XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks

Fang-Chun Kuo*, Kun Tan†, XiangYang Li†‡, Jiansong Zhang†, and Xiaoming Fu*
*Institute of Computer Science, University of Göttingen, Germany, Email: {fkuo, fu}@cs.uni-goettingen.de
†Microsoft Research Asia, Beijing, China, Email: {jiazhang,kuntan}@microsoft.com
‡ Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL, USA, Email: xli@cs.iit.edu

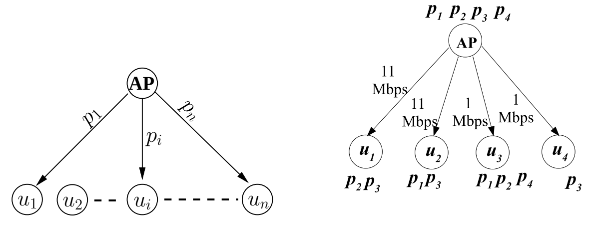
***Abstract*— It is well-known that wireless links are error-prone and require retransmissions for recovering frames from errors and losses. Network coding (NC) has been proposed for more efficient MAC-layer retransmissions in WLANs. However, existing schemes employed the reception report mechanism, which is both inefficient and expensive. Furthermore, they considered neither fairness nor the effects of time-varying heterogeneous wireless networks. These issues are critical for achieving full benefit of network coding. Without addressing them, these schemes may even impair system performance. In this paper, a novel MAC-layer retransmission scheme, namely XOR Rescue(XORR) is proposed. It estimates the reception status without extra overheads and devises a new coding metric, which accommodates the effects of the frames size and the channel condition. Finally, XORR employs NC-aware fair opportunistic scheduling, which is theoretically proven to be *fair*, i.e. not only the service time is evenly allocated, but also it *always* improves the expected goodput for *every* wireless station. It is further verified by theoretic analyses, extensive simulations and testbed experiments. Our results show that XORR outperforms the non-coding fair opportunistic scheduling and 802.11 by 25% and 40%, respectively.**

I. INTRODUCTION

The proliferation of 802.11 wireless networking products has encouraged the deployment of wireless local-area networks (WLANs). However, it is well-known that wireless networks are error-prone due to fading, interference and collisions. Recent measurement on IEEE 802.11-based WLANs has revealed that wireless links suffer from moderate to severe frame losses (20-60%) [2] [1].

Traditionally automatic repeat request (ARQ) is employed at the MAC-layer in contemporary WLANs for recovering corrupted frames. But ARQ-based retransmissions consume a significant portion of channel capacity in lossy wireless networks. Therefore, forward error coding (FEC) has been exploited to enhance conventional ARQ schemes, e.g. Hybrid ARQ [11], for recovering corrupted frames from partially correct receptions.

Recently network coding (NC) was employed for improving ARQ's efficiency in wireless networks [7], [8] . The philosophy of this NC-aided ARQ is illustrated in Fig. 1 (a). Assume that in a wireless network, the access point (AP) transmits the frame p_1 to the station u_1 and the frame p_2 to the station u_2, respectively. Both frames are lost. Due to the broadcast nature of wireless media, u_1 could overhear p_2 while u_2 could overhear p_1. In this case, instead of retransmitting these two frames individually, the AP can use NC and retransmit a *coded frame*, $p_1 \oplus p_2$. Both u_1 and u_2 can recover their lost frames once they receive the coded frame correctly. Thus, one retransmission can be saved. The potential coding gain of the NC-aided ARQ has been demonstrated in ER [7] and MU-ARQ [8].



(a) (b)

Fig. 1. (a) NC reduces the retransmissions in WLANs. (b) Example of network coding in the multi-rate WLAN.

However, the design in ER and MU-ARQ only focuses on coding, but fails to solve the following challenges in practice:

