

Multimedia Streaming for Ad Hoc Wireless Mesh Networks Using Network Coding

Basil Saeed¹, Chung-Horng Lung¹, Thomas Kunz¹, Anand Srinivasan²

¹Department of Systems and Computer Engineering, Carleton University, Ottawa, Canada

²EION Inc., Ottawa, Canada

Email: bsaeed@sce.carleton.ca, chlung@sce.carleton.ca, tkunz@sce.carleton.ca, anand@eion.com

Received March 2, 2013; revised April 2, 2013; accepted May 2, 2013

Copyright © 2013 Basil Saeed *et al.* This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

Over the past years, we have witnessed an explosive growth in the use of multimedia applications such as audio and video streaming with mobile and static devices. Multimedia streaming applications need new approaches to multimedia transmissions to meet the growing volume demand and quality expectations of multimedia traffic. This paper studies network coding which is a promising paradigm that has the potential to improve the performance of networks for multimedia streaming applications in terms of packet delivery ratio (PDR), latency and jitter. This paper examines several network coding protocols for ad hoc wireless mesh networks and compares their performance on multimedia streaming applications with optimized broadcast protocols, e.g., BCast, Simplified Multicast Forwarding (SMF), and Partial Dominant Pruning (PDP). The results show that the performance increases significantly with the Random Linear Network Coding (RLNC) scheme.

Keywords: Wireless Broadcast; Multimedia Streaming; Audio Streaming; Video Streaming; Network Coding; Random Linear Network Coding; PDP; SMF; BCast

1. Introduction

Broadcasting is a linear transmission mechanism including audio and video traffic in real time. Several types of devices such as TVs, radios, computers and mobile devices (cell phones) are used as receivers to gain access to one broadcasted traffic flow at a time per channel with pre-scheduled start and end times. The receivers are controlled by the users to be switched on or off, as well as being controlled by frequency tuning. Practically, all of the content offered by mobile, radio and TV stations today are available only in this approach. The digital multimedia broadcasting standards support high definition television (HDTV), multiple standard definition television (SDTV) program streams, and private data applications such as broadcast duplicate transmissions, multimedia pager data burst, HTML pages, audio streaming, video streaming, etc. [1].

With the explosion of Internet traffic seen over the past two decades, coupled with the ever increasing need to access critical data at any location, wireless networks have emerged as a means of effectively communicating in an on-demand fashion from nearly any location. This new communication paradigm is inherently based on a

broadcast medium and presents challenges not seen in traditional wire line networks due to the nature of the wireless medium in which users must share access to: 1) frequencies or 2) time-slots controlled by the Medium Access Control (MAC) model used. The challenges include bandwidth limitations, mobility impacts, energy consumption, unreliable transmission, security issues and dead spots.

We have witnessed an explosive growth in the use of multimedia applications with mobile and static devices lately. Multimedia broadcasting in wireless networks intends to transmit concurrently identical multimedia traffic to multiple receivers. The main important trends for mobile and static multimedia broadcasting are: 1) Mobile traffic is growing significantly, and will be dominated by video and voice; 2) Mobile devices are getting more powerful; 3) mobile graphics are getting better [2]. As a result, mobile multimedia users are expected to grow rapidly as shown in **Figure 1** which shows the growth trend for audio streaming applications presented in [3], and **Figure 2** which shows the growth trend for TV and video user as it is expected to blossom to more than four hundred and fifty million by 2014 [2].

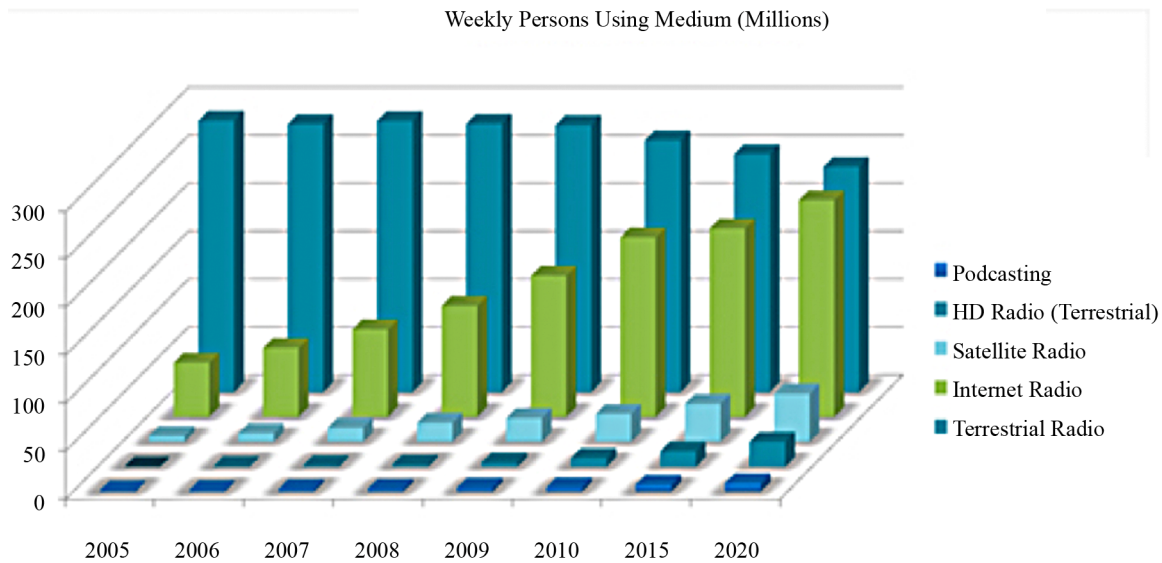
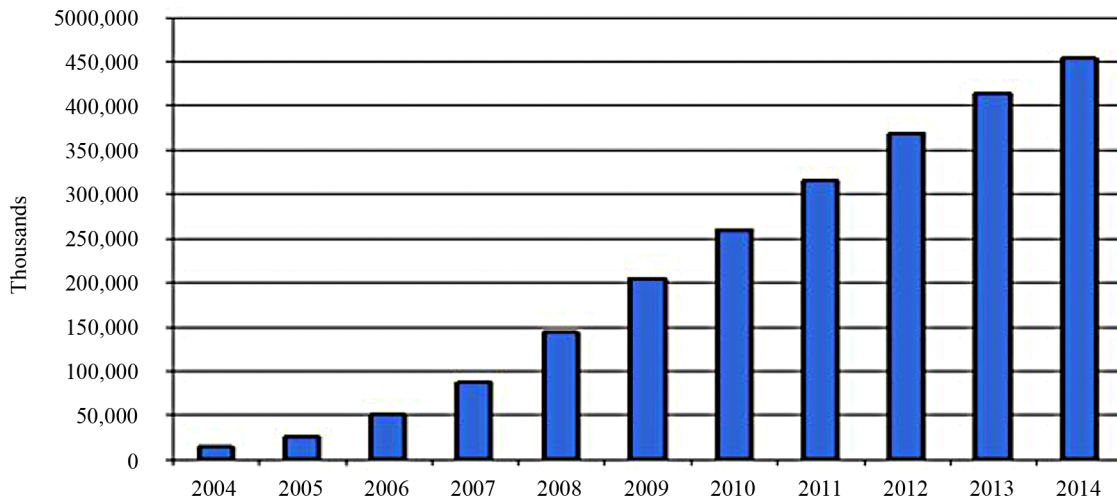


Figure 1. Audio Streaming Growth [3].



The Yankee Group forecasts the number of mobile video/YV users will blossom to more than 450 million by 2014.

Figure 2. Mobile video/TV users from 2004-2014 [2].

Due to the above mentioned growth trend in the number of mobile users, the wireless multimedia broadcasting services are faced with a number of challenges: the wireless capacities are still limited and can not support rich multimedia traffic to mobile devices because of the increase in video and voice qualities, which leads to an increase in the packet size of the multimedia traffic (video and voice).

Furthermore, multimedia-streaming traffic is extremely time-sensitive, as it requires timely delivery of each multimedia packet. Packets that are received beyond an application-specific deadline are ignored by the application and the transmission therefore wastes precious wireless bandwidth. The original playout delay at the receiver essentially influences the application acceptance and the

user's reliability [4].

Exploiting the characteristics of the wireless medium, especially the broadcast communication channel, Network Coding (NC) has emerged as a promising paradigm in order to increase the capacity or the throughput of the network. NC originally was proposed by Ahlswede *et al.* [5] for wired networks, but Li *et al.* [6] were among the first to apply network coding to the wireless domain. All nodes (sending or intermediate nodes) act as a relay and also can combine several received packets into one or several encoded outgoing packets, thus the performance of the network increases in terms of throughput, delay, and etc.

Due to the benefits of NC and the increasing popularity of multimedia applications, we were motivated to

apply NC mechanisms to multimedia wireless broadcasting applications. Likewise, many proposed broadcast protocols (with or without NC) have parameters that can be set to suitable values to tune their performance in the face of the above mentioned challenges. It is both theoretically and practically important to find a solution where several types of multimedia traffic can be delivered successfully to the destinations in a way that satisfies the customers.

In this paper, we have studied broadcast protocols that work well with both single source and multiple sources networks. The paper also has investigated the potential benefits of several NC techniques, such as Random Linear Network Coding (RLNC), Reed Solomon (RS) codes, and XOR, where the NC mechanism combines different packets generated by different sources into a single encoded packet. The paper investigates various NC techniques for multimedia streaming such as audio and video streaming. We have shown through simulations that RLNC is able to adapt well to audio and video streaming applications and to provide stable performance in terms of packet delivery ratio (PDR), latency and jitter. We also compare RLNC's performance with three popular and efficient broadcast protocols, e.g., BCast, Simplified Multicast Forwarding (SMF) and Partial Dominant Pruning (PDP).

The rest of the paper is organized as follows. Section 2 introduces the protocols used in this paper. Section 3 describes the simulation environment and simulation results for various scenarios. Finally, in Section 4, we conclude the paper and suggest some possible lines for future research.

2. Background

This section first discusses the BCast, SMF, and PDP broadcast protocols which are used in the paper. BCast, SMF and PDP are three efficient broadcast protocols for Mobile Ad-hoc Networks (MANETs) and wireless mesh networks [7,8]. Following that, the section briefly describes some network coding schemes that have been used for broadcast. Those network coding schemes include XOR, RS, and RLNC.

