Reliable Videos Broadcast with Network Coding and Coordinated Multiple Access Points

Pouya Ostovari and Jie Wu
Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122
Email: {ostovari, jiewu}@temple.edu

***Abstract*—As the popularity of wireless devices (e.g. smartphones and tablets) increases, watching videos over the Internet is becoming a main device application. Two important challenges in wireless communications are the unreliability of the wireless links and the interference among the wireless links. In order to exploit reliable video multicast, forward error correction and network coding can be used. In this paper, we propose a reliable video multicast method over wireless networks with multiple Access Points (AP). Using multiple APs, the multicast receivers can benefit from both spatial and time diversities, which results in more reliable transmissions. In order to utilize the shared wireless network efficiently, we propose a resource allocation algorithm. We show that a systematic concurrent transmission of the interfering AP nodes can enhance the total system performance and provide fairness to the client nodes. Therefore, in contrast with the previous resource sharing methods, which only permit the AP nodes that do not interfere with each other to transmit concurrently, we allow the interfering nodes to transmit concurrently, and we propose a two-phase resource allocation algorithm to further enhance the system utility. We show the effectiveness of our proposed method through extensive simulations.**

***Index Terms*—Streaming, video streaming, network coding, wireless networks.**

I. Introduction

In recent days we are witnessing a fast development in the hardware technologies of mobile devices, such as tablets and smartphones. As a result of these advances, the mobile devices are becoming very popular. With this increase in the popularity of these devices, watching videos over the Internet is becoming one of their main applications. Some recent studies show that the most dominant form of data traffic on the Internet is multimedia streaming. For example, YouTube and Netflix produce 20-30% of the web traffic on the Internet [1], [2]. Because of the popularity of wireless devices, a large portion of the users that watch videos use their wireless devices. One of the main challenges in wireless networks is the unreliability of the links. This challenge becomes more important in the case of video streaming, in which missing transmitting packets can greatly affect the quality of the videos received by the users.

Providing reliable transmissions was widely studied in the previous work. One of the common mechanisms used in providing reliable transmissions is using feedback messages, and Automatic Repeat reQuest (ARQ) is the most frequently used approach [3]. In order to reduce the overhead of ARQ messages, FEC (Forward Error Correction) and ARQ messages are combined in Hybrid-ARQ methods [4]–[6]. In addition to these methods, network coding (NC) techniques [7], such as random linear network coding [8] (RLNC) and Fountain codes [9], [10] can be used to provide reliable transmissions without the need to feedback messages. For example, in RLNC, the original packets are mixed together using random coefficients. The source node transmits randomly coded packets, and a destination node is able to decode all of the coded packets and retrieve the original packets once it receives a sufficient number of coded packets. In this way, the source node does not need to know which packets are missed by the destinations.

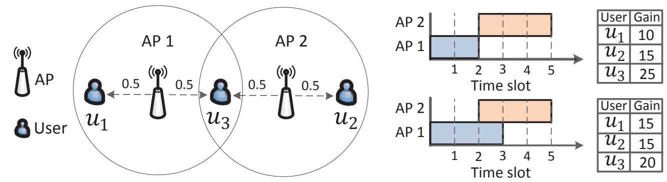


Fig. 1. The motivation example. The top right chart: non-overlapped transmissions. The bottom right chart: overlapped transmissions

Most of the existing research on video multicast using application layer FEC only considers a single access point (AP). In [11], the authors propose a video multicast scheme using multiple neighboring APs. In order to improve the reliability of video multicast, the authors use multiple coordinated APs to serve the users in a multicast session. In a wireless network where multiple APs are scattered in an area, the users that are located at the cell boundaries might experience low packet delivery rates. Using multiple APs can help to serve each user with different APs and enhance the performance of the video streaming. In order to prevent interference among the APs, the existing scheduling works to schedule the APs such that the interfering APs do not transmit at the same time. When we do not allow the APs that interfere with each other to transmit at the same time, the users that are located in the intersection of the coverage area will receive more packets than the users that are covered by a single AP.

Consider the example in Figure 1. Assume that the receiving rate of the users u_1 and u_2 from the APs 1 and 2 are 5 per time slot. Also, user u_3 can receive 5 packets per time slot from APs 1 and 2. Moreover, we want to schedule 5 time slots for these APs. If we schedule 2 and 3 time slots for AP 1 and 2, respectively, users u_1 and u_2 will receive 10 and 15 packets. Moreover, user u_3 will get a total of 25 packets. Assume that the number of packets that are coded together is 15. In the case of this non-overlapped scheduling, user u_1 will not be able to decode the coded packets. Now let us consider an overlapped transmission. We assign 2 slots for APs 1 and 2. Moreover, we assign 1 slot for the APs to transmit concurrently, during which user u_3 will not be able to receive any packets correctly due to the interference. Consequently, users u_1, u_2, and u_3 will receive 15, 15, and 20 packets, respectively. As a result, all of the users will be able to decode the coded packets and watch the video.

Motivated by the example in Figure 1, we propose a two-phase algorithm, in which we allow concurrent transmission of the interfering APs. In the first phase, we schedule the AP nodes such that the interfering nodes are not allowed to transmit at the same time. In the second phase, we modify the scheduling by permitting some overlapped transmissions if they can increase the fairness of the system or the number of received packets by the users.

The remainder of the paper is organized as follows. We provide a background on network coding and review the related work in Section II. In Section III we introduce the system setting and our objective. In Section IV we propose our video streaming methods with network coding. The simulation results are presented in Section V. Finally, we conclude the paper in Section VI.