- **Overheads for reception information**. To make the coding decision, the AP requires the knowledge of all frames overheard by each station, which are not destined to it. In previous work, this reception information is explicitly reported to the AP by each station, per frame [8] or periodically [7][1]. Such reception reports result in significant overheads which may overwhelm the coding benefit. Although overheads may be amortized by choosing a longer report interval, it will cause insufficient reception information at the AP. Hence, the coding benefit may vanish due to less coding opportunities. Our investigation in Sec. VI shows that it is difficult to make a trade-off, as it depends on many dynamic parameters such as the number of stations and wireless channel conditions.
- **Heterogeneous and time-varying networks**. Every NC scheme defines a coding metric for measuring the potential benefit of the candidate of coding-sets. ER and MU-ARQ choose the coded frame which enables the most stations to retrieve their frames. This coding metric, namely *maximal retrieval*, may reduce retransmissions, but does not necessarily improve the overall throughput unless the AP always transmits to all stations at a common rate. In current WLANs, the AP can adjust its transmission rate based on individual station's channel quality. Therefore, such a coding strategy may result in suboptimal performance, since the coded frame must be transmitted at the *lowest* transmission rate in the set of destined stations. This is illustrated by the example in Fig. 1 (b). The

---

[1] A reception report records the frames which are not destined to the station itself, while the ACK is a message to acknowledge receipt of a frame destined to itself.

station u_1 already overheard the frames p_2 and p_3, while the transmission rate between the AP and u_1 is 11Mbps. Similarly, the reception status and transmission rates of the other stations are also shown in Fig. 1 (b). With maximal retrieval, a coded frame, $p_1 \oplus p_2 \oplus p_3$, is transmitted firstly at the rate of 1Mbps, and p_4 is transmitted alone at the rate of 1Mbps. However, a better solution is transmitting a coded frame, $p_1 \oplus p_2$, at 11Mbps and another coded frame, $p_3 \oplus p_4$, at 1Mbps. This reduces the transmission time by 45%.

• **Frame scheduling**. A single shared queue is used for buffering frames in IEEE *802.11*. More specifically, each node maintains a single first-in-first-out (FIFO) transmission queue and all frames targeted for different destinations are buffered in a common FIFO queue. ER [7] inherits this single shared FIFO queue from *802.11* and further defers the retransmissions until a pre-defined threshold is reached for achieving more coding opportunities. However, the single FIFO queue causes head-of-line (HOL) blocking. Furthermore, deferring retransmissions in the FIFO queue causes frame-reordering which has an adverse impact on TCP.

A novel NC-aided retransmission scheme, namely XOR Rescue (XORR), is proposed in this paper. It is designed for addressing the aforementioned challenges by taking into account not only network coding but also its integration with the link characteristics and MAC layer:

1) The AP in XORR probabilistically estimates reception status based on the Bayesian-learning estimation technique. Therefore extra signaling overheads incurred by sending reception reports are completely eliminated.

2) A new coding metric, namely expected goodput, is used in XORR. Expected goodput is defined as the achievable system goodput by transmitting a certain frame. It naturally accommodates the heterogeneities in wireless networks, such as transmission rate, link reliability and frame size.

3) The problems of HOL blocking and frame reordering can be solved easily by using a per-station queue for each station. Only one frame is scheduled for each station till the frame is acknowledged (ACKed). However, the station which overhears fewer frames will have less possibility to have its frames coded at the AP. This may lead to starvation of a station experiencing a bad channel for prolonged periods of time. Therefore, an NC scheduler should have per-station queues and also consider fairness for avoiding those problems. To the best of our knowledge, this paper is the first work on the *NC-aware fair scheduling* in WLANs.

XORR is evaluated by theoretical analyses, extensive simulations as well as experiments on a prototype system. It is proved theoretically that XORR guarantees *temporal fairness* and *always* improves the goodput for *every* station. Our simulation results show that XORR can significantly reduce the number of retransmissions by 10-60% in various situations including heterogeneous and time varying channels. XORR outperforms ER by $10-25\%$, the non-coding fair opportunistic scheduler by $20-25\%$, and the existing IEEE 802.11 network by $30-40\%$, respectively. Furthermore, our XORR prototype is deployed on a real five-station wireless test-bed with pretty reliable links (loss rate $< 20\%$). Our experiments show that XORR improves goodput by 10.7% for UDP traffic and by 15.7% for TCP traffic, compared to the IEEE 802.11 network.

The rest of this paper is organized as follows. Related work is reviewed in Sec II. We present an overview and theoretical analysis of XORR. in Sec. III and IV, respectively. The design of XORR is presented in Sec V. Our simulations and experimental studies are presented in Sec. VI and VII, respectively. Finally, we conclude our paper in Sec. VIII.