2.1. BCast Protocol

The BCast protocol [8,9] is a scalable broadcast algorithm that uses 2-hop neighbor knowledge. Each node periodically sends discovery messages that contain a node's identifier, e.g., IP address, and its list of identified neighbors. When the discovery messages are received by a node from all its neighbors, then this node has 2-hop topology information. For example, when node "B" receives a discovery packet from node "A", then "B" knows all neighbors of "A". Also, if "B" has neighbors

not covered by one of its neighbors "A", "B" prepares a discovery message to be broadcasted to its neighbors. If "B" receives a discovery message from another neighbor "C" before sending its own discovery message, "B" will drop the discovery message and initiates a new message with the newly received information.

BCast is also a reliable broadcast algorithm, where nodes buffer the most recent packets. For example, if node "A" receives packet "N" from sender "B", "A" will check if packet "N-1" was received from the sender as well. If not, "A" will ask its neighbors to retransmit packet "N-1" by sending a "NACK" packet that includes the information about the missing or delayed packet. Nodes receiving a "NACK" packet will check their buffer, and if they have the missing or delayed packet buffered, they will schedule a retransmission. If nodes overhear another node retransmitting that packet, those nodes will cancel their own retransmission.

2.2. Simplified Multicast Forwarding

The Simplified Multicast Forwarding (SMF) protocol is an optimized multicast/broadcast protocol which provides a forwarding plane for packets from source to destination. The key benefit of the SMF protocol is the improvement of the performance of relays in dynamic networks. Using the concept of Multipoint Relays (MPRs) from the Optimized Link State Routing (OLSR) [10] Protocol, SMF heuristically selects only a subset of nodes to retransmit data packets to reach all nodes in a network. The subnet is created using a two-hop neighbor discovery mechanism by periodically sending Hello messages. The nodes then select and signal a subset of their one-hop neighbors as MPR nodes. Moreover, SMF has a duplicated packet detection mechanism to avoid retransmitting any repetitive packets. This mechanism uses the IP address Identification "ID" field to detect any duplication where each packet has a unique "ID" field [7,11].

2.3. Partial Dominant Pruning

Dominant Pruning (DP) is a broadcast protocol which aims to minimize the number of transmissions. Similar to the case of SMF, DP uses HELLO messages to collect information about 2-hop neighbors in the network. Based on the collected information and the Partial Dominant Pruning (PDP) algorithm, the source node selects a set of nodes to broadcast the sent packet rather than having all source's neighbors retransmit, taking into account which neighbors have already received the packet. This increases the data packet size, as more information will be attached to the header of the data packet. Moreover, the HELLO messages of the PDP algorithm are smaller, as they do not carry information about MPR selection, compared to SMF. The PDP algorithm is considered as one of the

optimized broadcast protocol for MANETs and wireless mesh networks [7,12].

2.4. SMF and PDP with Network Coding Techniques

NC-based transmission techniques seek to combine packets from different sources and transmit these combined packets into a single or multiple transmissions [1]. The receiver decodes the received combined packets to recover the original/desired data when it receives enough information that is needed in the decoding procedure. As shown in **Figure 3**, if nodes “1” and “3” wish to exchange information it could be performed via the following: 1) node “1” transmits S_1 to node “2”; 2) node “3” transmits S_3 to node “2”; 3) node “2” performs a coding operation on the information and transmits S_2 to both nodes “1” and “3”, taking a total of 3 time slots. The received information is then decoded at the edges, an operation that is based on the type of network coding employed in the original coding operation.

SMF and PDP have been extended with two NC techniques; XOR and Reed Solomon (RS) [7]. In both techniques, a node keeps monitoring the wireless medium and collects information about its 1-hop and 2-hop neighbors. A node opportunistically, and without additional control packets, maintains information about which packets were already received by what neighbor to guide its encoding decision. The exploit these coding opportunities, nodes accumulate data packets in a small internal buffer (typically 4 to 8 packets). Once the buffer is full or a protocol-specific timeout occurs, a node will determine which buffered packets to code together, based on its knowledge of the reception state of its neighbors and the specific NC scheme.

The XOR network coding (XOR-NC) technique is one of the simplest digital network coding techniques to encode and decode data packets. Using XOR-NC, the relay combines the received packets into one packet using an XOR operation and broadcasts the new packet to receivers. Receivers can then decode the encoded message by XORing it with their prior packet(s) [7].

Figure 3 shows an example of the XOR-NC approach which would be performed as following in the wireless networks. Both node “1” and node “3” send their packets respectively to the relay (node “2”) using different time slots. The relay combines the received packets into one

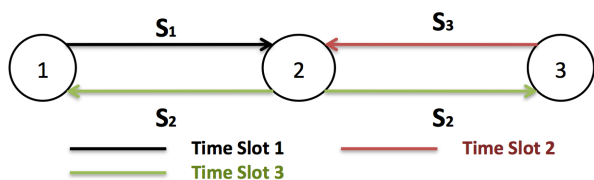


Figure 3. Network Coding.

packet using an XOR operation, and broadcasts the new packet to node “1” and node “3” in the third time slot. Node “1” and node “3” can decode the information sent to them via XORing the coded packet sent by the relay with their prior packets [14]. The total number of transmissions using this XOR-NC approach equals to 3 transmissions or time slots. This example shows that using the XOR-NC approach allows node to send more packets using fewer time slots.

Reed Solomon (RS) codes are mainly used as error-correcting techniques. RS codes have been used in data storage; compact discs and in the encoding/decoding systems that allow scratched CDs to sound perfect.

Figure 4 shows the processes of RS error correcting. An RS code can be represented as $RS(n, k)$ with s -bit symbols, where k is the size of the message in symbols of s bits that a source wants to encode into a codeword and n is the size of codeword. In **Figure 4**, the source wants to send the 1011 message to a receiver. The source encodes the 1011 message into codeword 1011010 and sends the codeword through a channel to the receiver, but during the transmission the codeword is changed from 1011010 to 1010010, as a transmission error caused the fourth bit to be flipped. At the receiver side, the decoder corrects the error and reclaims the source message 1011.

[13] introduces a new network coding technique that is based on RS codes. The technique assumes that:

- 1) The sender “ u ” has in the queue the set P of n native packets.
- 2) The set of u ’s neighbors “ v ” has received the set P_v .
- 3) P_v is a set of formerly received packets at v .
- 4) $k = \max\{|P - P_v|\}$, where k is the minimum number of encoded packets to send such that each neighbour v can decode the packets.

$$\Theta = \begin{pmatrix} 1^0 & 2^0 & \dots & n^0 \\ 1^1 & 2^1 & \dots & n^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1^k & 2^k & \dots & n^k \end{pmatrix} \tag{1}$$

Based on the previous assumptions, the Reed Solomon codes for the technique are generated by a $k \times n$ Vandermonde matrix Θ . The elements of the matrix are cho

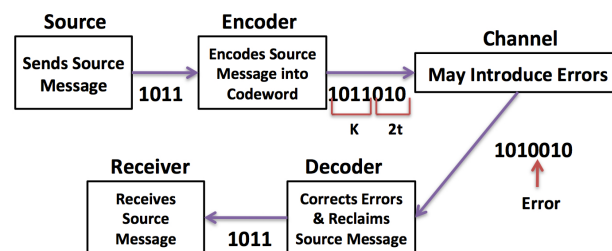


Figure 4. Reed Solomon [7].

sen from the finite field F_{2^d} . Equation (1) represents the $k \times n$ Vandermonde matrix.

In this technique, the sender “u” is responsible to perform the encoding operation which constructs a set of encoded packets Q where $Q = \Theta \times P$. After receiving the set Q by “v”, the receiver goes over all the received encoded packets, checks the set P_v and creates a new coefficient vector by ignoring the coefficient of P_v . After that, the receiver constructs a new decoding matrix A_v from the new coefficient vector, and solves $A_v \times Q$ to recover the set P which contains the native packets “n”. The protocol performance depends on the topology and node density of the network, which directly impacts the opportunistic listening mechanism.

2.5. Random Linear Network Coding

The Random Linear Network Coding (RLNC) technique encodes a packet using linear transformations. When a group of intermediate nodes try to deliver information to a destination, each node generates a new packet which is a linear combination of received packets. The destination then performs a Gaussian Elimination to decode the independent received packets in a linear manner [15,16]. Ho *et al.* [17] have shown that the use of random linear codes is an efficient and practical technique to design linear codes to be used. In RLNC, intermediate nodes independently and randomly choose the coefficients of the linear combination from a finite field called Galois Field (GF). To decode a packet, the receiver should receive an amount of linearly independent combinations of packets that equals to the number of source processes.

The process of RLNC is divided into two operations [18]; encoding and decoding operations. As shown in **Figure 5**, the encoder is responsible to perform the encoding operation which combines linearly the original messages with independently and randomly selected coefficients from GF. Assume that a node has N data packets P_1, P_2, \dots, P_N to send simultaneously. Using RLNC and $GF(2^s)$, where s is the size of the finite field, the node selects N coefficients $c_j = c_{j1}, c_{j2}, \dots, c_{jN}$ from the finite field to form a coding vector, and performs

$\sum c_{ji} \times P_i$ to generate a new encoded packet X_j that combines all data packets that the node wants to send. To illustrate the RLNC technique, a node has two data packets to send; P_1 and P_2 . The node can send both data packets simultaneously by choosing two coefficients (c_1, c_2) from the finite field and performing the previous linear operation which will result in an encoded packet that contains both P_1 and P_2 ; $X_1 = c_{11}P_1 + c_{12}P_2$. The actual summation occurs in the form of a matrix multiplication by forming the coding coefficients c_j and the data packets (P_1, P_2, \dots, P_N) as vectors. The data packets are vectors, as is the encoding vector chosen by an encoder. The same set of packets may be encoded multiple times, so the sender ends up with many different encoding vectors and packet vectors, and summarizes the net effect of all these operations in matrix form. Thus, the matrix form of the $c_{ji} \times P_i$ expression can be written as $X = C \times B$, where B is the data packets matrix for the packets that the node wants to send, C is the coefficient matrix and X is the encoded packet matrix.