II. Background and Related Work

A. Reliable Transmission

One of the main challenges in wireless networks is the unreliability of the links. In order to provide reliable transmissions in lossy environments, certain mechanisms, such as feedback messages, are used. The most frequently used mechanism for addressing this challenge is Automatic Repeat reQuest (ARQ) [3]. The main drawback of ARQ is that it imposes some overhead. Feedback messages increase the energy consumption of the nodes. Moreover, in single radio cases, the sender node needs to stop transmissions to receive feedback, which increases the transmission delay. In order to solve this issue, Hybrid-ARQ approaches [5], [6] are proposed, which combine FEC (Forward Error Correction) with ARQ. Also, the complex RMDP approach [6] uses Vandermonde [12] code and ARQ to provide reliability.

All of the methods that use feedback messages incur some overhead. Moreover, in some applications the feedback messages are not feasible. For instance, in multicasting applications, implementing feedback messages is very costly; the reason is that, transmitting a feedback by each receiver waists a large portion of the transmission time. Rateless codes [9], [10], which are also known as fountain codes, are an efficient way to provide reliable transmissions without using feedback messages. In these methods, the sender transmits coded packets until all of the destinations receive enough coded packets to be able to decode and retrieve the original packets. In rateless codes, the sender node can potentially create an infinite number of coded packets. A destination needs to collect enough coded packet, regardless of which packets have been lost. As a result, in contrast with the ARQ method, only one acknowledgment from each destination node is sufficient for the sender to know that the destination received all of the original packets. If we have k original packets to transmit, a destination node needs to receive $1 + \beta$ coded packets to decode them [10]. Here, β is a small number and shows the overhead of the rateless codes. It is shown that as $k \to \infty$, the overhead goes to zero [13]; thus rateless codes are only efficient in the case that a large number of packets is transmitted. In our model, the transmitted data is video, which is delay-sensitive. As a result, we should partition the data into segments of few packets, and perform coding inside each segment; thus, fountain codes are not appropriate for our work.

It is clear that when a set of users watch the same video, unicasting an independent data stream to each user is not efficient, as the broadcasting nature of the wireless medium is not considered. For this reason, video multicasting has received a lot of attention from the community. A major challenge that should be addressed in video multicasting is the heterogeneous channel condition of users. Wireless users experience different packet delivery rates, which is due to different distances to the APs, interference among the wireless transmissions, and etc. In order to solve this problem, scalable video coding (SVC) [14], [15], also known as multi-layer coding [16], are proposed. In SVC, a video is partitioned to a base layer and a set of enhancement layers. The base layer is required to watch the video. In contrast, the enhancement layers can increase the quality of the received videos. Using SVC, the users with different channel qualities can receive a different number of layers, and watch the received videos with different qualities.

B. Network Coding

The first application of network coding (NC) [17]–[19] was in wired networks. It is shown in [20] that NC achieves the capacity for the single multicast session problem. The authors in [8] show that if we select the coefficients of the coded packets randomly, we can achieve the capacity asymptotically, with respect to the finite field size. This approach is called random linear NC (RLNC). In RLNC, the generated coded packets are linear combinations of the original packets over a finite field, and the coefficients of this linear combinations are random. Each coded packets has a form of $\sum_{i=1}^{k} \alpha_i \times P_i$. Here, P and α are the packets and random coefficients. When a source uses RLNC for data transmissions, it generates and transmits random coded packets. A destination is able to decode the coded packets and retrieve all of the original packets when it receives k linearly independent coded packets. In order to decode the coded packets, Gaussian elimination can be used. Similar to the case of using fountain codes, a destination only needs to transmit a single acknowledgment once it receives k linearly independent coded packets.

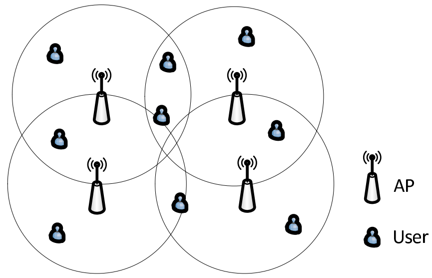


Fig. 2. The system model.

TABLE I
THE SET OF SYMBOLS USED IN THIS PAPER.

Notation	Definition
U/B	Set of user/AP nodes.
x_j	The fraction of time for which the jth AP is scheduled for transmission.
z_{kj}	The fraction of time that AP j is borrowed from AP k.
ϵ_{ji}	The loss rate of the link between the jth AP and the ith user.
$C(i)$	The set of AP nodes that cover the user i.
$N(j)$	The set of AP nodes that interfere with AP j.
r_i	The expected number of packets that user i receives from APs in $C(i)$.
b	Bandwidth of the AP nodes.
I_k	The kth independent set.
S	The set of independent sets.

III. SYSTEM MODEL

A. Setting and Objective

We consider an environment where the video servers forward a video stream to a set of neighboring APs that can help broadcast the video to all multicast users. We assume that APs and the video server are connected by wired links. These links are highly reliable and they are not the bottleneck. Consequently, we can assume that the video packets are ready at the APs to be transmitted to the users. There are m WiFi APs in our model that broadcast a video stream to a set of n wireless users, such as smartphones, tablets, or desktop computers. The system model and the set of symbols used in this paper are shown in Figure 2 and Table I, respectively.

As the transmissions are performed over wireless links, the transmitted video packets might be lost by some of the destination nodes. We represent the erasure probability of the link from AP node i to user j as ϵ_{ij}. Since using feedback mechanisms in broadcasting applications is a major challenge, we do not use feedback messages to report the lost or received packets. Each AP node has a circular coverage area. Moreover, the coverage area of the WiFi APs might overlap with the other APs. As a result, these WiFi devices will interfere with each other, and the user nodes that are located in the intersection area of these APs will not be able to receive the packets correctly if the APs concurrently transmit the packets.