II. RELATED WORK

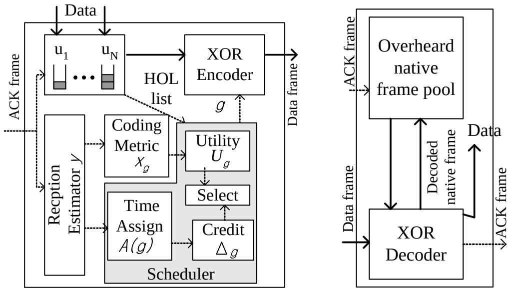
Recently, network coding has been found as an innovative means to enhance network performance by mixing information at intermediate nodes [3], [5], [6]. In particular, COPE [5] developed a practical network coding scheme for unicast in multi-hop wireless networks. It utilized network coding for initial transmissions only, but relied on MAC-layer ARQ for recovering losses. By contrast, XORR uses network coding for MAC-layer retransmissions in single-hop WLANs, which is complementary to the existing work.

It has been demonstrated in [4] that NC should be jointly considered with scheduling for maximizing the network capacity. However, Ref. [4] mainly focused on characterizing the capacity region of joint NC and scheduling, but no practical algorithms were presented. Furthermore, due to omitting opportunistic listening [4], its scheduler is not suitable for NC-aided retransmissions in WLANs. XORR presents not only a primary design of an NC-aware fair opportunistic scheduling for retransmissions in WLANs, but also a practical heuristic algorithm for efficient coding selection.

Two pieces of work most related to XORR are ER [7] and MU-ARQ [8]. MU-ARQ presented a theoretic performance analysis on an ideal model, where the AP has full knowledge of all frame receptions, while each station stores all received frames. MU-ARQ characterized the bounds of capacity improvement in such a system only in theory. XORR is designed with more practical guidelines. Our goal is not only to improve network throughput, but also to maintain fairness among all stations. Beside theoretical analysis, we also implemented XORR and tested it in a real environment.

ER provided a practical design to use NC for reducing retransmissions. But its design focused only on coding, without considering multi-rate support and time-varying link characteristics in WLANs. It also employed periodic reception reports, which cause significant signaling overheads. In contrast, XORR completely eliminates signaling overheads by probabilistically estimating reception status. Furthermore XORR overcomes frame reordering and HOL blocking, which was seen in ER. Consequently XORR works well in multi-rate wireless networks and guarantees to always improve the performance of every station.

A work was recently proposed about the reliability gain of NC for reliable multicast in multi-hop wireless networks [12]. Ref. [12] presents an analysis on the expected number of transmissions using ARQ, FEC and NC, with tree-based reliable multicast, and shows NC reduces the need of transmission. XORR focuses on improving the performance of single-



(a) AP side (b) Station side

Fig. 2. XORR Architecture.

hop wireless networks with NC-based ARQ. XORR shows significant performance improvement even for unicast traffic.

III. Overview

Consider a WLAN, where there is a wireless AP and a set of N stations, i.e., $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ which are associated with the AP. Stations only communicates with the AP directly. Let r_j and γ_j denote the transmission rate and the link reliability of between the AP and the station u_j, respectively. Assume that all wireless links are mutually independent, i.e., the variables $r_i, \gamma_i, (1 \leq i \leq N)$ are independent. Due to the broadcast nature of the wireless medium, when the AP transmits a frame to a station u_i, another station $u_j (j \neq i)$ may overhear the frame with a probability of γ_j.

(b) AP (sender) side: The system architecture of XORR is shown in Fig. 2. The AP maintains a queue for each station. For preventing frame reordering, only the HOL frames of the stations are the candidates for scheduling selection. Let p_i denote the HOL frame of the station u_i. For the purpose of scheduling, the AP classifies all frames into two groups: if this HOL frame p_i has never been transmitted before, then p_i belongs to $TxGroup$ and is called an *original frame*; Otherwise, it belongs to $RetxGroup$.

For making coding decisions, the AP estimates the reception status of all stations. This information is used by the coding metric calculator for evaluating the coding benefits of all possible coding-sets. Based on the coding metric calculation, the scheduler selects either one frame or multiple frames, which achieve the best performance under the fairness constraint. If multiple frames are selected, these frames are encoded using the XOR operation. These XORed frames are called *coded frames*, while un-XORed frames are called *native frames*.