The encoding procedure is also used as the re-coding process where encoding is done in intermediate nodes in the network, using received encoded packets. However, the coding vector in the re-coding process is formed by arithmetic operations between the new coefficients, which are chosen by the intermediate node, and the coefficients which were used to construct the received encoded packets. To illustrate, assume that an intermediate node received two packets $X_1 = c_{11}P_1 + c_{12}P_2$ and $X_2 = c_{21}P_1 + c_{22}P_2$. To perform the re-coding process the intermediate node chooses new coefficients (d_{11}, d_{12}) . After performing the re-coding process, the intermediate node constructs X'_1 as followed:

$$\begin{aligned} & d_{11}X_1 + d_{12}X_2 \\ &= d_{11}(c_{11}P_1 + c_{12}P_2) + d_{12}(c_{21}P_1 + c_{22}P_2) \\ &= (d_{11}c_{11})P_1 + (d_{11}c_{12})P_2 + (d_{12}c_{21})P_1 + (d_{12}c_{22})P_2 \\ &= [(d_{11}c_{11}) + (d_{12}c_{21})]P_1 + [(d_{11}c_{12}) + (d_{12}c_{22})]P_2 \end{aligned}$$

where $[(d_{11}c_{11}) + (d_{12}c_{21})]$ and $[(d_{11}c_{12}) + (d_{12}c_{22})]$ are the new coefficients for the new encoded packet X'_1 .

The second process of the RLNC technique after the

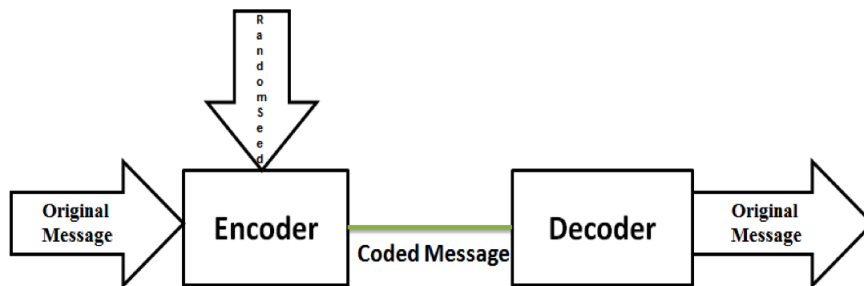


Figure 5. Random Linear Network Coding Diagram [18].

encoding process is the decoding process, which is performed by the destination node after receiving encoded packets. The destination node applies a Gaussian Elimination method to decode linearly independent received packets. The destination node converts received packets into a matrix form and inserts the formed matrix row by row into its decoding matrix. The decoding matrix contains the destination's original packet and the encoded received packets. After that, the destination solves the linear system: $B = C^{-1} \times X$, where B is the recovered data packets matrix, C^{-1} is the inverse coefficient matrix and X is the encoded received packet matrix. If the destination node receives an *innovative* packet which increases the rank of a decoding matrix, then the *innovative* packet will be inserted as the last row of the decoding matrix. The innovative packet is a packet that contains new information which is needed to decode the encoded packet at the destination. On the other hand, if the destination node receives a *non-innovative* packet, then the *non-innovative* packet will be inserted and reduced to a row of *zeros* by the Gaussian Elimination method and ignored.

The work in [15,16] shows how using RLNC can improve the performance of wireless networks in terms of delay and PDR. The RLNC protocol runs over a CSMA MAC layer protocol, and is affected by collisions, interference, and random scheduling of the packets characteristic of wireless networks. The selected MAC layer protocol is based on broadcast transmissions. The protocol does not use any acknowledgement mechanism as acknowledgements are not needed in most multimedia streaming transmissions. Moreover, the MAC layer protocol in this paper and in [15,16] has the same specifications as IEEE 802.11b, where the maximum data rate is 11 Mbit/s using a frequency of 2.4 GHz. However, IEEE 802.11b cannot support a larger packet size and high data rates, such as required for HDTV and SDTV traffic. As a result, the data rate was increased from 11 Mbit/s (as in [15,16]) to 24 Mbit/s for this paper so that several types of multimedia traffic can be studied.

The proposed RLNC protocol has a parameter, called Forwarding Factor, which essentially helps the nodes to decide whether it should create a new random combination if the node receives an innovative packet. For example, if the value of the Forwarding Factor is 1, then the node waits for one innovative packet to create a random combination of packets.

Another factor considered in the RLNC technique is the NC Timer (T) of the MAC protocol. When the NC timer expires, the node makes a decision to create and broadcast the new encoded packet. The NC timer is a uniform random variable in $[0, T_{max}]$. The main advantage of the NC Timer is that it enhances the progress of delivering innovative encoded packets by reducing non-

innovative packet transmissions. The NC timer gives the nodes the chance to collect more innovative packets and send out richer combinations. The main disadvantage of the NC Timer is that it increases the overall delay of the network. As a result, the choice of T_{max} shall be mainly considered based on the performance of the network. This paper adopts a T_{max} value of 20 ms, which was also used in [16].

3. Performance Metrics and Simulation Results

All selected protocols were implemented in the Network Simulator 2 (NS-2). This section explains the simulation scenarios in detail, which model multimedia-streaming applications such as audio and video streaming applications. One or multiple Base Stations (BSs) with an antenna range of 230 m are placed in a $720\text{ m} \times 720\text{ m}$ area where the BSs are geographically centered and evenly distributed in the simulation area to act as broadcast sources. The location of the BSs is shown in **Table 1**. The simulations are divided into two parts: a static scenario and a mobility scenario.

- Static scenario "**Scenario#1**": mobile wireless nodes remain in the same locations around 1-4 BS(s).
- Mobility scenario "**Scenario#2**": 2 BSs are used only and randomly surrounded by 10 wireless nodes that move individually around the whole simulation area based on the Random Waypoint mobility model, with speed selected between 1 - 10 m/s and 0 pause time.

Audio and video traffic will be sent from the BS(s) to the mobile devices in both scenarios. Mobile nodes may serve as intermediate nodes or relays in an ad hoc fashion. The other simulation parameters considered for the study are given in **Table 1**. **Table 2** explains the abbreviations used in the subsequent result graphs to denote specific protocols.

Each simulation result is the average of 10 simulation runs with different node locations which are chosen randomly by Bonnmotion [21]. In addition, 95% confidence intervals are used for the simulation results. For evaluating audio and video streaming applications' quality, two main factors need to be kept in mind: 1) the highest acceptable latency for the audio and video streaming applications was set as 200 ms, based on [22]; and 2) higher PDR generally results in higher audio and video quality. Therefore, to evaluate the performance of the NC protocols running audio and video streaming applications, three metrics were used:

PDR: the ratio of the total number of packets received to the number of packets meant to be received.

Latency: the average amount of time elapsed between the source sending a packet and all receivers decoding the packet.

Table 1. System Parameters.

Parameters	Routing			
	SMF	PDP	BCast	MAC Protocol
Radio-propagation model		Two ray ground		
MAC Protocol		IEEE 802.11b		
Network coding type	NO ^a , XOR or RS		NO ^a	RLNC
Link capacity (Mbps)		54		
Packet Size (Bytes)		208 ^b and 512 ^c		
Data Rate (in Kbps)		64 ^b and 2000 ^c		
Traffic Type		UDP		
Simulation time(s)		100		
Number of BS	1	2	3	4
Location of BSs (x, y)	1:(360,360)	1:(180,360) 2:(540,360)	1:(180,180) 2:(360,540) 3:(540,180)	1:(180,180) 2:(180,540) 3:(540,180) 4:(540,540)

^a: NO: No network coding technique used; ^b: values specify an MPEG audio streaming application [19]; ^c: values specify a video streaming application [20].

Table 2. Figure Abbreviations.

Abbreviations	Full Routing Protocol Name
PDP_nocode	PDP protocol without network coding
PDP_rscode	PDP protocol with Reed Solomon network coding
PDP_xorcode	PDP protocol with XOR network coding
SMF_nocode	SMF protocol without network coding
SMF_rscode	SMF protocol with Reed Solomon network coding
SMF_xorcode	SMF protocol with XOR network coding
BCast	Broadcast Protocol
RLNC	Random Linear Network Coding Protocol

Jitter: the variance of the latency between packets from the same data flow.

3.1. Audio Streaming Application Results

This sub-section discusses the results of the tested protocols' performance simulating an audio streaming application.

Figures 6 and 7 show the PDR results for Scenario#1 and Scenario#2, respectively. From the results, it is clear that PDR for the RLNC protocol is the highest. In Scenario#1, the PDR of the RLNC protocol decreases slightly from 100% to 96% as the number of BSs (senders) increases, but still remains the highest compared to the PDR of the other protocols. The reason is that SMF and PDP protocols without NC techniques flood the network with packets, and the BCast protocol wastes a lot of bandwidth in discovery node procedures. As a result, increasing the number of BSs will increase the delivery of redundant packets, which increases congestion in the network; hence the PDR decreases. Moreover, by increasing the number of BSs, the PDR of SMF and PDP protocols with NC techniques is negatively impacted as the receiver needs more information in order to decode the received encoded packets, where the needed informa-

tion is not always available (the opportunistic overhearing of neighbor receptions assumes that no packets are lost due to collisions). However, the RLNC protocol depends on the coding coefficients, rather than the overhearing opportunity, which are added in the headers of the encoded packets during the encoding procedure to decode the encoded packets. Thus, the RLNC protocol is not affected by increasing the number of BSs, as long as sufficient innovative packets are received eventually.