Our goal in this paper is to provide a fair video multicast to the users by properly scheduling the AP nodes. In more details, we want to maximize the expected number of packets that are received by the users. Since feedback messages are not available in our model, we use RLNC to transmit the video packets to the users. Using RLNC, the AP nodes do not need to know which packets are lost or received by the users, and the users only need to receive a sufficient number of coded packets to be able to decode the coded packets.

It is typical in wireless networks to schedule the interfering wireless devices to perform transmissions at different times, such that they do not interfere with each other. However, motivated by the example in the introduction (Figure 1), overlapped transmission of the interfering APs might provide a more fair scheduling. As a result, we allow overlapped transmission of the AP nodes in our scheduling algorithms.

The problem of reliable broadcasting in the case of overlapped transmission can be modeled as a linear programming optimization. However, for m AP nodes, we have a total of 2^m sections that need to be scheduled. As a result, this problem is computationally intractable, and we use a two-phase heuristic algorithm to find an efficient scheduling. In the first phase, we schedule the AP nodes with the constraint that the interfering AP nodes should not transmit at the same time. Then, in the second phase, we try to add some overlapped transmissions to increase the number of packets that are received by the users that receive fewer packets than the other nodes and, potentially, these packets might not be sufficient to decode the coded packets and watch the video.

B. Interference Model

We consider two interference models: complete interference and non-complete interference among the APs, which are discussed in the following subsections.

1) Complete Interference Graph: In this model, each AP node interferes with all of the other AP nodes. In other words, the interference graph of the network is a complete graph. In order to prevent interference among the APs in the case that the interference graph is a complete graph, the AP nodes should not be scheduled at the same time.

2) Non-Complete Interference Graph: In the case that the coverage area of some AP nodes are disjoint, those AP nodes can be scheduled to transmit at the same time. In this case, the interference graph is not a complete graph.

IV. VIDEO MULTICASTING WITH NETWORK CODING

In the following subsections, we first describe our NC scheme. Then, we propose our scheduling methods in the case of complete interference graph. At the end, we extend our method to the case of non-complete interference graph.

A. Video Coding Scheme

In order to provide reliable transmission without using feedback messages, we use RLNC. We first partition each video into equal size packet segments. We then perform RLNC to code the packets of each segment. Figure 3(a) shows the packets of a video, which are partitioned to a set of segments. The encoded packets of the original video are shown in Figure 3(b). We did not show the coefficients in the

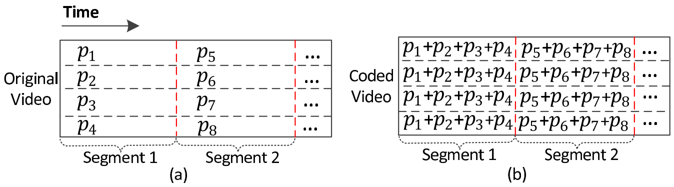


Fig. 3. NC scheme. The coefficients are not shown for simplicity; (a) Segmentation of an original video; (b) RLNC inside each segment.

figure for simplicity. For instance, $p_1 + p_2 + p_3 + p_4$ means $\alpha_1 p_1 + \alpha_2 p_2 + \alpha_3 p_3 + \alpha_4 p_4$, where α_1 to α_4 are random coefficients; thus, in Figure 3, the encoded packets of each segment are different. This coding process is performed on the video servers, and they transmit the coded packets to the APs using the wired reliable links.

RLNC helps us to provide reliable transmissions without the need for feedback messages. When we code k packets together, any k linearly independent coded packets are sufficient to decode the coded packets. As a result, the coded packets have the same importance and contribute the same amount of information to the clients. Therefore, the AP nodes do not need to retransmit the lost packets by a client node and they can transmit a different coded packet instead. In contrast, if we do not use NC, the AP nodes need to know the lost packets and retransmit them. Using feedback messages in the multicast application is a huge challenge, which can be solved with the help of NC.

When we use RLNC to code k packets, the coded packet cannot be decoded by a user until it receives k linearly independent coded packets. As a result, when the user watches the video, a video lag problem might occur. The users can resolve this problem by buffering the received coded packets of each segment and delaying the playback of the video for a specific amount of time. In this way, the decoding delay does not result in a video lag problem. Computing the buffering time is beyond the scope of our current work.

B. APs Scheduling Algorithm for Complete Interference Graph

Our goal is to have a fair scheduling that maximizes the expected number of packets that the users receive. In other words, we want to maximize the minimum number of received packets by each AP node. As we will see in the next section, in the case of disjoint transmissions by the APs, the solution of this problem can be found in a polynomial time by solving a linear programming optimization. In contrast, in the case of overlapped scheduling of m WiFi AP nodes, the potential number of concurrent transmissions is $2^m - 1$. As a result, the time complexity of this scheduling is high. For this reason, we use a two-phase scheduling algorithm for the AP nodes. In the first phase, we find the optimal scheduling in the case of disjoint transmissions by the APs, which is done using a linear programming optimization. Then, in the second phase, we use the result of the first phase as an initial solution, and try to enhance the total utility by allowing some concurrent transmission of the interfering AP nodes. In the second phase, we use linear programming to find the concurrent transmissions that can increase the total utility.

