Adaptive Network Coding for Wireless Access Networks

Tuan Tran
School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, Oregon 97331
Email: trantu@eecs.orst.edu

Thinh Nguyen
School of Electrical Engineering and Computer Science
Oregon State University, Corvallis, Oregon 97331
Email: thinhq@eecs.orst.edu

***Abstract*—We propose a framework for optimizing the quality of service of multiple simultaneous flows in wireless access networks via network coding. Specifically, we consider the typical scenario in which multiple flows originate from multiple sources in the Internet and terminate at multiple users in a wireless network. In the current infrastructure, the wireless base station is responsible for relaying the packets from the Internet to the wireless users without any modification to the packet content. On the other hand, in the proposed approach, the wireless base station is allowed to perform *network coding* by appropriate linear mixing and channel coding of packets from different incoming flows before broadcasting a single flow of *mixed or coded* packets to all wireless users. Each user then uses an appropriate decoding method to recover its own packets from the set of coded packets that it receives. We show that in principle, for the given channel conditions and QoS requirements, appropriate mixing and channel coding of packets across different flows can lead to substantial quality improvement for both real-time and non-real time flows. On the other hand, blind mixing can be detrimental. We formulate this mixing problem as a combinatorial optimization problem, and propose a heuristic algorithm based on simulated-annealing method to approximate the optimal solution. Simulation results verify the performance improvement resulting from the proposed approach over the non-network coding and the state-of-the-art network coding approaches.**

I. INTRODUCTION

In today communication networks such as the Internet and wireless ad hoc networks, data delivery is performed via the store-and-forward routing, using which, the intermediate routers do not alter the contents of packets as they traverse hop-by-hop from a source to a destination. In contrast, *Network coding* (NC), a new routing approach pioneered by Ahlswede *et al.* [1], allows intermediate routers to generate data on its output link by mixing data from multiple of its input links. In this way, it is theoretically possible to achieve the throughput capacity of a multicast/broadcast session, while this is not possible with the store-and-forward routing scheme. Recently, NC technique has also been applied successfully to increase throughput in wireless networks [2]–[4]. Having said that, supporting sophisticated functionalities at intermediate routers goes against the end-to-end design principle [5] which argues for simple routers to increase performance and scalability. This principle has been cited in part for the huge success of the Internet.

In this paper we propose using NC at a wireless base station to improve Quality of Service for all-type applications. Specifically, we consider the typical scenario in which, multiple flows originate from the Internet, traverse a wireless base station, and terminate at multiple users in a wireless access network. As such, the wireless base station is allowed to perform *network coding* by appropriate linear mixing and channel coding of packets from different incoming flows before broadcasting a single flow of *mixed or coded* packets to all wireless users. Each user then uses an appropriate decoding method to recover its own packets from the set of coded packets that it receives.

Studying NC in this setup is not entirely new. NC has been suggested in the context of Wi-Fi and WiMax networks. Indeed, Nguyen *et al.* [6] proposed some XOR based network coding schemes together with a scheduler at a Wi-Fi Access Point (AP) to improve throughput where feedback is readily available. Nguyen *et al.* further extended this work to reduce delay for video transmission in [7]. Eryilmaz *et al.* [8] also proposed to use NC technique that employs a large finite field, rather than XOR-based NC techniques, to improve throughput at the BS. In addition, Tran *et al.* [9] also showed that employing NC jointly with channel coding across the multiple wireless unicast sessions can significantly increase the overall wireless throughput.

What new in this paper, however, is a framework for applying NC coding at the BS, that optimizes for the quality of service of both real-time and non-real time traffic. We show that in principle, for the given channel conditions and QoS requirements, appropriate mixing and channel coding of packets across different flows can lead to substantial quality improvement for both real-time and non-real time applications, e.g. video streaming and FTP. On the other hand, blind mixing of flows may actually degrade their qualities. Quantifying the performance of flow mixing, and finding the optimal mixing with respect to some metric, are the main objectives of this paper. We formulate this mixing problem as a combinatorial optimization problem, and propose a heuristic algorithm based on simulated-annealing method, to approximate the optimal solution. Simulations confirm the performance improvement resulting from the proposed *controlled mixing* approach over the non-network coding and the state-of-the-art network coding approaches. We first introduce the background and related work.