(c) Station (receiver) side: When a station u_i overhears a native frame $p_j (j \neq i)$, it stores p_j into its native frame pool and may use it for decoding other coded frames. When a coded frame p_g (g denotes the set of frames that are encoded.) is received, u_i tries to decode *any* native frame immediately. A station can successfully decode a native frame from p_g only if it has all other frames in its local frame pool. If the station u_i successfully decodes a frame destined to itself, i.e., p_i, it should transmit an acknowledgment (ACK) to the AP for confirming the reception. Upon receiving the ACK, the AP removes p_i from the queue. If the decoded frame is for other stations, i.e. $p_j (j \neq i)$, it is stored in the local frame pool. If the coded frame fails to be decoded, it is discarded.

IV. Understand coding gain

Before delving into the detailed design, this section provides some insights into the expected performance gain of XORR and the factors affecting it. Note that for simplicity a simple static model is considered in this section, where all stations in the network have the same transmission rate r as well as the same link reliability γ, all frames have the same size and all N stations are always backlogged. In Sec. V, we will further prove that XORR can always outperform the non-coding scheme in the dynamic and heterogeneous wireless scenario.

Assume that the AP has the perfect knowledge on reception status. The AP starts to recover lost frames when the number of frames in $RxGroup$ reaches a threshold, which is denoted by $\delta, \delta \leq N$. Otherwise, the AP transmits original frames. Let $\mathbf{K}$ denote the expected coding size (*i.e.* the number of native frames XORed in a coded frame). An upper and a lower bound on the expected *coding-set* size, $\mathbf{K}$, is presented in Lemma 1[2].

Lemma 1: The expected coding-set size $\mathbf{K}$ satisfies,

$$\sum_{\kappa=1}^{\delta} 1 - (1 - \gamma^{(\kappa-1)\kappa})^{\lfloor \frac{\delta}{\kappa} \rfloor} \leq \mathbf{K} \leq \sum_{\kappa=1}^{\delta} 1 - (1 - \gamma^{(\kappa-1)\kappa})^{\binom{\delta}{\kappa}}.$$

As shown in Lemma 1, a larger retransmission threshold δ results in a larger coding-set size $\mathbf{K}$. In other words, a better scheduler can opportunistically defer retransmissions for providing more coding opportunities, so that it can potentially encode more frames into one retransmission. However, the threshold δ is restricted by the number of backlogged stations N, *i.e.* $\delta \leq N$. Therefore, if there are more backlogged stations in the network, the threshold can be adjusted to a higher value for more coding opportunities.

The *coding gain* is defined as the ratio of the goodput achieved by XORR to that by the non-NC approaches. Theorem 2 characterizes the coding gain of XORR with the coding-set size $\mathbf{K}$.

Theorem 2: The coding gain of XORR is $B = \frac{\mathbf{K}}{1-\gamma+\gamma\mathbf{K}}$.

Theorem 2 shows that a better coding gain is achieved with a larger coding-set size. Thus, there is no coding gain ($B = 1$) when no coded frame is transmitted ($K = 1$). Combining with Lemma 1, the coding gain is affected by the link reliability γ, the threshold δ and the number of stations N.

In order to investigate the impact of the number of stations, we let $\delta = N$. In other words, we can always get the maximal coding gain under certain number of stations. Table I gives some numerical results of the lower and upper bounds of XORR's coding gain with respect to different numbers of stations N in the network. It shows that with a moderate number of N, XORR can effectively reduce retransmissions and thus improve the system performance.

V. XORR Design

A. Reception estimation

The AP maintains a score-table $\mathcal{Y}$ that has $N \times N$ entries. Each entry $y_{i,j}$ represents the probability that u_i has the HOL

---

[2] Owing to the space limitation, please refer to our report [15] for the proofs of all lemmas and theorems.

	$N = 10$		$N = 100$		$N \to \infty$	
γ	Lower	Upper	Lower	Upper	Lower	Upper
0.8	1.15	1.19	1.18	1.23	1.21	1.23
0.6	1.26	1.39	1.35	1.52	1.45	1.55
0.4	1.29	1.54	1.47	1.88	1.73	2.60
0.2	1.14	1.58	1.59	2.15	2.09	2.78

TABLE I
NUMERICAL RESULTS OF CODING GAIN.

native frame p_j destined to station u_j. The reception table $\mathcal{Y}$ is updated once a frame is sent (either a native frame p_j or a coded frame p_g) or an ACK is received.