1) Phase 1: Disjoint Transmissions Scheduling:

In this phase, we find a basic solution for the problem, and we do not allow concurrent transmission of the APs. We can find the optimal scheduling in the case of disjoint transmissions by solving the following linear programming:

$$\max \; y \tag{1}$$

$$s.t \quad \sum_{j \in B} x_j \leq 1 \tag{2}$$

$$r_i = \sum_{j \in C(i)} b \cdot x_j (1 - \epsilon_{ji}), \quad \forall i \in U \tag{3}$$

$$y \leq r_i, \quad \forall i \in U \tag{4}$$

We note the fraction of time that AP j is scheduled to transmit as x_j. The main constraint of the scheduling is that the AP nodes should not transmit at the same time. As a result, the total fraction of time that is assigned to the APs should not exceed 1, which is represented as the set of Constraints (2). The transmission bandwidth of the AP nodes is represented as b. As a result, assuming that user i is in the transmission range of AP j, the expected number of packets that user i receives from AP j is equal to $b \cdot x_j (1 - \epsilon_{ji})$. Since we do not allow concurrent transmissions, the expected total number of packets that a user receives is equal to the summation of the expected packets that it receives from the APs that cover this user. The set of Constraints (3) calculate the expected total received packets by each user.

The set of Constraints (4) are fairness constraints. We represent the expected total number of packets that are received by user i as r_i. In order to have a fair scheduling, the expected number of packets that are received by the users should be close to each other. As a result, instead of maximizing the total number of received packets by all of the user nodes, we try to maximize the minimum expected number of packets that are received by any user. For this purpose, we use auxiliary variable y, and set it to less than or equal to the received rate by the users. Also, we set the objective to maximizing y. We refer to this optimization as 'fair scheduling phase 1' (FS1).

If we do not consider the fairness as a constraint, the problem can be formulated as the following linear programming optimization:

$$\max \sum_{i \in U} r_i \tag{5}$$

$$s.t \quad \sum_{j \in B} x_j \leq 1 \tag{6}$$

$$r_i = \sum_{j \in C(i)} b \cdot x_j (1 - \epsilon_{ji}), \quad \forall i \in U \tag{7}$$

Here, we removed the set of Constraints (4), and changed the objective function of the optimization to maximizing the total expected number of received packets by all of the destination nodes. We refer to this optimization as 'unfair scheduling phase 1' (US1).

2) Phase 2: Concurrent Transmissions Scheduling: After Phase 1, we have an optimal disjoint transmission scheduling for the AP nodes. In Phase 2, we use the output of the first phase as the input of the second phase optimization and allow concurrent transmission of the AP nodes such that it can increase the fairness of the system. In order to reduce the time complexity of the algorithm, we only permit two interfering APs to transmit at the same time. The idea is that we increase the fraction of time x_j that node AP node j is scheduled to transmit by allowing node j to use an extra x_{kj} portion of the time to transmit, where x_{kj} is the fraction of time that is borrowed from AP node k. In this way, the users that are in the common coverage area of AP nodes j and k will not be able to receive any packet due to the interference. As a result, assuming that user i is such a user, its expected number of received packets will be reduced by $b \cdot x_{kj}(1-\epsilon_{ji})$. Moreover, if node i is only covered by AP j, it will receive $b \cdot x_{kj}(1-\epsilon_{ki})$ extra packets. In this case, the optimization becomes the following linear programming optimization:

$$\max \ y \tag{8}$$

$$s.t \quad \sum_{j \in B} z_{kj} \leq x_k \quad \forall k \in B \tag{9}$$

$$s_i \leq r_i + \sum_{k \notin C(i)} \sum_{\substack{j \in B \\ i \in C(j)}} b \cdot z_{kj}(1-\epsilon_{ji}) - \sum_{j \in C(i)} \sum_{\substack{k \in C(i) \\ j \neq k}} b \cdot z_{kj}(1-\epsilon_{ki}), \quad \forall i \in U \tag{10}$$

$$y \leq s_i, \quad \forall i \in U \tag{11}$$

Here, Constraint (11) is similar to Constraint (4). Moreover, the objective function is the same as in the first phase. The total time that AP k can lend to the other AP nodes cannot exceed the transmission time x_k that is assigned to it in the first phase, which is expressed as the set of Constraints (9). The set of Constraints (10) calculates the expected number of received packets by the users in the case of overlapped transmissions. In (10), r_i is the calculated expected number of received packets in phase 1. The first summation calculates the additional number of packets that are received by user i in the case of overlapped transmissions. Moreover, the second summation is equal to the number of packets that are missed due to the interference among the AP nodes that cover user i. We refer to this optimization as the 'fair scheduling phase 2' (FS2). The summary of our fair scheduling method is shown in Algorithm 1.

In the case of not considering the fairness, the optimization becomes:

$$\max \sum_{i \in U} s_i \tag{12}$$

$$s.t \quad \sum_{k \in B} z_{jk} \leq x_j \quad \forall j \in B \tag{13}$$

$$s_i \leq r_i + \sum_{k \notin C(i)} \sum_{\substack{j \in B \\ i \in C(j)}} b \cdot z_{jk}(1-\epsilon_{ki}) - \sum_{j \in C(i)} \sum_{\substack{k \in C(i) \\ j \neq k}} b \cdot z_{jk}(1-\epsilon_{ji}), \quad \forall i \in U \tag{14}$$

Which is called 'unfair scheduling phase 2' (US2).

Algorithm 1 Fair Scheduling Algorithm (Complete interference graph)

```
// Phase 1
Maximize y subject to Constraints (2), (3), and (4)
for each i ∈ U do
    r_i = Σ_{j∈C(i)} b · x_j (1 − ε_ji)
// Phase 2
Maximize y subject to Constraints (9), (10), and (11)
```

C. APs Scheduling Algorithm for Non-Complete Interference Graph

In the interference graph, the nodes that are not connected to each other do not interfere. In graph theory, these nodes are called independent sets. In order to schedule as many AP nodes as possible at the same time, we need to find the maximum independent sets. Maximum independent set is a set of nodes in which none are connected to each other, and the size of this independent set is more than any other independent set. However, it is well-known that finding a maximum independent set is an NP-complete problem. To overcome this problem, we can find maximal independent sets instead of the maximum independent set. A maximum independent set is defined as an independent set in that if we add any additional node to this set, the set will no longer be an independent set.