In Scenario#2, the PDR of the RLNC protocol approximately reaches 100% over various speeds. On the other hand, the PDR results of the remaining protocols are overlapping and decrease with increasing speed of the mobile wireless nodes (~82% for BCast and ~80% for the other protocols at 10 m/s). The reason is that BCast, SMF and PDP depend on neighbor information. As a result, increasing the speed of the wireless nodes will negatively affect the delivery of the packets, as the wireless mobile nodes keep moving around in the simulation area, reducing the accuracy of neighbor knowledge. Similarly, SMF and PDP protocols with NC techniques depend on the opportunistic listening mechanism (to collect information to decode the encoded packets). When a new node moves around the neighborhood, neighbors do not know what packets that node received before, which

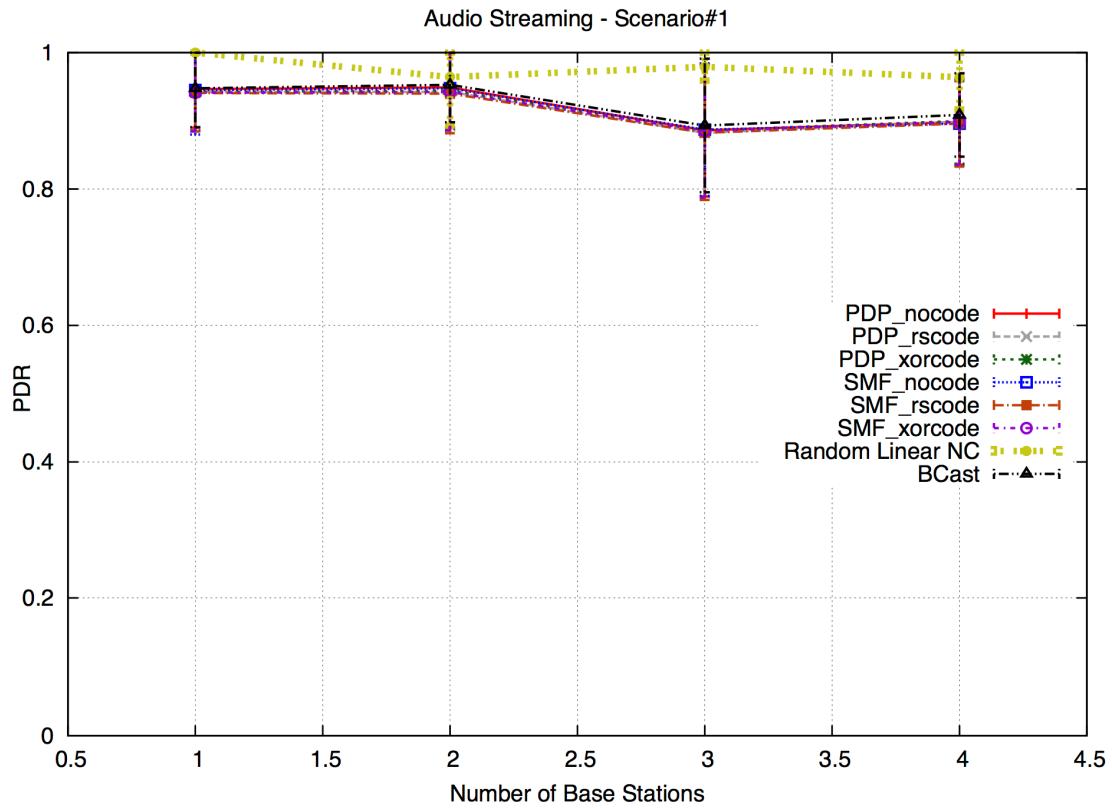


Figure 6. PDR for Scenario#1.

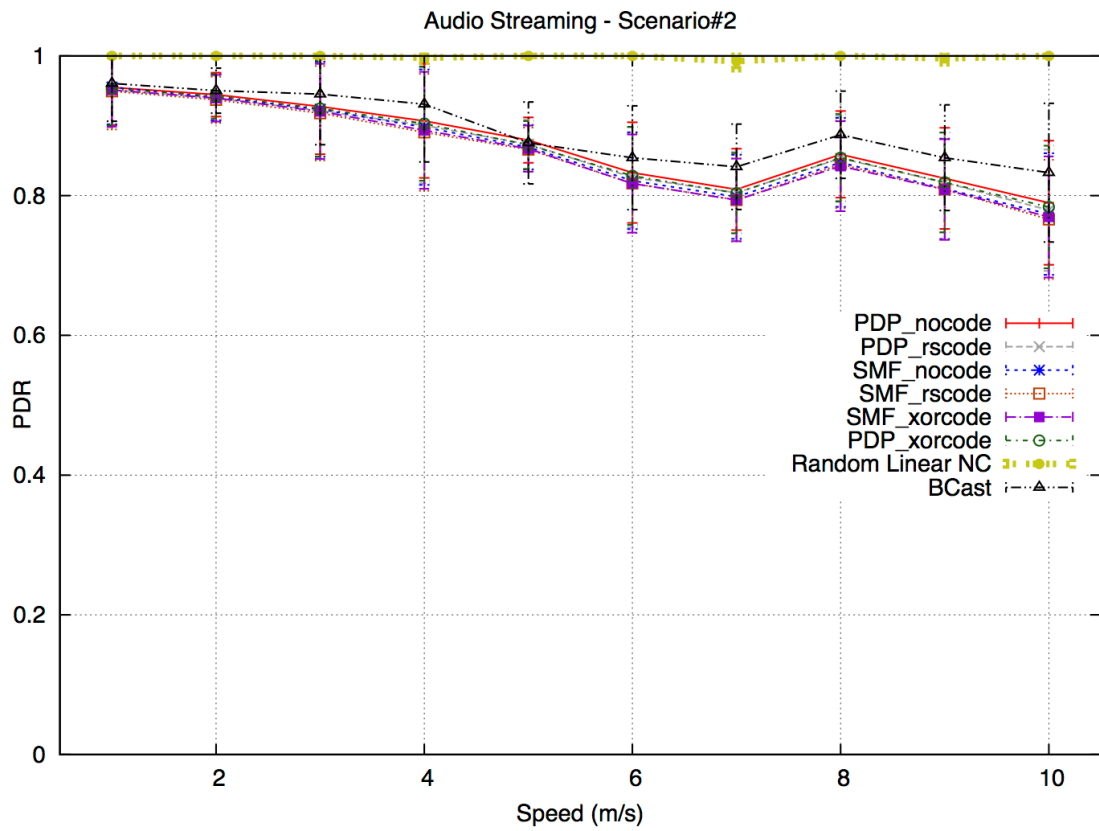


Figure 7. PDR for Scenario#2.

then results in a reduction of coding opportunities. This leads to a decline of the PDR. On the other hand, the RLNC protocol is not affected by the speed of the mobile wireless nodes because of reasons similar to the previous scenario.

Figures 8 and 9 show the latency results for Scenario #1 and Scenario#2, respectively. From the results, it is apparent that the latency for the RLNC protocol is the lowest and has an acceptable value (below 200 ms) for audio streaming application. In Scenario#1, the latency values slightly increase for all tested protocols as the number of BSs increases. However, the latency result for RLNC remains the lowest compared to all other protocols and is in the acceptable range (200 ms) for audio streaming application. The reason is that the sender (a relay or a BS) encodes and mixes a large number of packets into a single encoded packet and delivers it to the receiving nodes in a shorter delivery time. The latency results of BCast, SMF and PDP without NC increase and remain acceptable until the number of BSs exceeds two. The reason is that the number of flooded packets in the network increases by increasing the number of BSs, which increases the queue size, increasing the end-to-end latency. The latency results of SMF and PDP protocols with NC techniques are substantially higher (approximately 2.5 times) than the latency of the other tested protocols and are not acceptable for audio streaming ap-

plications due to the large size of the queue when the number of BSs increases and the additional protocol-specific delay. This additional delay is caused by the protocols to buffer packets until a protocol-specific timeout occurs or the buffer is full. Only then will the protocol explore if there are coding opportunities. The timeout value could be reduced, but that will result in a reduction of coding opportunities.

In Scenario#2, the latency values of the tested protocols are not affected by the speed of the wireless nodes. Using the RLNC protocol, the sender encodes and mixes a large number of packets into a single encoded packet and delivers it to the receiving nodes in a shorter delivery time. The latency values for BCast and PDP & SMF protocols without NC are much higher than the RLNC protocol's latency (but below 200 ms). The reason is that packets are queued in only a few intermediate nodes (relays), increasing the queue length and latency. The latency of SMF and PDP protocols with NC techniques is not acceptable for audio streaming application, as the results are higher than 200 ms. The reason for the high latency is the additional protocol-specific delay already discussed in Scenario#1. Moreover, the latency values for BCast is even lower (less by approximately 25 msec) than the latency values for PDP & SMF protocols without NC. The reason is that the nodes have the ability of overhearing other nodes retransmitting the missing or

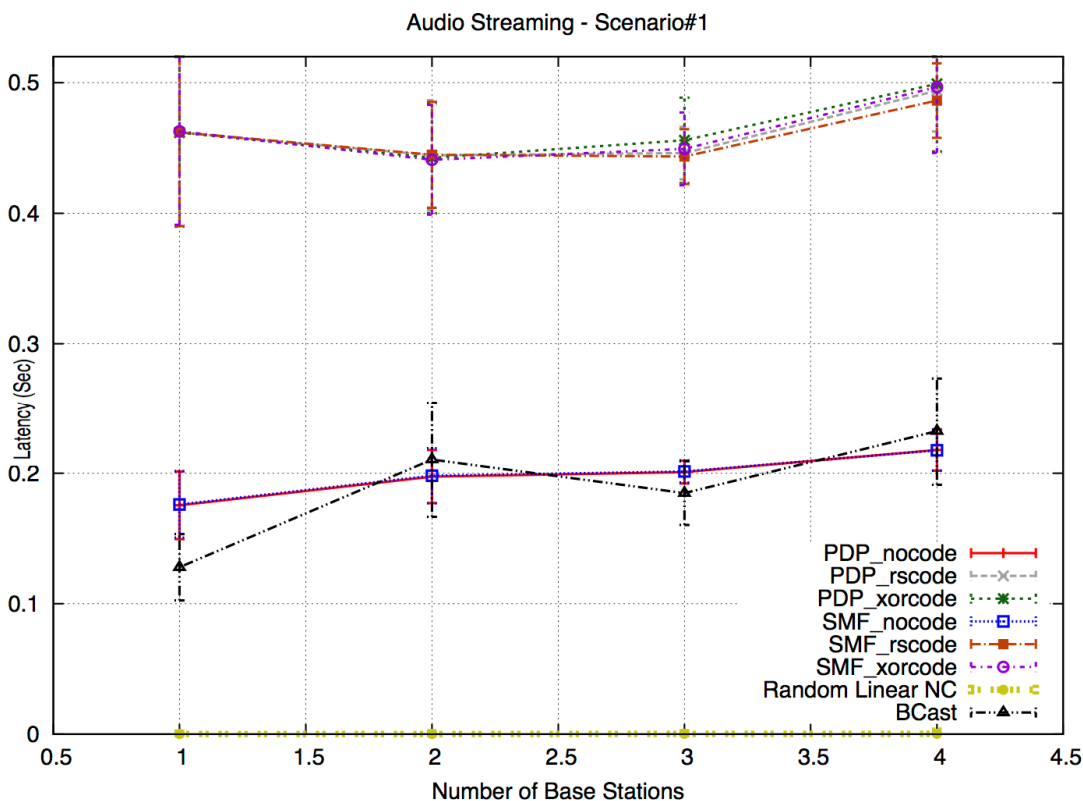


Figure 8. Latency for Scenario#1.