Initially, the table contains all zeros. When a native frame p_j is transmitted, then the probability that u_i does not have p_j after the transmission is the joint probability of two events: u_i has no p_j before the transmission and u_i does not receive this transmission. Thus, we have

$$y_{i,j}^{t+1} = 1 - (1 - y_{i,j}^{t})(1 - \gamma_i), \ i \neq j. \tag{1}$$

When a coded frame p_g is transmitted, the estimation of $y_{i,j}$ depends on if the station is in the set g or not. If the station u_i is not in the set, i.e., u_i, $i \notin g$, it may decode a native frame p_j in g, if possible. Then, the probability that u_i ($i \notin g$) does not have p_j after the transmission is the joint probability of two events: u_i does not have p_j before the transmission and u_i fails to decode p_j. Therefore, we have

$$y_{i,j} = 1 - (1 - y_{i,j}) \cdot \left[(1 - \gamma_i) + \gamma_i (1 - \prod_{q \in g \setminus \{j\}} y_{i,q}) \right] . \tag{2}$$

The term in the square-bracket represents the probability that u_i fails to decode p_j, which might be caused by either reception failure (the first part) or an insufficient number of native frames received for decoding the coded frame (the second part).

There are two cases if the station u_k that are in the set, i.e., u_k, $k \in g$. If the AP receives an ACK from u_k, it means that u_k has successfully decoded its frame p_k from p_g. This implies that u_k must have all other native frame p_q, $q \in g \setminus \{k\}$. Thus, the AP will update $y_{k,q}$ as

$$y_{k,q}^{t+1} = 1, \forall q \in g \setminus \{k\}$$

If u_k fails to acknowledge, the reason may be either that u_k fails to receive the transmission or that it does not have all needed native frames for decoding. Thus, $y_{k,j}$, $j \in g \setminus \{k\}$ is estimated based on the Bayes-law. Define $\overline{y_{k,j}} = 1 - y_{k,j}$ and $\Pr\left(\overline{ACK_k^{t+1}}\right)$ as the probability that u_k does not acknowledge at time $t+1$. Then, we have $\overline{y_{k,j}^{t+1}} = \frac{\overline{y_{k,j}^{t}}}{\mathbf{Pr}\left(\overline{ACK_k^{t+1}}\right)}$. Note $\mathbf{Pr}\left(\overline{ACK_k^{t+1}}\right) = (1 - \gamma_k) + \gamma_k (1 - \prod_{q \in g \setminus \{j\}} y_{k,q})$. Thus, $y_{k,j}^{t+1}$ is updated as

$$y_{k,j}^{t+1} = 1 - \frac{1 - y_{k,j}^{t}}{(1 - \gamma_k) + \gamma_k (1 - \prod_{q \in g \setminus \{j\}} y_{k,q}^{t})}. \tag{3}$$

When a station u_i successfully decodes its own frame p_i, it transmits an ACK to the AP. Since p_i will never be transmitted again, the corresponding column in $\mathcal{Y}$ is reset to zero for initializing the estimations of the next frame to u_i. ACK can piggyback the station's information about its received native frames for further facilitating XORR recovery. When the AP receives an ACK piggybacking such information, the AP updates the corresponding estimation to 1.

B. Coding metric

XORR uses *expected goodput*, the achievable system goodput by transmitting a certain frame, for evaluating the coding benefit. The expected goodput naturally takes into account the coding effect, transmission rates and the status of wireless link.

The expected goodput of transmitting a native frame p_j is $r_j \gamma_j$, while the goodput of transmitting a coded frame p_g depends on the probability that the coded frame can be decoded by its stations, namely *decoding capability*. The *decoding capability* can be estimated from the score-table $\mathcal{Y}$ maintained by the AP.

Definition 1 (Decoding capability): The decoding capability $D_{i,g}$, $i \in g$ is the probability that the station u_i can retrieve its frame p_i by decoding a coded frame p_g. For the station u_i, the decoding capability is $D_{i,g} = \prod_{j \in g \setminus \{i\}} y_{i,j}$.

Let L_g be the size of p_g, then $L_g = \max_{j \in g} L_j$. Let r_g be the transmission rate for transmitting the coded frame, p_g, then $r_g = \min_{j \in g} r_j$. Thus, the transmission time of p_g is $T_g = \frac{L_g}{r_g}$. Finally, the expected goodput χ_g of transmitting the coding-set g is the sum of the expected goodput of each station $u_i, i \in g$, i.e.,

$$\chi_g = \sum_{i \in g} \chi_{i,g} = \sum_{i \in g} \frac{L_i}{T_g} \cdot \gamma_i \cdot D_{i,g}. \tag{4}$$

As implied by Definition 1, the decoding capability $D_{i,g}$ decreases with the size of the coding-set. In other words, the more frames are included in the coded frame, the less chance the coded frames is decoded. Thus, the expected goodput of some coding-set is low if the coding-set includes many frames.