In the case of overlapped scheduling of $|S|$ independent sets, the potential number of concurrent transmissions is $2^{|S|} - 1$. As a result, similar to the case of full interference graph, the time complexity of this scheduling is high. Therefore, we use a three phase low complexity algorithm to schedule the APs for this interference model. We find a set of independent sets in the first phase. In the second phase, we find the optimal scheduling in the case of disjoint independent sets transmissions. This can be done using linear programming. Then, in the third phase, we use linear programming to find the concurrent transmissions of the independent sets, such that it increases the fairness.

1) Phase 1: Finding the Independent Sets: In the first phase, we first construct the interference graph of the AP nodes. For this purpose, we show each AP with a node in the interference graph, and we connect each pair of nodes if their correspondent AP nodes interfere with each other. In order to prevent interference among the APs, we should only allow a set of nodes to transmit at the same time if they do not interfere with each other. A common approach for the scheduling methods is to use maximal independent sets

Algorithm 2 Maximal Independent Sets

Input: B and $N(j)$ $\forall j \in B$
$S = \{\}$, $A = B$
Unmark the nodes in B
while There is an unmarked node in B **do**
  while A is not empty **do**
    $I = \{\}$
    Find a node j in A with the minimum degree
    Mark node j in B
    $I = I \cap j$; $A = A/N(j)$
  $S = S \cup I$
  Put the unmarked nodes of B in A

instead of maximum independent sets, which can be found in a polynomial time.

The details of the algorithm that finds the maximal independent sets are shown in Algorithm 2. The algorithm puts the unmarked AP nodes of B in set A. In order to find an independent set, at each iteration the algorithm, find a node j in A with the minimum degree and add it to the independent set. It then removes the neighbors of node j from the set A and mark node j in B as marked. This process is repeated until set A becomes empty. After finding an independent set, all of the nodes in the independent set are marked in set B. We then add the unmarked nodes in B to set A, and run the algorithm again to find another independent set from the set of unmarked nodes. This process is repeated until all of the nodes in B become marked.

2) Phase 2: Disjoint Transmissions Scheduling: Similar to the previous section, the optimal scheduling in the case of disjoint transmissions can be found by solving the following linear programming:

$$\max \ y \tag{15}$$

$$s.t \quad \sum_{h:I_h \in S} x_h \leq 1 \tag{16}$$

$$r_i = \sum_{h:I_h \in S} \sum_{j \in C(i), j \in I_h} b \cdot x_h(1-\epsilon_{ji}), \quad \forall i \in U \tag{17}$$

$$y \leq r_i, \quad \forall i \in U \tag{18}$$

There is no interference among the AP nodes in the same independent sets. Therefore, in order to impede interference, it is enough to make sure that the different independent sets are not scheduled at the same time. It means that the total fraction of time that different independent sets are scheduled should be less than or equal to one, which is represented as the Constraint (16). Here, x_h is the fraction of time that independent set I_h is scheduled and the set of all independent sets it noted as S. In the case that AP j belongs to independent set I_h and it covers user i, the expected number of packets that user i receives from AP j is equal to $b \cdot x_h(1-\epsilon_{ji})$. In order to calculate the expected total number of packets that are received by user i we take the summation of the received packets over different independent sets, which is expressed as the set of Constraints (17). Moreover, similar to the set of Constraints (4), we have the fairness Constraints (18). We call this optimization 'fair scheduling with non-complete interference phase 1' (FSN1).

In the case of not considering fairness, the optimization becomes as follows:

$$\max \sum_{i \in U} r_i \tag{19}$$

$$s.t \quad \sum_{h:I_h \in S} x_h \leq 1 \tag{20}$$

$$r_i = \sum_{h:I_h \in S} \sum_{j \in C(i), j \in I_h} b \cdot x_h(1-\epsilon_{ji}), \quad \forall i \in U \tag{21}$$

which is referred to as 'unfair scheduling with non-complete interference phase 1' (USN1).

3) Phase 3: Concurrent Transmissions Scheduling: After phase 2, we have an optimal disjoint transmission scheduling for the APs. In Phase 3, we use the output of the previous phase and modify it by allowing concurrent transmission of the APs such that it can increase the fairness of the system. Similar to the case of complete interference graph, in order to reduce the time complexity of the algorithm, we only permit two interfering APs to transmit at the same time. In this case, the optimization becomes as the following linear programming optimization:

$$\max \ y \tag{22}$$

$$s.t \quad \sum_{h:I_h \in S} z_{gh} \leq x_g \quad \forall g : I_g \in S \tag{23}$$

$$\begin{aligned} s_i \leq r_i &+ \sum_{g \notin C(i)} \sum_{i \in C(h)} b \cdot z_{gh}(1-\epsilon_{hi}) \\ &- \sum_{h \in C(i)} \sum_{\substack{g \in C(i) \\ g \neq h}} b \cdot z_{gh}(1-\epsilon_{gi}), \quad \forall i \in U \end{aligned} \tag{24}$$