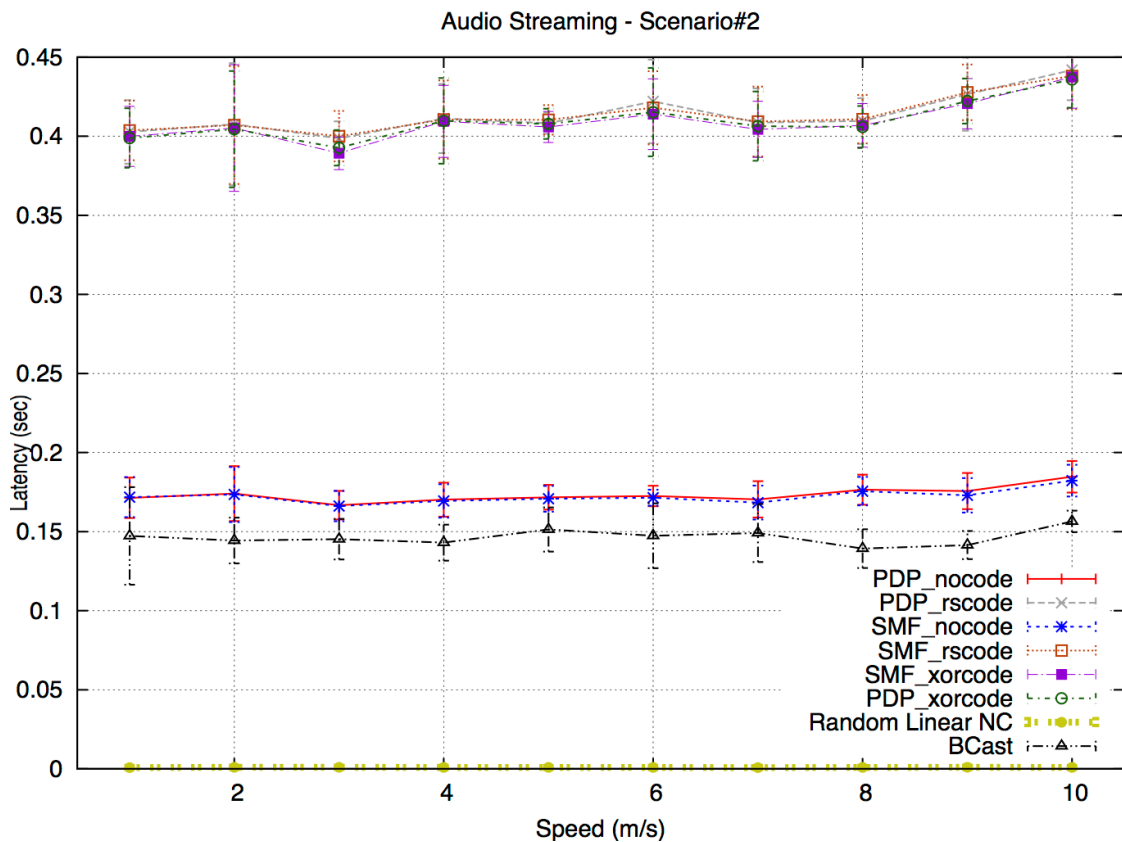


Figure 9. Latency for Scenario#2.

delayed packets. This overhearing ability makes these nodes cancel their own retransmissions if they know that another node already has retransmitted these packets. As a result, the length of nodes' queues is reduced (compared to PDP and SMF without NC), which leads to lower latency.

Figures 10 and 11 show the jitter results for Scenario #1 and Scenario#2, respectively. The jitter values of SMF and PDP with NC are the highest compared to the other jitter values. The jitter is again caused by the extra buffering: a node has to wait for the timeout if it does not have a full buffer to transmit packets earlier. Moreover, with an increasing number of BSs the buffers are filled up faster, so the jitter for PDP/SMF with NC decreases until nodes start experiencing congestion. Without NC, SMF/PDP/BCast fill up buffers unevenly (forwarders/MPRs have more packets in buffers) which leads to an increase in the jitter. On the other hand, RLNC has the lowest jitter in both scenarios as the sender encodes and mixes a large number of packets into a single encoded packet and delivers it to the receiving nodes in a shorter delivery time. Packet transmissions are spread out evenly among all nodes, based on the forwarding factor, avoiding the formation of long queues at a subset of node. As a result, this will not only decrease the delay but also result in more consistent per-packet delay/a reduction in the

jitter of the network.

Figures 12 and 13 show the PDR results for Scenario#1 and Scenario#2, respectively. The PDR of RLNC remains the highest. On the other hand, the PDR confidence intervals of the remaining protocols are overlapping and decrease when increasing the number of BSs. Figure 12 shows that the PDR results for the tested protocols (excluding RLNC) vary from approximately 26% to 71%. The results show that when increasing the size of data packets and the data rates to meet the requirements of the video streaming application, the PDR of RLNC remains the highest and stays above 90%, whereas the PDR of other tested protocols drops below 80% for one BS and drops even lower for four BSs, reaching as low as 25%.

3.2. Video Streaming Application Results

This sub-section discusses the results of the tested protocols' performance when simulating a video streaming application.

The results for video streaming application (including PDR, Latency and Jitter) for Scenario#2 were only tested with RLNC, as it was shown to be the best protocol for video streaming in Scenario#1. In Scenario#2, the PDR results for video streaming applications (which have

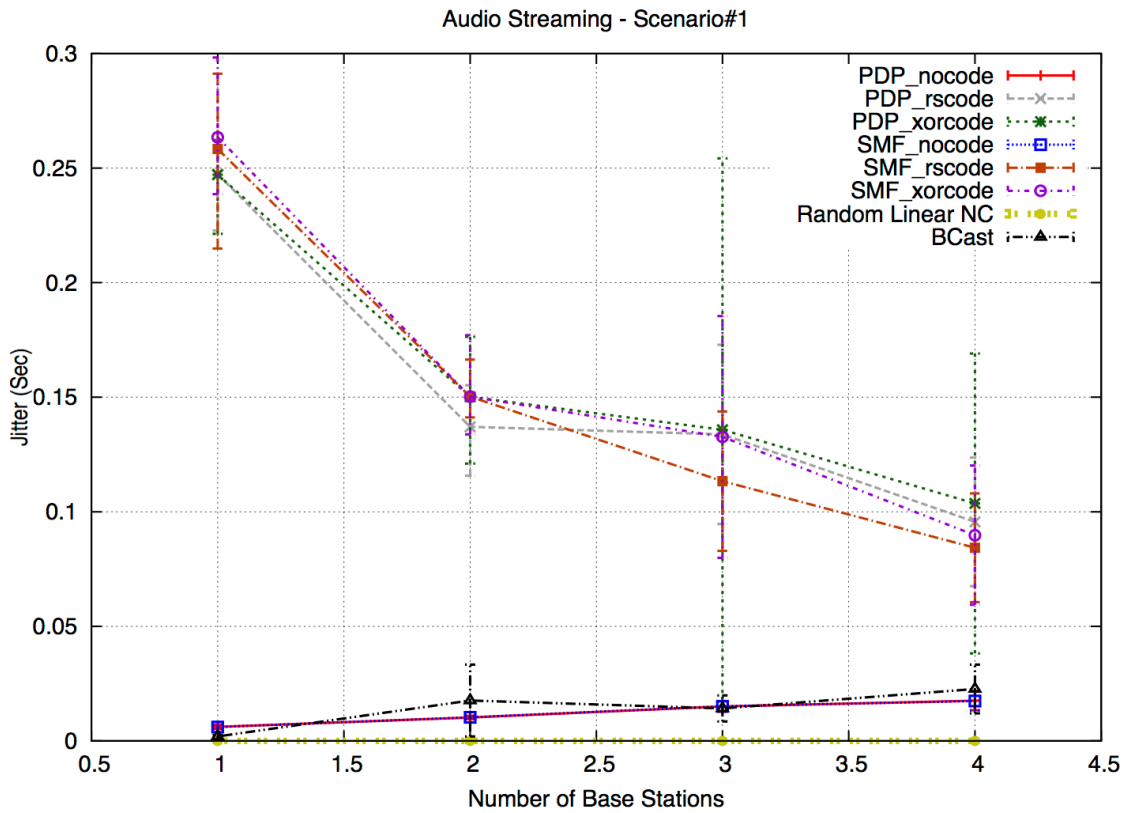


Figure 10. Jitter for Scenario#1.