II. BACKGROUND AND RELATED WORK

The notion of NC, i.e., mixing of data at intermediate nodes to increase the overall multicast capacity of a network, was first proposed in the seminal paper by Ahlswede *et al.* [1]. Ahlswede provided an existence proof of some network codes (method of mixing data at intermediate nodes) that achieve multicast capacity. In recent years, NC has been also applied successfully to wireless ad hoc networks [2],[10]. A classical example first proposed by Wu *et al.* [10] for efficient information exchange in a wireless ad hoc network is shown in Fig. 1. Two nodes U_1 and U_2 want to exchange their packets through U_3. Packet a sent by U_1 to U_2 is relayed through U_3, and packet b sent by U_2 to U_1 is relayed through U_3. As a result, U_3 has both a and b. In an existing wireless ad hoc network, U_3 has to perform two transmissions, one transmission for sending a to U_2, and another one for sending b to U_1. Now using NC, upon

978-1-4244-7116-4/10/$26.00 ©2010 IEEE

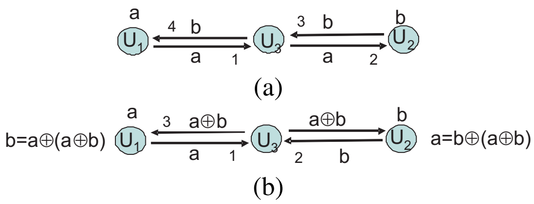
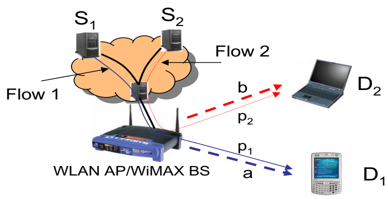


Fig. 1. (a) Information exchange using (a) the store-and-forward scheme using 4 transmissions and (b) the NC scheme using only 3 transmissions.



(a)

Time slot	1		2		3		4
UNI | RNC	a | $c_1 = \alpha_1 a + \beta_1 b$		b | $c_2 = \alpha_2 a + \beta_2 b$		a | $c_3 = \alpha_3 a + \beta_3 b$		b
D_1	X	X	-	O	O	O	-
D_2	-	O	X	X	-	O	O

(b)

Fig. 2. (a) A transmission scenario of unicast flows in WLAN/WiMAX network. (b) Lost packet patterns at the receivers; in each time slot, the first and second packets correspond to transmitted packets of the traditional and NC techniques. "X", "O" and "-" denote lost, successful and "*don't care*" packets, respectively.

receiving a and b, U_3 broadcasts $(a \oplus b)$ to U_1 and U_2. Since U_1 has a, it can recover b as $b = a \oplus (a \oplus b)$. Similarly, U_2 can recover a as $a = b \oplus (a \oplus b)$. Following this work, Katti *et al.* [2] proposed an opportunistic XOR-based scheme for wireless mesh networks and showed a substantial bandwidth improvement over the current store-and-forward approach.

In similar spirit, NC has been proposed to improve throughput in wireless access networks by Nguyen *et al.* [6] and Eryilmaz *et al.* [8]. Fig. 2(a) shows a simple example of how performing NC at a wireless base station can increase the throughput. Assuming that, there are two concurrent flows $f_1 : S_1 \rightarrow D_1$ and $f_2 : S_2 \rightarrow D_2$, both share the same wireless channel from the BS to the receivers.

With the existing infrastructure, upon receiving packets a and b, the BS uses the first and second time slots to deliver a and b to D_1 and D_2, respectively. The receivers use ACK or NAK message, respectively, to signal the BS whether they receive a packet correctly or not. Assuming that, after the first two time slots, the BS has a packet loss pattern as shown in Fig. 2.(b), that is, packet a is lost at D_1 while packet b is lost at D_2. If we assume that, it takes one time slot to retransmit a lost packet, then the BS needs two more transmissions. In all, 4 time slots are needed to deliver packets a and b to D_1 and D_2, respectively.

Now consider the randomized NC (RNC) technique introduced by Koetter *et al.* in [11]. Upon receiving two packets a and b, the BS generates *coded* packets by linearly combining a and b with random coefficients. For example, a coded packet c_i is generated as

$$c_i = \alpha_i a + \beta_i b, \tag{1}$$

where α_i and β_i are coefficients drawn at random from a large finite field.

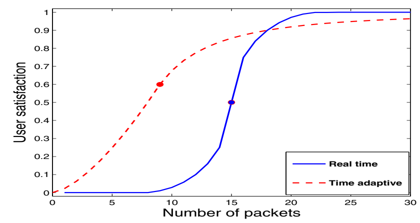


Fig. 3. Satisfaction functions of different applications versus the number of packets.

