results for 1000 runs

We have also done wifi simulations to compare the performance of the analyzed cooperative (*c-arq*) method where cooperation among receivers is achieved with joint retransmissions at the sender and the traditional (*t-arq*) method utilized in 802.11 networks where sender establishes a separate ARQ link with very receiver. Simulations are carried out for *p* values ranging from 0.1 to 0.9 and *k* values up to 40. The results, in terms of total number of sent messages when a buffer of 100 messages is sent to *k* receivers, are illustrated in **Fig. 9**. These results show that, the number of messages sent for traditional method is always larger than that of the cooperative scheme. Especially when the number of receivers increases, there is a significant and large reduction in the total number of sent messages when joint retransmissions are used.

5. CONCLUSION

In this paper, we propose a generalized formula for the analysis of the reliable transmission of a network buffer of size N to k receivers over lossy broadcast channels when the sender protocol joins the retransmissions for multiple receivers. We present the derivation of the formula for the total number of messages transmitted in a single broadcast transfer session. The derived formula can further be used in the analysis of cooperative ARQ based protocols over wireless networks. We also show that using joint retransmissions significantly reduces the number of sent messages, hence the energy consumption.

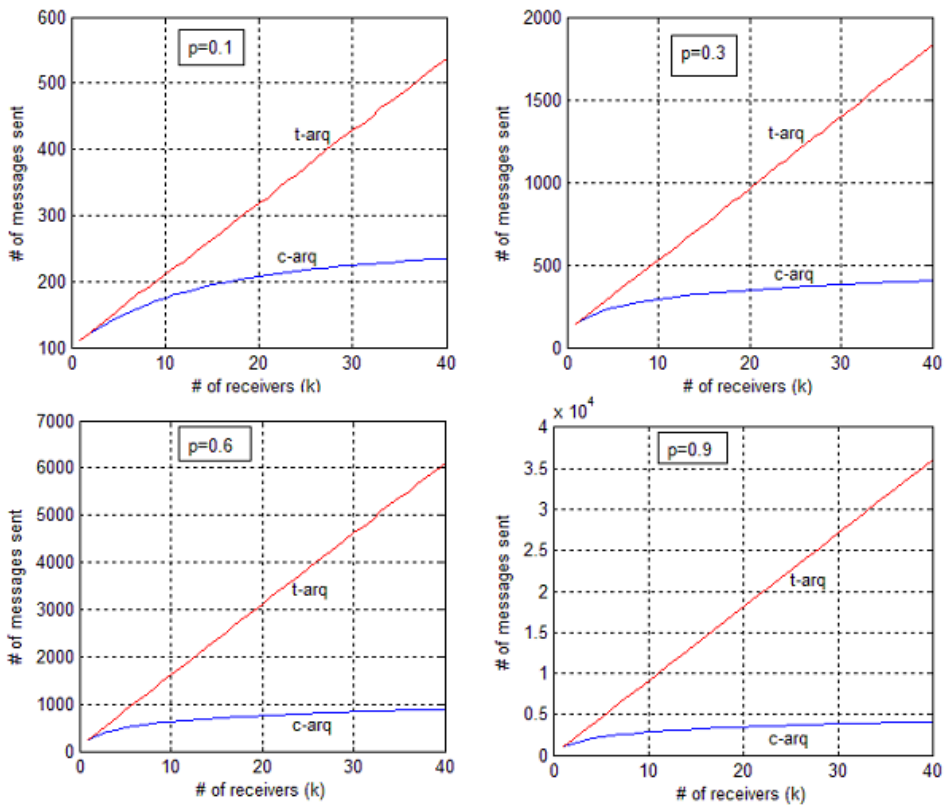

Fig. 9. Comparison of Traditional and Joint ARQ

REFERENCES

[1] I. Chlamtac, C. Petrioli and J. Redi, "Energy-conserving go-back-n arq protocols for wireless data networks," in *Proc. of ICUPC '98: IEEE International Conference on Universal Personal Communications*, vol. 2, pp. 1259-1263, Florence, Italy, October 5-9, 1998.

[2] F. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *Proc. of 1st IEEE International Workshop on Sensor Network Protocols and Applications (SNPA)*, pp.102-112 Anchorage, Alaska, May 11, 2003.

[3] B. Ouyang, X. Hong and Y. Yi, "A comparison of reliable multicast protocols for mobile ad hoc networks," in *Proc. of IEEE SoutheastCon*, pp. 339-344, Ft. Lauderdale, FL, USA, April 8-10, 2005.

[4] D. Koutsonikolas and Y. C. Hu, "The case for fec-based reliable multicast in wireless mesh networks," in *Proc. of DSN '07: The 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, pp. 491-501, Edinburgh, UK, June 25-28, 2007.

[5] D. Koutsonikolas and Y. C. Hu, "Exploring the design space of reliable multicast protocols for wireless mesh networks," Ad Hoc Networks, vol. 7, issue 5, pp. 932-954, July, 2009.

[6] J. Kuri and S. K. Kasera, "Reliable multicast in multi-access wireless lans," in *Proc. of INFOCOM '99: 18th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2, pp. 760-767, New York, NY, March 21-25, 1999.

[7] B. Liu, F. Ren, C. Lin and Y. Ouyang, "Performance analysis of retransmission and redundancy schemes in sensor networks" in *Proc.of ICC 2008: IEEE International Conference on Communications*, pp. 4407-4413, Beijing, May 19-23, 2008.

[8] W. Song and W. Zhuang, "Performance analysis and enhancement of cooperative retransmission strategy for delay-sensitive real-time services" in *Proc.of GLOBECOM 2009: IEEE Global Communications Conference*, pp. 1-6, Honolulu, Hawaii, USA, 30 November - 4 December 2009.

[9] T. Predojev, J. Alonso-Zarate and M. Dohler, "Energy efficiency of cooperative ARQ strategies in low power networks," 2012 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Orlando FL, 25-30 March 2012.

[10] J. Alonso-Zarate, L. Alonso, G. Kormentzas, R. Tafazolli and C. Verikoukis, "Throughput Analysis of a Cooperative ARQ Scheme in the Presence of Hidden and Exposed Terminals," Mobile Networks and Applications, vol. 17, issue 2 (April 2012), pp. 258-266.

Mustafa Bulent Mutluoglu received the B.S. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, in 1994, the M.S. and Ph.D. degrees in Computer Engineering from Yildiz Technical University, Istanbul in 1999 and 2008, respectively. He is currently an instructor in the Department of Computer Engineering at Yeditepe University, Istanbul, Turkey. His current research interests include optimum tree construction and network reprogramming in wireless sensor networks.

Sebnem Baydere received her BSc and MSc degrees in computer engineering from METU, Ankara, Turkey in 1984 and 1987 respectively. She received the PhD degree in computer science from UCL, London in 1990. Dr. Baydere is a Professor and the Director of Wireless Sensor Networks Research Lab in the Department of Computer Engineering at Yeditepe University, Istanbul, Turkey since 1996. Her current research interests include wireless adhoc sensor networks, mobile computing and distributed systems.