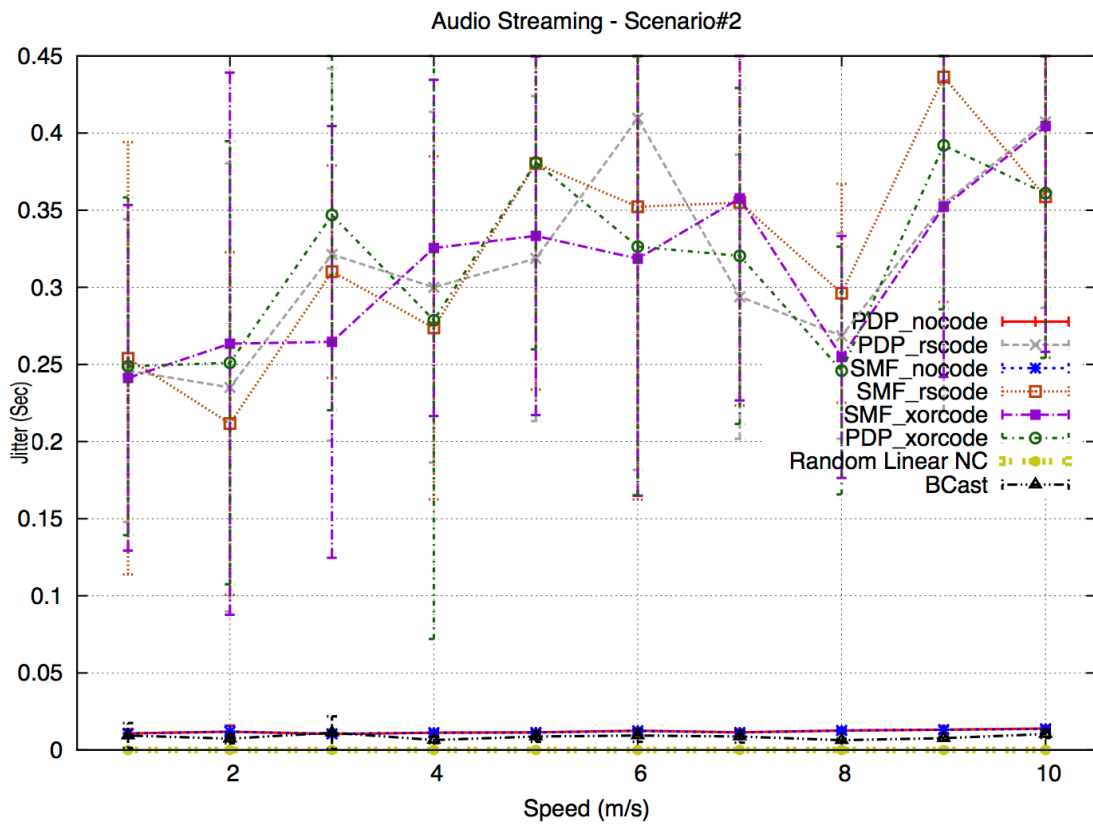


Figure 11. Jitter for Scenario#2.

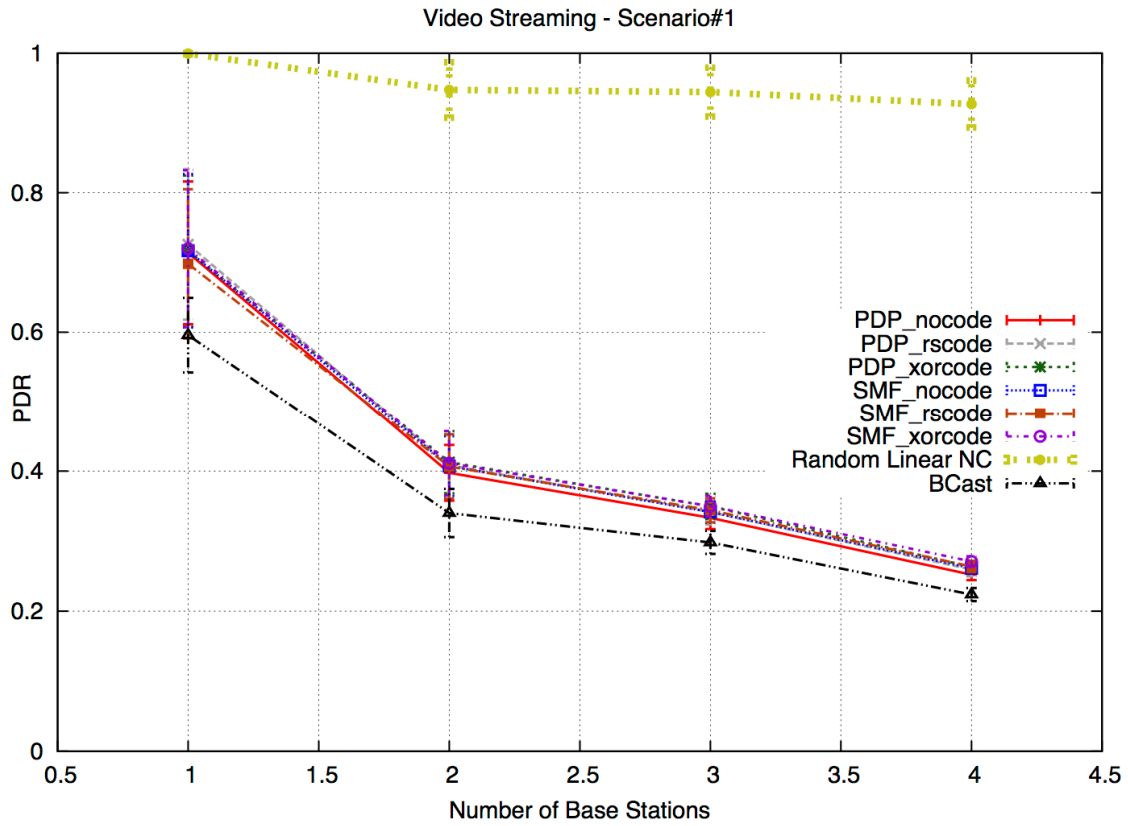


Figure 12. PDR for Scenario#1.

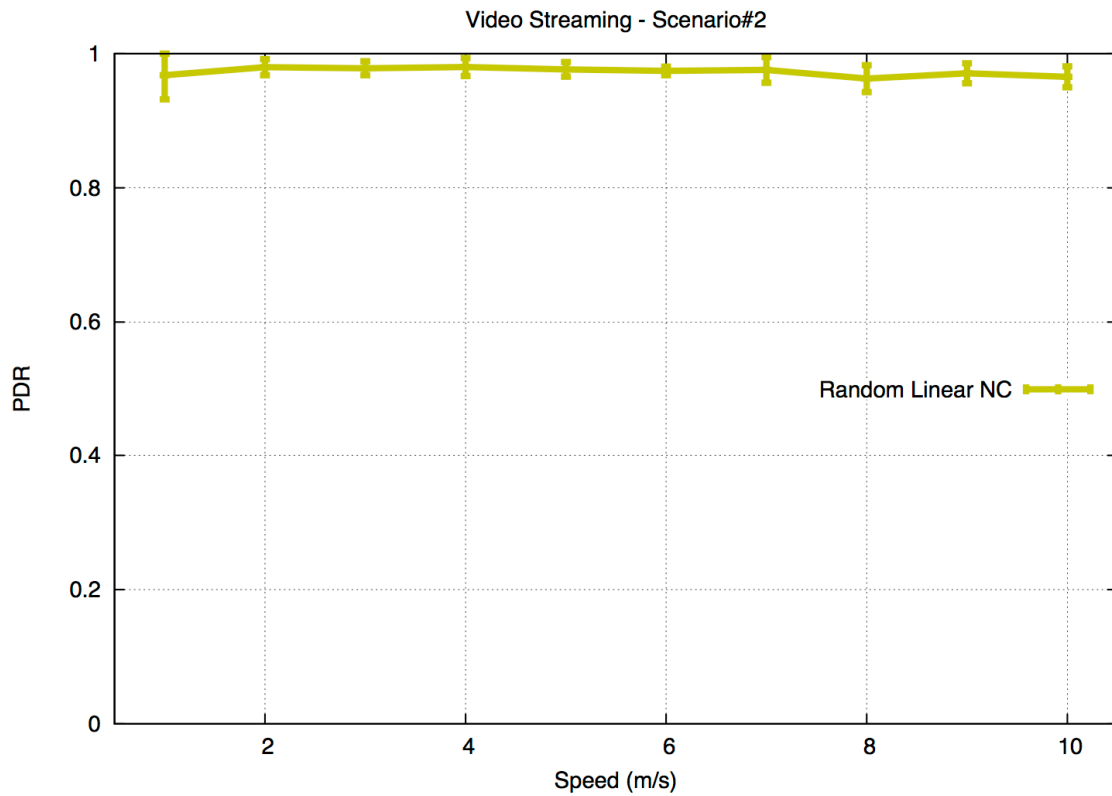


Figure 13. PDR for Scenario#2.

higher data rate and larger packet size compared to audio streaming application) are not impacted by mobility and maintain a steady and high PDR (greater than 90%) as shown in **Figure 13**. The reason for this, as discussed in the audio streaming application results, is that the RLNC protocol depends on the coefficients, which are added in the headers of the encoded packets during the encoding procedure to decode the encoded packets. Therefore, the protocol performance will not be affected by the speed of the mobile wireless nodes, as long as a sufficient number of innovative packets is eventually received by each node.

The results for video streaming application (including PDR, Latency and Jitter) for Scenario#2 were only tested with RLNC, as it was shown to be the best protocol for video streaming in Scenario#1. In Scenario#2, the PDR results for video streaming applications (which have higher data rate and larger packet size compared to audio streaming application) are not impacted by mobility and maintain a steady and high PDR (greater than 90%) as shown in **Figure 13**. The reason for this, as discussed in the audio streaming application results, is that the RLNC protocol depends on the coefficients, which are added in the headers of the encoded packets during the encoding procedure to decode the encoded packets. Therefore, the protocol performance will not be affected by the speed of the mobile wireless nodes, as long as a sufficient number of innovative packets is eventually received by each node.