The two coded packets are then broadcasted to both receivers. Assuming that, after the first two transmissions, the packet loss pattern at the BS is the same as before (the first packet is lost at D_1, successful at D_2 while the second packet is successful at D_1 and lost at D_2). In the third time slot, the BS generates and broadcasts another coded packet. Suppose at this time, both D_1 and D_2 receive the coded packet correctly. Then, each receiver has received successfully a total of two coded packets, and will be able to recover their own packets by solving a system of linear equations. We assume that the coefficients are included in the packet header to enable the receiver to set up the equation. With sufficiently large packet size, this overhead is negligible [12]. Overall, RNC approach needs only 3 transmissions to deliver two packets to the intended receivers.

However, it is not always beneficial to mix packets from different flows (more results will be shown in the next Sections). In fact, blind mixing can be detrimental. This paper advocates an informed packet mixing approach to maximize the QoS requirements of different flows.

III. System Model and Transmission Techniques

To quantify and compare the performance of our informed packet mixing approach, we consider the following transmission techniques.

A. Transmission Techniques

We consider a *last mile* wireless scenario consisting of one BS and M receivers. In particular, assume that there are M flows from sources S_i to destinations D_i for $i = 1, 2, \ldots, M$, and these flows share the wireless channel from the BS to the receivers. We associate each destination D_i with a demand S_i, the number of packets that the receiver D_i wants to receive per time unit. The ith flow has a packet-level FEC coding scheme denoted as (n_i, k_i) where a code (n_i, k_i) means that we use $n_i \geq k_i$ time slots to transmit k_i packets.

Without loss of generality, let us assume that BS has enough memory to store data for one transmission period $N = \sum_{i=1}^{M} n_i$. In this paper we consider three techniques: unicast (UNI), Randomized Network Coding (RNC), and the proposed Adaptive Randomized Network Coding (ARNC). Before proceeding to the details of analysis, let us describe the protocol used in each technique.

$\triangleright$ **Unicast (UNI):** This is a common traditional approach. In each period, the BS uses a pre-specified maximum n_i time slots to transmit k_i data packets for receiver D_i. This pre-specification of FEC level can be viewed as a method to accommodate varying QoS levels as specified by different flows. If there are packet losses, the BS uses $n_i - k_i$ redundant time slots to retransmit the lost packets. The BS switches to transmit data packets for the next receiver when it receives an ACK message from the current receiving user or all the time slots intended for the current receiver have been used.

▷ **Randomized Network Coding (RNC):** Transmissions in RNC are classified into two phases: the basis and the augmentation. In the basic phase, all $\sum_{i=1}^{M} k_i$ original packets will be transmitted first. The receivers cache all the packets, including ones that are not intended for them. Next, the BS mix all the packets from all users, to generate coded packets and broadcasts them to all receivers. Again, the n_i and k_i are pre-specified as in the UNI technique. The receivers which have lost some of their original packets now can receive the coded packets to decode their own data based on the method described in Section II. Note that, because packets are not mixed in the basis phase, a receiver is able to obtain a fraction of its packets, even though it cannot recover all the mixed packets in the augmentation phase.

▷ **Proposed Adaptive Randomized Network Coding (ARNC):** The BS now has the ability to choose what flows to be mixed together based on their channel conditions, types of services and priorities. Note that all these parameters (channel conditions, types of services, and priorities) of different flows have been considered in UNI and RNC techniques through their assigned FEC codes. Assuming that the M incoming flows are partitioned into G groups. Then, for each group, the BS uses RNC to transmit data to the receivers inside that group only. Assuming n_i and k_i are fixed as in the previous two techniques, the objective of the BS is to determine the optimal partition to maximize the average user satisfaction over all users. Clearly, mixing packets from all incoming flows may result in poor performance due to mismatch in type of services, priorities, and channel conditions, while mixing packets of flows with similar characteristics might be beneficial. A precise mathematical formulation of this partitioning problem will be given in the next section.

B. Satisfaction Function

Before formulating the *mixing* problem, a metric is needed in order to quantify the performance of different transmission protocols. We note that any suitable metric can be used with our proposed optimization framework. In this paper, we simply use a satisfaction function which estimates the level of satisfaction with the given quality of service (QoS). Obviously, the function depends on the number of packets received successfully in a specified period of time, and its type of service (ToS). In our paper, we consider two types of services: time-sensitive applications, e.g., video streaming, and time adaptive applications, e.g., file transfer and email.