$$y \leq s_i, \quad \forall i \in U \tag{25}$$

The objective function is the same as the second phase. The total time that independent set g can lend to the other independent sets cannot exceed the transmission time x_g that is assigned to it in the first phase, which is expressed as the set of Constraints (23). We use the set of Constraints (24) to calculate the expected number of received packets by the users in the case of overlapped transmissions. In (24), $C(i)$ is the set of independent sets that cover user i, and r_i is the expected number of received packets by user i, which is calculated in phase 1. Similar to the case of complete interference graph, the first and second summations in (24) calculate the additional number of packets that are received by user i and the number of packets that are missed due to the interference among the APs, respectively. We call this optimization 'fair scheduling with non-complete interference phase 2' (FSN2). If we do not consider the fairness the optimization becomes:

$$\max \sum_{i \in U} r_i \tag{26}$$

Algorithm 3 Scheduling Algorithm

// Phase 1
Construct the conflict graph of APs in B
Find the set of independent sets S using Algorithm 2
// Phase 2
Maximize (16) subject to Constraints (17), and (18)
for each $i \in U$ **do**
    $r_i = \sum_{j \in C(i)} b \cdot x_j (1 - \epsilon_{ji})$
// Phase 3
Maximize (23) subject to Constraints (24), (25)

$$s.t \quad \sum_{h: I_h \in S} z_{gh} \leq x_g \quad \forall g : I_g \in S \tag{27}$$

$$\begin{aligned} s_i \leq r_i &+ \sum_{g \notin C(i)} \sum_{i \in C(h)} b \cdot z_{gh} (1 - \epsilon_{hi}) \\ &- \sum_{h \in C(i)} \sum_{\substack{g \in C(i) \\ g \neq h}} b \cdot z_{gh} (1 - \epsilon_{gi}), \quad \forall i \in U \end{aligned} \tag{28}$$

We refer to this optimization as 'unfair scheduling with non-complete interference phase 2' (USN2).

V. Evaluation

In this section, we evaluate the proposed methods through extensive simulations. We measure the total number of received packets, number of decodable packets, and fairness of the methods.

A. Simulation Setting

In order to evaluate the methods, we implemented a simulator in the MATLAB environment. We evaluate all of the methods on 1,000 topologies with random link delivery rates. The presented plots of this paper are based on the average outputs of the simulation runs. We assume that the delivery rate of the wireless links are independent. We randomly distribute the nodes in a 20×20 M square area, and compute the delivery rate of the links between the APs and the users based on the Euclidean distance between them. For any two nodes separated by distance L, we use the Rayleigh fading model [21] to calculate the successful delivery probability:

$$P = \int_{T^*}^{\infty} \frac{2x}{\sigma^2} e^{-\frac{x^2}{\sigma^2}} dx \tag{29}$$

where we set:

$$\sigma^2 \triangleq \frac{1}{(4\pi)^2 L^{\alpha}} \tag{30}$$

the path loss order $\alpha = 2.5$, and the decodable SNR threshold $T^* = 0.006$.

B. Simulation Results

In the following subsections, we first evaluate our proposed method for the complete interference graph. We then report our evaluations for the case of non-complete interference graph.

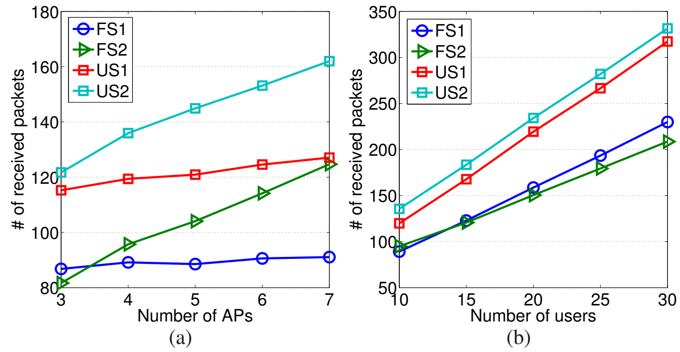


Fig. 4. Total number of received packets by the users. (a) Number of users=10. (b) Number of APs=4.

1) Complete Interference Graph: In this section we compare the 'fair scheduling phase 1' (FS1), 'fair scheduling phase 2' (FS2), 'unfair scheduling phase 1' (US1), and 'unfair scheduling phase 2' (US2) together. In the first experiment, we measure the effect of number of AP nodes on the total number of received packets by the user nodes. In Figure 4(a), we set the number of user nodes to 10, and vary the number of AP nodes from 3 to 7. The total number of received packets increases as we increase the number of AP nodes. The reason is that, as we increase the number of APs, each user will be covered with more APs. As a result, there is a higher chance that each user has at least one wireless channel with a good condition. Consequently, the total number of delivered packets to the users increases.

Figure 4(a) shows that the unfair method results to the highest total number of received packets. The reason is that, in the optimization, we set the objective function to maximizing the total number of received packets. In phase 2 of the proposed fair scheduling, we allow some of the AP nodes to transmit concurrently in order to increase the number of received packets by the users. Figure 4(a) confirms the effectiveness of the concurrent transmission of the AP nodes. For 3 APs, FS2 results in fewer received packets than the FS1. The reason is that, the probability that many users are covered by a single AP is high. As a result, there is no opportunity for phase 2 to increase the received packets. On the other hand, the objective of FS2 is to increase the fairness.