Figures 14 and 15 show the latency results for Scenario#1 and Scenario#2, respectively. In Scenario#1, the latency results of RLNC for video streaming application remain the lowest compared to other tested protocols as shown in **Figure 14**. The reason is that the sender delivers the encoded packet to the receiving nodes in a shorter delivery time as packet transmissions are spread out evenly among all nodes, based on the forwarding factor, avoiding the formation of long queues at a subset of node. On the other hand, the latency results of the remaining protocols decrease as the number of BSs increases and becomes more acceptable for video streaming broadcast application. The reason is that the data rate of the video streaming application is very high (almost 2000 kbps). As a result, the packets may be queued faster (compared to the audio streaming application) in different intermediate nodes or forwarded faster to different nodes in BCast, SMF and PDP. These protocols utilize the neighborhood information for reducing the number of duplicate retransmissions while forwarding a broadcast packet. This reduces the number of packets that are queued in the intermediate nodes. Thus, increasing the data rate will increase the transmissions speed by intermediate nodes for lost packets. In other words, the packets will be delivered faster compared to the packets in the case of audio streaming application. Furthermore, the latency values for BCast are even lower than the latency values for

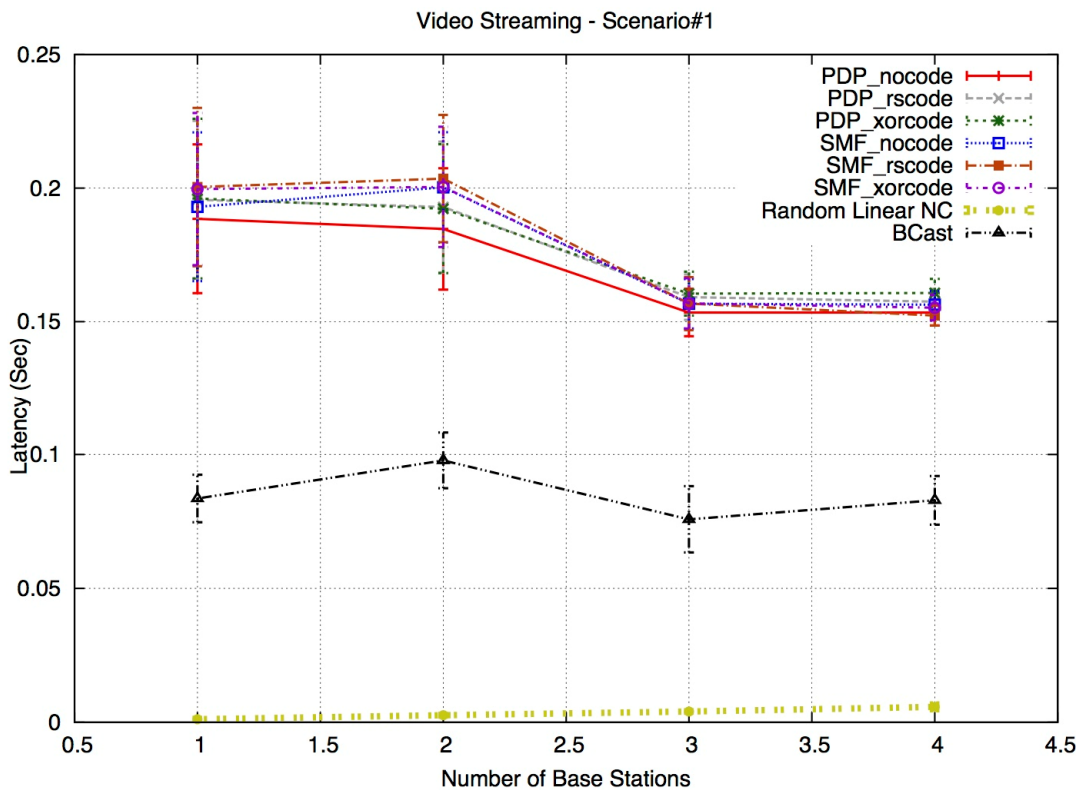


Figure 14. Latency for Scenario#1.

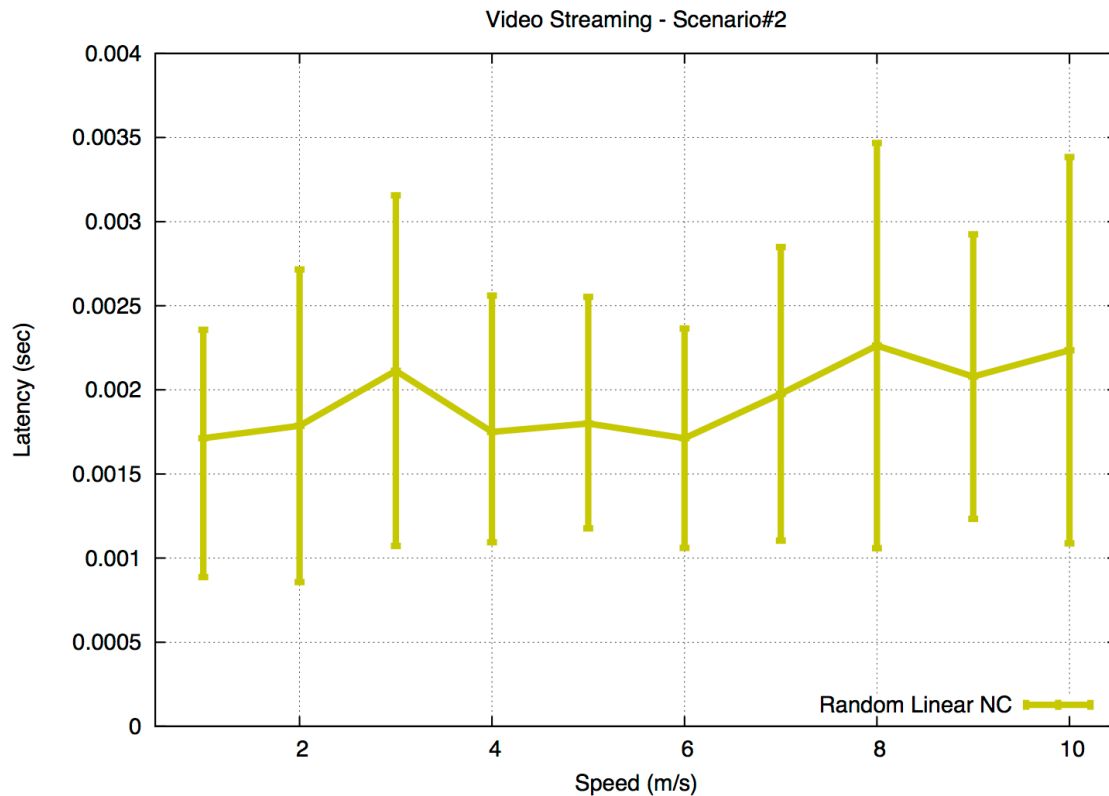


Figure 15. Latency for Scenario#2.

PDP & SMF protocols without NC. The reason of this is the ability of the nodes to overhear other nodes retransmitting the missing or delayed packets.

As a result, these nodes cancel their own retransmissions if they overhear another node already has retransmitted these packets. Hence, the length of nodes' queues is reduced which shortens the latency values to approximately 50% (compared to PDP and SMF without NC). Moreover, in case of SMF and PDP protocols with NC techniques, the high data rate of the video streaming application fills up the internal buffer with the required number of data packets faster, resulting in faster transmissions by intermediate nodes. Coding opportunities are explored before the timer expires, resulting in faster packet forwarding and faster packet reception at the receiving nodes.

In Scenario#2, it is apparent that the latency for the RLNC protocol, as shown in **Figure 15**, has again a low and acceptable value (below 200 ms) for video streaming application. This is due to the fact that the sender encodes and mixes a large number of packets into one encoded packet and delivers it to the receiving node in a shorter delivery time. Moreover, the RLNC protocol does not use any additional buffering to wait for extra information to be used to decode the received encoded packets, as all needed coefficients are stored at the header of the encoded packet in RLNC protocol. And unlike other

tested protocols, traffic is not concentrated on the MPRs/Forwarders, but spread across all nodes, resulting in shorter queues.

Figures 16 and 17 show the jitter results for Scenario#1 and Scenario#2, respectively. For Scenario#1, the jitter values of SMF and PDP with NC are the highest compared to the other jitter values. The jitter is caused by the extra buffering as discussed in the previous section already. Furthermore, with an increasing number of BSs these buffers fill up faster, so the jitter for PDP/SMF with NC drops until nodes start experiencing congestion. On the other hand, RLNC has the lowest jitter in both Scenario#1 and Scenario#2 as the receiving nodes use the coefficients to decode the received encoded packets. Similar to the reason of the jitter results for the audio streaming application, the sender encodes and mixes a large number of packets into a single encoded packet and delivers it to the receiving nodes in a shorter delivery time. Packet transmissions are spread out evenly among all nodes, based on the forwarding factor, avoiding the formation of long queues at a subset of node. Also, innovative packets are transmitted right away in the RLNC protocol. Thus, this will not only decrease the delay but also result in more consistent per-packet delay/a reduction in the jitter of the network. The jitter results of BCast, SMF, and PDP without NC also have low values but higher than RLNC. The reason for this result is that

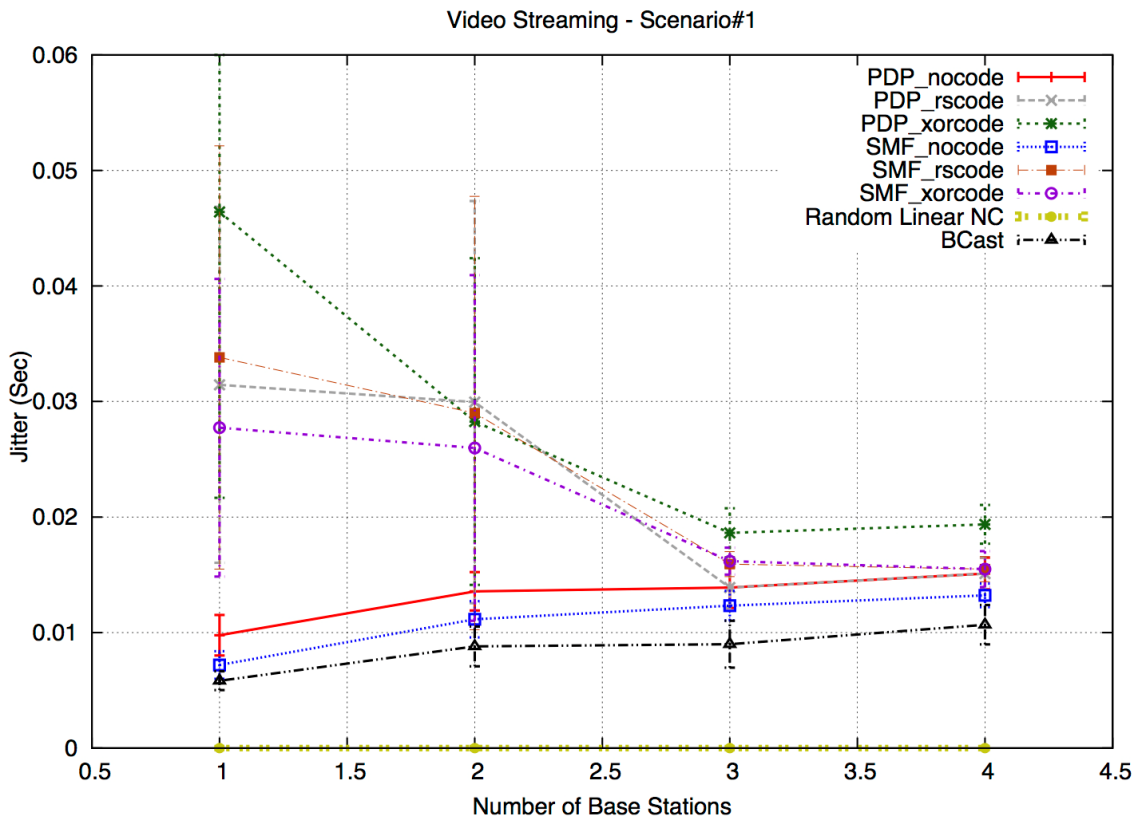


Figure 16. Jitter for Scenario#1.