For the time-sensitive applications, we assume that the application data is separated into layers. For example, a video streaming is packetized into basic and enhanced layers. In order to maintain the minimal QoS, a receiver needs to receive at least N_0 packets of the basic layer per time period. If the number of received packets is less than N_0, the QoS of the receiver decreases significantly. However, if more packets of the enhanced layer are received, the QoS of the receiver just increases slightly. One can think this as more details of a picture are added into the basic frame. We adopt the satisfaction function proposed in [13]. That is,

$$\gamma_i = \mathcal{F}(S_i)$$
$$= \begin{cases} 0 & S_i < \gamma_0 . N_0 \\ \gamma_0 - \gamma_0 \sqrt{1 - \frac{S_i - \gamma_0 N_0}{(1-\gamma_0) N_0}} & \gamma_0 N_0 \leq S_i < N_0 \\ \gamma_0 + (1-\gamma_0)\sqrt{1 - \frac{S_i - (2-\gamma_0) N_0}{(1-\gamma_0) N_0}} & N_0 \leq S_i < (2-\gamma_0) N_0 \\ 1 & S_i \geq (2-\gamma_0) N_0, \end{cases} \tag{2}$$

where S_i denotes the number of packets received successfully, N_0 denotes the minimal number of packets to maintain a satisfaction factor of γ_0.

For time-adaptive applications, the characteristic of satisfaction function is different. In order to obtain a satisfaction factor of γ_0, the receiver needs to receive at least N_0 packets per time period. However, when the number of received packets is less or greater than N_0, respectively, the satisfaction function decreases or increases slightly. The function is given by

$$\gamma_i = \mathcal{F}(S_i) = \begin{cases} \gamma_0 . \frac{S_i}{N_0} & S_i < N_0 \\ 1 - \frac{(1-\gamma_0) N_0}{S_i} & S_i \geq N_0 . \end{cases} \tag{3}$$

The satisfaction functions of different types of applications versus the number of packets are shown in Fig. 3. Given the satisfaction functions, we call a transmission technique is the best technique if it produces the largest expected satisfaction over all users.

IV. Problem Formulation of Adaptive Randomized Network Coding

Due to page limit, we only provide formulation for the ARNC approach. The problem formulation and analysis of the UNI and RNC techniques can be found in [14].

Let $\mathcal{G}$ denote a partition of the incoming flows and $|\mathcal{G}|$ denote the number of groups in $\mathcal{G}$. Let M_i denote the number of flows in group i. Per each group, we use RNC technique described above, to transmit the data. Consider the ith group and let $N_i = \sum_{j=0}^{M_i} n_{ij}$ and $K_i = \sum_{j=0}^{M_i} k_{ij}$ denote the total available time slots and information packets need to be transmitted for group i. Here (n_{ij}, k_{ij}) denotes the FEC code applied to the flow transmitting to receiver j of group i.[1] We have the probability that receiver j of group i, D_{ij}, can recover its all data is

$$P_{ij}^{s} = (1-p_{ij})^{k_{ij}} \left[\sum_{l=0}^{K_i - k_{ij} - 1} \binom{N_i - k_{ij}}{l} p_{ij}^{N_i - k_{ij} - l} (1-p_{ij})^{l} \right]$$
$$+ \sum_{s=0}^{k_{ij}} \binom{k_{ij}}{s} p_{ij}^{k_{ij}-s} (1-p_{ij})^{s}$$
$$\times \sum_{t=K_i - s}^{N_i - k_{ij}} \binom{N_i - k_{ij}}{t} p_{ij}^{N_i - k_{ij} - t} (1-p_{ij})^{t}. \tag{4}$$

We now calculate the probability that receiver D_{ij} recovers m out of k_{ij} original packets ($m < k_{ij}$).

$$P_{ij}(m) = \sum_{l=m}^{K_i - 1} \binom{N_i - k_{ij}}{l - m} \binom{k_{ij}}{m} p_{ij}^{N_i - l} (1-p_{ij})^{l}. \tag{5}$$

Let a random variable γ_{ij} denote the satisfaction of receiver D_{ij}. Then the average satisfaction over all users is given by

$$\gamma = E\left[\sum_{i=1}^{|\mathcal{G}|} \sum_{j=1}^{M_i} c_{ij} \gamma_{ij} \right], \tag{6}$$

where c_{ij} denotes the weighted factor (priority) for the jth flow in group i; this factor is proportional to the price that the receiver D_i has to pay to the service provider; thus, the higher price, the higher priority, and $E[.]$ denotes the expected

[1] We abuse the notation slightly since this FEC code is a some original FEC code.

function. Now, a partition scheme is optimal if it maximizes the average satisfaction over all users. This optimization problem can be formulated as:

Maximize $\{\gamma\}$

subject to:

$$\sum_{i=1}^{|\mathcal{G}|}\sum_{j=1}^{M_i} k_{ij} = K \tag{7}$$

$$\sum_{i=1}^{|\mathcal{G}|}\sum_{j=1}^{M_i} n_{ij} = N \tag{8}$$

$$0 \leq \gamma_{ij} \leq 1 \text{ for } i = 1, 2, \dots, |\mathcal{G}|, j = 1, 2, \dots, M_i \tag{9}$$

$$\mathcal{G} \in \Omega \tag{10}$$

where

$$E[\gamma_{ij}] = \mathcal{F}(k_{ij})P_{ij}^s + \sum_{m=0}^{k_{ij}-1} \mathcal{F}(m)P_{ij}(m),$$

and Ω denotes the collection of all the nonempty-subset partitions of flows $\{f_1, f_2, ..., f_M\}$, N and K respectively denote the total number of time slots and data packets in one transmission period.