In the next experiment, we measure the effect of number of users on the total number of received packets. In Figure 4(b), we set the number of AP nodes to 4, and vary the number of users from 10 to 30. As expected, the total number of received packets increases as we increase the number of users. Since the application in our model is multicasting in a wireless environment, each user that is in the coverage area of the AP nodes can overhear and receive the transmitted packets. Moreover, in our model, the packet reception by the users are independent of each other. As a result, the relation of the total number of received packets and users is linear. Furthermore, Figure 4(b) depicts that the unfair method results in a greater total number of received packets. The figure

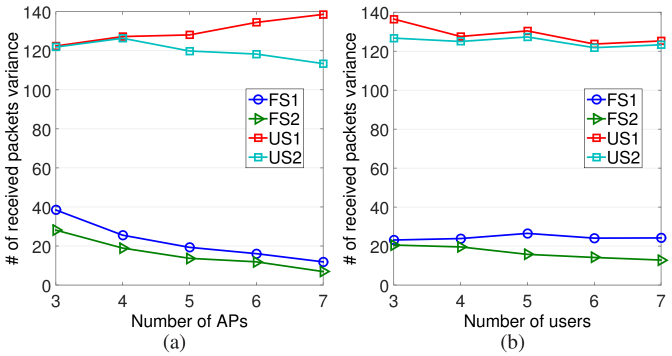


Fig. 5. Variance of the number of received packets by the different users. (a) Number of users=10. (b) Number of APs=4.

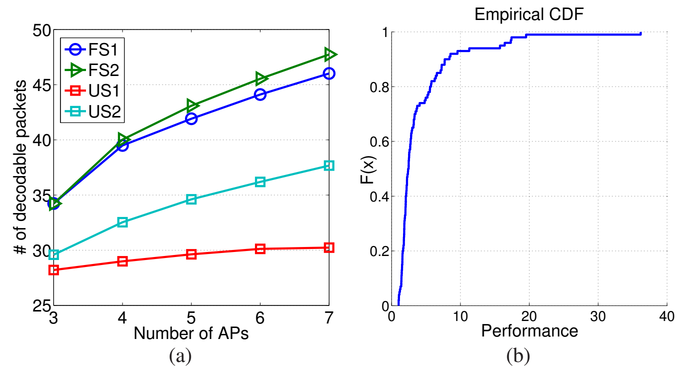


Fig. 6. (a) Number of decodable packets by the users, number of users=10. (b) Empirical CDF of the performance of the proposed 2-phase algorithm over disjoint transmissions.

confirms the increase in the total number of packets in the case on allowing concurrent transmission of the interfering APs. As the number of users increases, the total number of received packets in the FS2 method becomes less than the FS1 method. The objective in both of the methods is fairness. However, concurrent transmissions in FS2 provides more fairness, and in the case of many users, providing fairness sacrifices the total number of received packets.

The first two experiments show that the unfair method results in a greater number of received packets. Also, allowing concurrent transmissions in the case of fair scheduling might increase or decrease the total number of received packets. In order to check the fairness of the proposed methods, we calculate the variance of the number of received packets by the different users and depict the result in Figure 5(a). The figure shows that the unfair method has a higher variance compared to the fair method. Moreover, allowing the AP nodes to transmit at the same time reduces the variance of number of received packets.

It can be inferred from Figure 5(a) that the variance of the total number of received packets in the US1 increase as we increase the number of APs. This is because US1 does not consider the fairness. In contrast, the concurrent transmissions in US2 reduces the variance compared to the US1 method, and the variance almost does not change as we increase the AP nodes. Having more AP nodes provides more opportunities for the FS1 and FS2 methods to provide fairness; thus their variance decreases as we increase the number of APs.

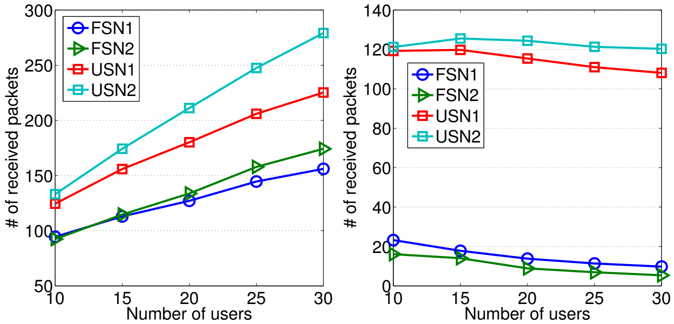
We show the effect of number of users on the variance of received packets in Figure 5(b). As expected, US1 and US2 methods have the highest variance. Moreover, allowing concurrent transmissions by the AP nodes reduces the variance in both of the unfair and fair scheduling methods. The variance of all of the methods is almost fixed, and does not change as we increase the number of users.

Figure 6(a) shows the number of decodable packets by the users. We set the number of packets that are coded together to 5. A user is able to decode the coded packets if it receives at least 5 linearly independent coded packets. Otherwise, the received coded packets are useless and undecodable. As expected, the FS2 method has the greatest number of decodable packets, as it considers the fairness. The objective in the US1 and US2 methods is maximizing the received packets. Therefore, some of the users might receive much more packets than 5. These nodes can decode the coded packets; however, the extra coded packets are useless. On the other hand, the users that receive less than 5 coded packets are not able to decode any of them.

In Figure 6(b), we show the empirical CDF of the performance of the concurrent transmission over the disjoint transmissions. We define the performance as the ratio variance of the number of received packets in the case of disjoint transmissions by the variance in the case of allowing concurrent transmissions. In other words, we divide the variance of the number of received packets in the FS1 method by that of the FS2 method, and draw the empirical CDF. The figure shows that, in 50% of the runs, the variance of FS1 is 20% more than that of FS2 method. Moreover, in 10% of the cases, the variance of FS1 is at least 8 times that of FS2.

2) Non-complete Interference Graph: We repeat the evaluations for the case on non-complete interference graph. We compare the 'fair scheduling with non-complete interference phase 1' (FSN1), 'fair scheduling with non-complete interference phase 2' (FSN2), 'unfair scheduling with non-complete interference phase 1' (USN1), and 'unfair scheduling with non-complete interference phase 2' (USN2) together. In order to reduce the density of the nodes and make the interference graph of the AP nodes non-complete, we increase the size of the filed in which the nodes are randomly scattered to 30×30 M. The other parameters are the same as those in the previous section.