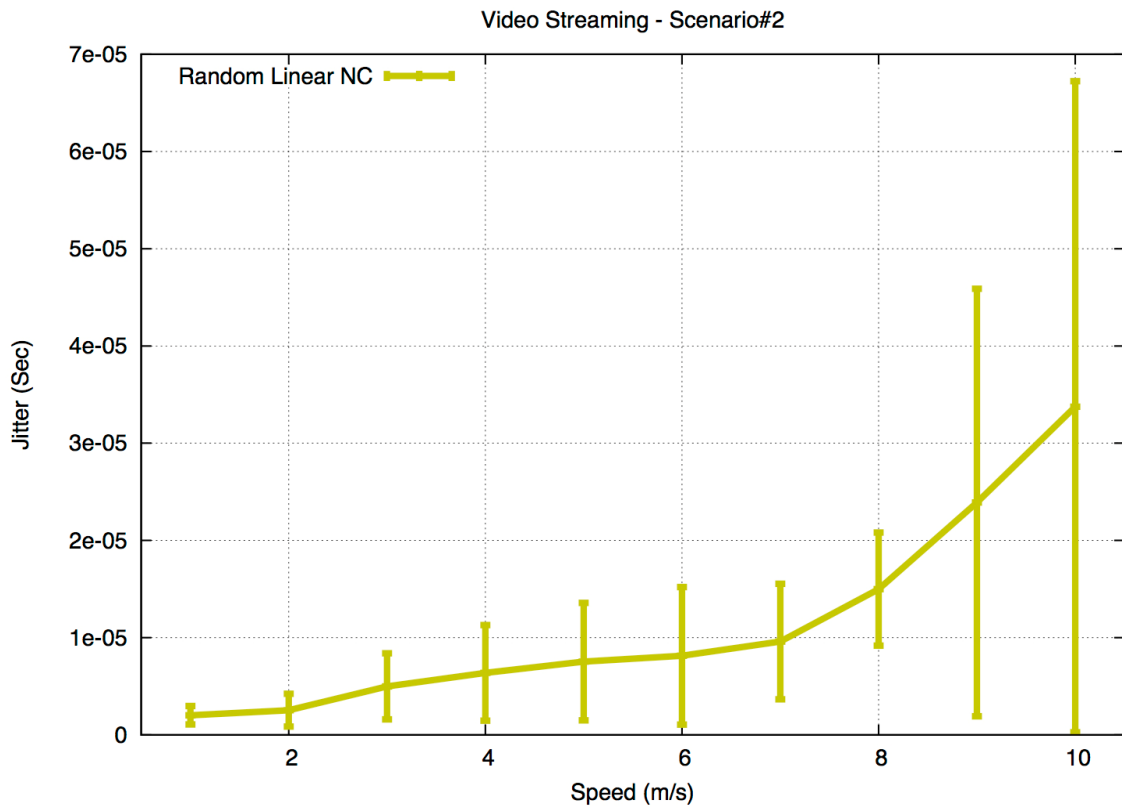


Figure 17. Jitter for Scenario#2.

some packets may be queued in different intermediate nodes or forwarded to different nodes. Buffers for BCast, SMF, and PDP are filled up unevenly (forwarders/MPRs have more packets in the buffer). Thus, latency is affected by these factors which impact the jitter.

4. Conclusions and Future Work

This paper explores ways to efficiently support video and audio streaming applications which are becoming popular. The paper compares a number of efficient broadcast schemes, either based on packet forwarding or network coding (or a combination of both). The simulation results show that the best protocol to be used for audio streaming application is a protocol based on RLNC as it delivers the highest PDR and the lowest latency and jitter. As a result, RLNC improves the overall performance of the multimedia streaming applications. On the other hand, the use of XOR network coding and RS network coding techniques negatively affects the performance of audio streaming application in terms of latency and jitter; latency is considered one of the main performance metrics to evaluate the quality of the multimedia streaming applications.

Moreover, it is clear that protocols that transmit data through a backbone (BCast, SMF and PDP) suffer from increased jitter. Also, protocols that buffer packets to exploit coding opportunities will have poor latency. Therefore for NC-based protocols to have low latency, whenever a node receives an innovative packet, the packet should immediately get propagated in the network to keep the latency small. However, many NC-based protocols do not forward the packets immediately.

Finally, protocols that do not optimize the required number of packet transmissions run into the problem of increasing queue sizes and eventually network congestion, resulting in poor PDR. Going forward, we are currently conducting further studies on HDTV and SDTV applications.

5. Acknowledgements

The authors would like to thank the Ontario Centers of Excellence (OCE) and EION Wireless for their support.

REFERENCES

- [1] R. Ducey, "Multimedia Broadcasting and the Internet." http://www.iif.hu/inet_96/b3/b3_2.htm
- [2] O. Oyman, "Enabling Mobile Video Services over WiMAX and LTE", *Proceedings of IEEE Vehicular Technology Society of the Institute of Electrical and Electronics Engineers*, September 2010.
- [3] Arbitron/Edison Media Research, "Internet and Multimedia 11: New Media Enters the Mainstream," 2003. <http://www.radiostreamingnews.com/2010/02/streaming-radio-growth-charts.html>
- [4] J. Khan and R. Zaghal, "Jitter and Delay Reduction for Time Sensitive Elastic Traffic for TCP-interactive Based World Wide Video Streaming over ABone," *Proceedings of the 12th IEEE International Conference on Computer Communications and Networks*, Dallas, 22 October 2003, pp. 311-318.
- [5] R. Ahlswede, et al., "Network Information Flow," *IEEE Transaction of Information Theory*, Vol. 46, No. 4, 2000, pp. 1204-1216. [doi:10.1109/18.850663](https://doi.org/10.1109/18.850663)
- [6] S. R. Li, R. W. Yeung and N. Cai, "Linear Network Coding," *IEEE Transaction of Information Theory*, Vol. 49, No. 2, 2003, pp. 371-381. [doi:10.1109/TIT.2002.807285](https://doi.org/10.1109/TIT.2002.807285)
- [7] T. Kunz and L. Li, "Broadcasting in Multihop Mobile Tactical Networks: To Network Code or Not," *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference*, Caen, 28 June 2010, pp. 676-680.
- [8] T. Kunz, "Reliable Multicasting in MANETs," Contractor Report, Communications Research Centre, Ottawa, 2003.
- [9] T. Kunz, "Implementing BCAST," Contractor Report, Communications Research Centre, Ottawa, 2004.
- [10] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks," *Proceedings of IEEE International Multi Topic Conference (INMIC '01)*, Lahore, 30 December 2001, pp. 62-68.
- [11] J. Macker, I. Downard, J. Dean and B. Adamson, "Evaluation of Distributed Cover Set Algorithms in Mobile Ad Hoc Network for Simplified Multicast Forwarding," *ACM SIGMOBILE Mobile Computing and Communications Review*, Vol. 11, No. 3, 2007, pp. 1-11. [doi:10.1145/1317425.1317426](https://doi.org/10.1145/1317425.1317426)
- [12] A. Rahman, M. E. Hoque, F. Rahman, S. K. Kundu and P. Gburzynski, "Enhanced Partial Dominant Pruning (EPDP) Based Broadcasting in Ad Hoc Wireless Networks," *Journal of Networks*, Vol. 4, No. 9, 2009, pp. 895-904. [doi:10.4304/jnw.4.9.895-904](https://doi.org/10.4304/jnw.4.9.895-904)
- [13] L. Li, R. Ramjee, M. Buddhikot and S. Miller, "Network Coding-Based Broadcast in Mobile Ad Hoc Networks," *Proceedings of INFOCOM*, Anchorage, 6 May 2007.
- [14] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard and J. Crowcroft, "XORs in the Air: Practical Wireless Network Coding," *Proceedings of IEEE/ACM Transactions on Networking*, Vol. 16, No. 3, June 2008, pp. 497-510.
- [15] E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, "A Proactive Network Coding Strategy for Pervasive Wireless Networking," *Proceedings of IEEE GLOBECOM*, Washington DC, 26-30 November 2007, pp. 5271-5276.
- [16] E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, "On MAC Scheduling and Packet Combination Strategies for Random Network Coding," *Proceedings of IEEE ICC*, Glasgow, 24-28 June 2007, pp. 3582-3589.
- [17] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi and B. Leong, "A Random Linear Network Coding Approach to Multicast," *IEEE Transaction on Information Theory*, Vol. 52, No. 10, 2006, pp. 4413-4430. [doi:10.1109/TIT.2006.881746](https://doi.org/10.1109/TIT.2006.881746)

- [18] P. Vingelmann, P. Zanaty, F. H. Fitzek and H. Charaf, "Implementation of Random Linear Network Coding on Opengl-Enabled Graphics Cards," *Proceedings of European Wireless*, Aalborg, 17-20 May 2009.
- [19] MPEG-2 Systems.
<http://mpeg.chiariglione.org/faq/mp2-sys/mp2-sys.htm#mp2-1>
- [20] "IEEE 802 Tutorial: Video over 802.11," 2010.
<http://www.ieee802.org/802tutorials/index.htm>
- [21] BonnMotion Mobility Generator.
<http://www.informatik.unibonn.de/IV/BonnMotion>
- [22] "Audio/Video Streaming over 802.11."
<http://www.ieee802.org/802tutorials/.../video%20over%20802%2011%20Tutorial-final.ppt>