V. Heuristic Algorithm for Optimal Mixing

The combinatorial optimization problem above is hard. Thus, in this section, we describe a simulated-annealing heuristic algorithm based on the Markov Chain Monte Carlo (MCMC) method to approximate the solution [15].

$\triangleright$ **Simulated-Annealing Based Algorithm (SAB)**: In this section, we show how to appropriately construct a target distribution, and use MCMC to obtain the solution. Consider a scenario with M concurrent flows traversing through the BS. Let Ω be the set of all possible partitions, and let $S(x)$ be the average satisfaction of a partition $x \in \Omega$. We represent each partition by an M-tuple group index as $x = (i, j, \dots, k)$ where i indicates that the first flow belongs to group i, the second flow belongs to the group j, and so on. The objective is to maximize the average satisfaction over all users. That is,

$$\max_{x\in\Omega} S(x) = \max_{x\in\Omega}\left\{\sum_{i=1}^{|\mathcal{G}|}\sum_{j=1}^{M_i} c_{ij}\gamma_{ij}\right\}. \tag{11}$$

We should note that the number of possible partitions in Ω is very large, that is [16]

$$|\Omega| = \sum_{k=1}^{M}\frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^M. \tag{12}$$

Hence, using exhaustive search, even for a reasonably small number of flows, is infeasible for time-sensitive applications. Moreover, every time a flow joins, terminates, or its channel condition changes, the AP needs to repartition again. Instead, by using MCMC method we will show that the time to achieve near optimal solution will be substantially reduced.

We first define the target distribution to be the Boltzmann probability density function (pdf):

$$f(x) = Ce^{\frac{S(x)}{T_B}}, \tag{13}$$

where C is a normalized factor and T_B is temperature.

As seen, if $S(x)$ is large, then $f(x)$ is large. Thus, with high probability, we will draw samples corresponding to $S(x)$ which by design, will maximize the average user satisfaction. Next, we need a mechanism for moving from one state to another in the chain. To do so, we define a neighbor of a partition in the sample space Ω:

Definition 5.1: A partition y is called a neighborhood of a partition x if and only if x and y differ in only one element.

From the above definition, y can be generated from x by replacing an element of x with a different one drawn at random from the index set $1, 2, \dots, M$. For example, $M = 5$, the partition $x = (1, 1, 3, 2, 3)$ has a neighbor $y = (1, 1, 1, 2, 3)$ since x and y differ in the third element.

We now propose a Simulated-Annealing based algorithm to generate samples according to the target distribution. We propose a transition function $q(x, y)$ from state x to one of its neighbors. Specifically, an element of x is selected uniformly at random, and then it is replaced by one of the possible indexes uniformly. Therefore, it is clear that

$$q(x, y) = q(y, x) = \frac{1}{M(M-1)}. \tag{14}$$

Consequently, the acceptance probability, i.e., the probability that the chain moves from the current state x to a new state y, is given by

$$\begin{aligned}\alpha(x, y) &= \min\left\{1, \frac{f(y)q(y, x)}{f(x)q(x, y)}\right\} \\ &= \begin{cases}1 & \text{if } S(y) \geq S(x) \\ e^{\frac{S(y)-S(x)}{T_B}} & \text{if } S(y) < S(x)\end{cases}\end{aligned} \tag{15}$$

As designed, the Boltzmann distribution becomes more and more concentrated around the global maximizer by gradually decreasing the temperature T_B. Pseudocode of the Simulated-Annealing based algorithm is described in Algorithm 1.

Algorithm 1 : Simulated-Annealing based algorithm.

Input: M, c_i, $RS(n_i, k_i)$.
Output: Optimal Flow Partition.