We first measure the effect that the number of user nodes has on the total number of received packets by the users. We set the number of AP nodes to 10, and change the number of users from 10 to 30. In Figure 7(a), the total number of received packets increases as we increase the number of users. Similar to Figure 4(b), the relation of the received packets and the number of users is linear. For 10 users, the number of received packets in the case of FSN1 and FSN2 are almost the



(a) (b)

Fig. 7. Non-complete interference graph. (a) Total number of received packets, number APs=10. (b) Variance of the number of received packets, number of APs=10.

same. The reason is that, for a small number of users, there is a high chance that each user is covered only by a single AP. The USN1 an USN2 deliver a higher number of packets, which is due to their objective function.

In the last experiment, we vary the number of users from 10 to 5, and measure the variance of the number of received packets by the users. Similar to Figure 5 (b), the number of users does not have a huge affect of the fairness of the USN1 and USN2 methods. However, a higher number of users slightly reduces the variance of the FSN1 and FSN2 methods.

C. Simulation Summary

- The overlapped transmission of the AP nodes not only increases the total number of received packets in the unfair methods, but also reduces the variance of number of the received packets, compared to that of the non-overlapped transmissions.
- In the case of fair scheduling, the overlapped transmissions increase the fairness by decreasing the variance of the number of received packets. However, depending on the topology, it might increase or decrease the total number of received packets

VI. Conclusion

One of the main applications of wireless devices such as smartphones and tablets is watching videos over the Internet. In this paper, we propose using multiple APs and NC to multicast video streams to the client nodes. With the help of multiple APs, the client nodes will experience spatial and time diversity and will receive more packets. Moreover, with the help of NC, all of the transmitted packets have the same importance. As a result, reliable transmissions can be achieved without a need for feedback messages. In order to efficiently use the shared wireless network, we propose a resource allocation algorithm.

In contrast with the previous resource sharing methods which do not allow the interfering APs to transmit concurrently, we allow a systematic concurrent transmission. We show that this systematic concurrent transmission of the interfering APs can improve the system performance in terms of fairness and number of received packets. We propose a two-phase resource allocation algorithm to enhance the system utility. We depict the effectiveness of our proposed methods through extensive simulations.

Acknowledgment

This work is supported in part by NSF grants CNS 149860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, ECCS 1231461, ECCS 1128209, and CNS 1138963.

References

[1] A. Finamore, M. Mellia, M. Munafò, R. Torres, and S. Rao, “Youtube everywhere: impact of device and infrastructure synergies on user experience,” in *ACM IMC*, 2011, pp. 345–360.

[2] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, “Internet inter-domain traffic,” in *ACM SIGCOMM*, 2010, pp. 75–86.

[3] H. Djandji, “An efficient hybrid arq protocol for point-to-multipoint communication and its throughput performance,” *IEEE Transactions on Vehicular Technology*, vol. 48, no. 5, pp. 1688–1698, 1999.

[4] B. Zhao and M. Valenti, “Practical relay networks: a generalization of hybrid-ARQ,” *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 7–18, 2005.

[5] ——, “The throughput of hybrid-ARQ protocols for the gaussian collision channel,” *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 1971–1988, 2001.

[6] L. Rizzo and L. Vicisano, “RMDP: an FEC-based reliable multicast protocol for wireless environments,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 2, no. 2, pp. 23–31, 1998.

[7] R. Ahlswede, N. Cai, S. Li, and R. Yeung, “Network information flow,” *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.

[8] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, “A random linear network coding approach to multicast,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.

[9] A. Shokrollahi, “Raptor codes,” *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[10] M. Luby, “LT codes,” in *The 43rd Annual IEEE Symposium on Foundations of Computer Science*, 2002, pp. 271–280.

[11] M. Choi, W. Sun, J. Koo, S. Choi, and K. G. Shin, “Reliable video multicast over wi-fi networks with coordinated multiple aps,” in *INFOCOM*, 2014, pp. 424–432.

[12] L. Rizzo, “Effective erasure codes for reliable computer communication protocols,” *ACM SIGCOMM Computer Communication Review*, vol. 27, no. 2, pp. 24–36, 1997.

[13] P. Cataldi, M. Shatarski, M. Grangetto, and E. Magli, “Lt codes,” in *IIH-MSP’06*, 2006, pp. 263–266.

[14] T. Fang and L. Chau, “Op-based channel rate allocation using genetic algorithm for scalable video streaming over error-prone networks,” *Image Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 71 323–1330, 2006.

[15] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the scalable video coding extension of the h. 264/avc standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[16] P. Ostovari, A. Khreishah, and J. Wu, “Multi-layer video streaming with helper nodes using network coding,” in *IEEE MASS*, 2013, pp. 524–532.

[17] R. Koetter and M. Medard, “An algebraic approach to network coding,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782– 795, Oct 2003.

[18] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, “Xors in the air: practical wireless network coding,” in *ACM SIGCOMM*, 2006, pp. 243–254.

[19] P. Ostovari, J. Wu, and A. Khreishah, *Network Coding Techniques for Wireless and Sensor Networks*. Springer, 2014.

[20] S. Li, R. Yeung, and N. Cai, “Linear network coding,” *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.

[21] C. Wang, A. Khreishah, and N. Shroff, “Cross-layer optimizations for intersession network coding on practical 2-hop relay networks,” in *Asilomar*, 2009, pp. 771–775.