STEP 1: Initialize the starting state X_0 and temperature T_0. Set $n = 0$.
STEP 2: Generate a new state Y from the proposal $q(X_n, y)$.
STEP 3:
if $S(Y) \geq S(X_n)$ **then**
  $X_{n+1} = Y$
else
  $U \sim U(0, 1)$ {Generate a uniform random variable.}
  if $U < \alpha(X_n, Y) = e^{\frac{S(Y)-S(X_n)}{T_n}}$ **then**
    $X_{n+1} = Y$
  else
    $X_{n+1} = X_n$
  end if
end if
STEP 4: Decrease the temperature $T_{n+1} = \beta.T_n$ where $\beta < 1$, increase n by 1 and repeat from **STEP 2** until stopping.
STEP 5: Return a scheme x that produces the maximal weighted-average satisfaction.

$\triangleright$ **Convergence**: Guarantee of convergence to the target distribution using the SAB algorithm is shown via the following theorem:

Theorem 5.2: Samples drawn from the SAB algorithm form a Markov chain whose states satisfy the detailed balance equation:

$$\pi(x)P(x,y) = \pi(y)P(y,x) \quad \forall x, y \in \Omega, \tag{16}$$

where $\pi(x)$, $\pi(y)$ are the stationary distributions of states x and y; $P(x,y)$ and $P(y,x)$ are respectively the transition probabilities from state x to state y and vice versa.

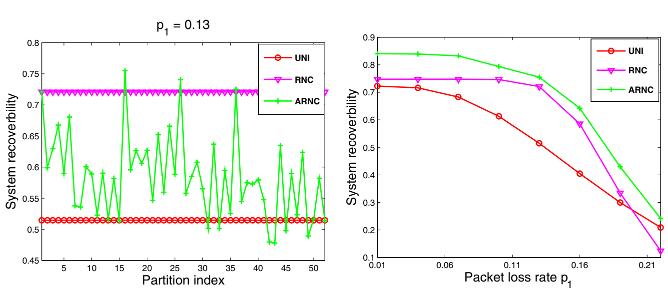
Proof: The proof is omitted due to the page limit. ■

VI. Simulations and Discussions

We consider a realistic wireless access network having a diversity in applications and channel conditions. For example, a scenario where different types of users are connecting to a base station and some of them are moving. It is not easy to set up a network with a large number of users satisfying those conditions. Moreover, in the exhaustive search method, we need to scan all the possible channel partitions, which increases exponentially with the number of flows, to find the optimal partition. Therefore, in a reasonable network settings, we consider a single-hop wireless network consisting of 5 incoming flows. In particular, we assume that there are two types of applications: time-sensitive and time-adaptive. The BS decides a coding scheme for a flow based on the cost which the user had paid to the service provider. That is, the higher cost, the higher priority. Note that the type and priority of a packet can be easily elaborated in the header of the transmitted packets. In the UNI technique, the BS uses the priorities of the incoming flows to assign their redundancies, and they will be used in all techniques for a fair comparison. We consider the typical WLAN channel with a transmission rate of 2 Mbps, equivalent to $N = 133$ time slots or 133 1.5Kbyte packets. In addition, a time-adaptive application requires 18 data packets per second, corresponding to a rate of 27 Kbps, while a time-sensitive application requires 25 and 30 data packets, corresponding to rates of 37.5 and 45 Kbps, for medium and high QoS, to achieve the satisfaction factor of 1. These numbers are equivalent with the number of frames per second in video streaming.

TABLE I
PARAMETERS OF THE TRANSMISSION FLOWS.

Flow ID	Rx ID	(n_i, k_i)	p_i	Service type	Priority
$f1$	D_1	$(21, 18)$	–	Time adaptive	1
$f2$	D_2	$(21, 18)$	0.05	Time adaptive	1
$f3$	D_3	$(22, 18)$	0.05	Time adaptive	2
$f4$	D_4	$(31, 25)$	0.05	Real time	3
$f5$	D_5	$(38, 30)$	0.05	Real time	4



(a) (b)

Fig. 4. (a) System recoverability versus partition schemes for the case of 5 flows. The transmission parameters of the flows are summarized in Table I, and $p_1 = 13\%$. (b) System recoverability versus packet loss rate p_1, the other parameters are given in Table I.

If the number of data packets received at each receiver is less than the required packets, its satisfaction will decrease according to the satisfaction function as described in Section III-B. The transmission parameters of the incoming flows are given in Table I. These parameters are set based on the types of applications, priorities of the incoming flows, and the bandwidth availability. In addition, the redundancy used for each incoming flow depends on its priority, for example, in our experiments we set priorities 1, 2, 3 and 4 corresponding to redundancies of 15%, 20%, 25% and 30%. Note that these parameters will be applied to all techniques.

A. System Recoverability

We first show the benefit of *informed* mixing and the drawback of blind mixing by examining the probability that all the receivers can decode their packets. Fig. 4(a) shows this probability versus all different partitions, i.e., ways of mixing data when the packet losses of receivers from D_2 to D_5 are set to 5% while that of receiver D_1 is 13%. We map each partition to an integer on the x-axis. The number of possible partitions is an exponential function of the number of the incoming flows, and is equal to the sum of the Stirling numbers of the second kind as shown in Eq. (12). We also plot the recoverability probabilities for UNI and RNC techniques on the same graph for comparison. They are indicated by straight lines since these techniques do not depend on the partitions. Recall that UNI does not mix packets from different flows. RNC sends the original packets then the mixed redundant packets, so the amount of mixing here is rather minimal. As seen, RNC is clearly better than UNI. It is interesting to note that, at least in this scenario, blind mixing is generally better than UNI but worse than ARNC (ARNC technique chooses the optimal partition scheme to encode and transmit packets). This is indicated by the fact that some of the partitions lie above RNC line. In fact, the proposed ARNC find precisely this best partition by mixing flows f_1 and f_4 into one group while the other flows f_2, f_3 and f_5 into another. Note that the objective of our optimization in this case is not the user satisfaction but the recoverability probability of all users. Next, we evaluate the probability that all receivers can recover the data as a function of packet loss rates. In this scenario, the packet loss rates for receivers D_i for $i = \{2, 3, 4, 5\}$ are shown in Table I while that of receiver D_1 is varied from 1% to 25%. The result is shown in Fig. 4(b). As shown, when p_1 is small, i.e., less than 18%, mixing all the incoming packets is better than transmitting them separately. This is indicated by the graph produced by RNC technique which is higher than that of the UNI technique. However, that is not the case when the packet loss rate p_1 is greater than 19%, UNI technique outperforms RNC.

Intuitively, this is because when mixing packets from all the flows, the receiver D_1 with bad channel condition will not be able to receive enough packets to decode its own packets. This alone can lead to substantial reduction on the overall recoverability. In contrast, with proper mixing, ARNC outperforms all other techniques in every scenario.

B. User Satisfaction Factor

We now evaluate the average user satisfaction as a function of the channels' conditions. In particular, we set $p_3 = p_5 = 5\%$, while varying p_1 from 1% to 20%, $p_2 = p_1 + 0.01$, and $p_4 = p_1 + 0.02$. These settings are applied to make the channel conditions more realistic in a diversity wireless network. The other parameters of the network are set the same as before as

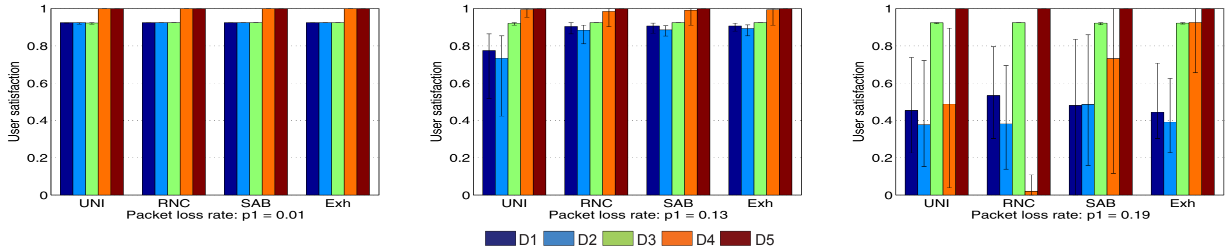


Fig. 5. Users' satisfaction factors.

given in Table I. The satisfaction of a receiver is evaluated based on the number of useful packets received successfully and the type of application that a user uses. The basic satisfactions, i.e., γ_0, of time-sensitive and time-adaptive applications are set at 0.5 and 0.6 when the number of useful received packets is 50% of the intended packets. This is because the time-sensitive applications are more vulnerable to packet loss rates. In the SAB algorithm, we set initial temperature $T_0 = 1$ and cooling down scale factor $\beta = 0.9$.

Fig. 5 represents the satisfaction factors of the users using different techniques. The left, middle and right graphs represent the ranges of low, medium and high packet loss rates, respectively. In the low loss rate, we set $p_1 = 0.01$. As seen, all the techniques satisfy QoS of all users. For all techniques, the receivers with the first and second highest priorities, i.e., D_4 and D_5, obtain satisfactions approximately 1 while the other receivers obtain a satisfaction factor around 0.9. This is intuitively plausible since in this case, resource is plenty, no optimization is needed, and all users get what they want.

In the medium loss range, i.e., $p_1 = 0.13$, all the techniques still can maintain the users' satisfaction factors at relatively high level. However, UNI technique starts reducing the QoS of the receivers having high packet loss rates with low priorities, i.e., receivers D_1 and D_2, due to its separate transmission method.

Now, in the high packet loss range, we set $p_1 = 0.19$. As seen in the rightmost graph, all the receivers with low packet loss rates, D_2 and D_5, can be kept with high QoS. However, when using the UNI and RNC, the satisfaction factors of the other receivers with high packet loss rates, D_2 and D_4, have been reduced significantly. Notably, for receiver with time-sensitive application, D_4, its satisfaction factor is decreased substantially when mixing its packets with all other flows in the RNC technique. Obviously, ARNC with exhaustive search always achieves the best performance. However, an interesting observation is that SAB algorithm can approximate the optimal solution very well with only 10 iterations. In particular, the satisfaction factor of receiver D_4 is around 0.78 and about 10% less than that of the exhaustive search, but 30% higher than that of UNI, the second best technique.

VII. Conclusions and Future Work

We have investigated the problem of how to mix flows or perform network coding at a BS in a wireless access network, in order to improve the QoS of the wireless applications. We have shown that blind mixing, in the sense that all incoming flows are mixed together, then broadcast to the receivers, may actually reduce the quality of wireless applications. We have proposed an optimization framework in which we consider the types of applications, service priorities and channels conditions to control the amount of flow mixing, in order to maximize the average quality over all flows. A heuristic algorithm called SAB is proposed to approximate optimal solution efficiently.

As our future work, there are many interesting directions that we want to extend from this paper. Mathematically finding an upper bound on the runtime of the SAB algorithm is still a theoretical open question. Other directions included design and analysis of a more sophisticated Markov chain that allows a faster convergence speed are also worthwhile to pursue.

References

[1] R. Ahlswede, N. Cai, R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, pp. 1204–1216, July 2000.

[2] S Katti, D Katabi, W Hu, H Rahul, and M. Medard, "The importance of being opportunistic: Practical network coding for wireless environments," in *Proc. 43rd Annual Allerton Conference on Communication*, 2005.

[3] Sachin Katti, Hariharan Rahul, Wenjun Hu, Dina Katabi, Muriel Medard, and Jon Crowcroft, "Xors in the air: Practical wireless network coding," in *ACM SIGCOMM*, Sept. 2006.

[4] T. Tran, T. Nguyen, B. Bose, and V. Gopal, "A hybrid network coding technique for single-hop wireless networks," *IEEE Journal of Selected Topics in Advanced Communication (JSAC) - Special Issue on Network Coding for Wireless Communication*, Jun. 2009.

[5] J. Saltzer, D. Reed, and D. Clark, "End-to-end arguments in system design," *ACM Transaction on Computer System*, vol. 2, no. 4, pp. 277–288, November 1984.

[6] D. Nguyen, T. Tran, T. Nguyen, and B. Bose, "Wireless broadcast using network coding," *IEEE Transactions on Vehicular Technology*, Feb. 2009.

[7] D. Nguyen, T. Nguyen, and X. Yang, "Wireless multimedia transmission with network coding," in *IEEE Packet Video Workshop 2007*, November 2007.

[8] A. Eryilmaz, A. Ozdaglar, and M. Medard, "On delay performance gains from network coding," in *CISS*, march 2006.

[9] T. Tran, T. Nguyen, and B. Bose, "A joint network-channel coding technique for single-hop wireless networks," in *Fourth Workshop on Network Coding, Theory, and Applications*, Jan. 2008.

[10] Y. Wu, P. A. Chou, and S.-Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," in *Technical Report MSR-TR-2004-78, Microsoft Research*, Aug. 2004.

[11] R. Koetter and M. Medard, "Beyond routing: An algebraic approach to network coding," in *Proceedings of the 2002 IEEE Infocom*, 2002.

[12] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Allerton Conf. Communication, Control and Computing Monticello*, 2003.

[13] Guowang Miao and Zhisheng Niu, "Bandwidth management for mixed unicast and multicast multimedia flows with perception based qos differentiation," in *ICC 2006, pp. 687 - 692*, 2006.

[14] T. Tran, *Throughput Optimization for Single-Hop Wireless Networks Using Network Coding*, Ph.D. thesis, Oregon State University, 2010.

[15] Mark Jerrum and Alistair Sinclair, "The markov chain monte carlo method: an approach to approximate counting and integration," in *Approximation Algorithms for NP-hard Problems," D.S.Hochbaum ed., PWS Publishing, Boston*, 1996.

[16] M. Abramowitz and I. A. (Eds.) Stegun, "Handbook of mathematical functions with formulas, graphs, and mathematical tables," in *9th printing. New York: Dover, pp. 824-825*, 1972.