It is straightforward that XORR should not encode any original frame with other frames, since such a coded frame is not decodable for any station. Therefore, XORR only applies NC in *RetrGroup*. Since the maximal number of frames in *RetxGroup* is N, each station only needs a queue containing $(N - 1)$ native frames in its local frame pool for decoding.

Furthermore, we define the *biased* expected goodput χ_g^*:

$$\chi_g^* = \begin{cases} \theta \chi_i, & \text{if } p_i \text{ is an original frame} \\ \chi_g, & \text{otherwise} \end{cases} \tag{5}$$

where χ_g is the expected goodput of coded frame g. $\theta \geq 1$ is called *deferring retransmission factor*. The deferring retransmission factor *theta* gives some bias for scheduling original frames. By biasing for original frames, the AP defers the recovery of a lost frame for a while so it may exploit potential coding opportunities.

In order to be integrated into our scheduling, the coding metric will be mapped to *utility* defined in the scheduling, which will be detailed in Sec. V-C.

C. NC-aware fair opportunistic scheduling

Traditionally fair opportunistic scheduling was proposed for improving the system utility by exploiting multiuser diversity [9] under fairness constraints. In XORR, fair opportunistic scheduling is extended for NC by selecting *a coding-set*. Its utility for measuring the scheduling benefit is mapped from the *coding metric* and is maximized under the *temporal* fairness constraint. Note that although temporal fairness is adopted in XORR, our mechanism is actually general enough to accommodate other fairness definitions.

For supporting NC, XORR extends the credit-based approach in [9], which maintains bounded temporal fairness among all stations. Each station is assigned a state variable, *credit*, for maintaining fairness, which is denoted as K_i. The *deficit* credit of a station in the coding-set g is defined as its assigned service time minus its credit, i.e., $\Delta_i = \mathcal{A}(g,i) - K_i$. The *deficit* credit of the coding-set g is defined as the maximal deficit credit of all its member stations, i.e., $\Delta_g = \max_{i \in g} \Delta_i$.

Then, XORR scheduler is defined as follows:

$$\hat{g}^t = \arg\max_g \left(U_g^t - \Delta_g^t\right), \tag{6}$$

where U_g^t is the utility of transmitting the coding-set g. XORR scheduler balances between the transmission utility and fairness. It tries to select a coding-set (possible with only one frame) that maximizes the utility while having minimal service time *deficit* for maintaining fairness. A station accumulates its credit if it is not selected in a coding-set.

If the coding-set contains only one frame, the service time is simply the channel time needed for transmitting the frame. However, when transmitting a coded frame, the transmission time T_g is shared among the stations in that coding-set. Let $\mathcal{A}(g,i)$ denote the portion of service time assigned to station u_i, $i \in g$. Obviously, we have $\sum_{i \in g} \mathcal{A}(g,i) = T_g$, where T_g is the overall time for transmitting the coded frame p_g. We will elaborate how service time is shared later in Section V-D.

Following the scheduling decisions, all backlogged stations update their credits as described in Fig. 3. Once a set g is selected and the coded frame is transmitted, all the stations in g decrease their credits by the fraction of service time assigned to them (Line 2-4). If any station has deficit ($\Delta_g > 0$), all stations adjust their credits by adding Δ_g (Line 6-8). As a result, unscheduled stations may accumulate their credits and all stations have non-negative credit values.

```
1: function UpdateCredit($g$)
2: for $u_j, j \in g$ do
3:     $K_j \leftarrow K_j - \mathcal{A}(g,j)$
4: end for
5: if $\Delta_g > 0$ then
6:     for all $u_j \in \mathcal{U}$ do
7:         $K_j \leftarrow K_j + \Delta_g$
8:     end for
9: end if
```

Fig. 3. Pseudo-code for updating credits.

The following theorem shows that the scheduling discipline defined in Eq. (6) achieves bounded temporal fairness.

Theorem 3 (Temporal fairness): With XORR scheduler, for any two stations u_i and u_j that are continuously backlogged over any interval $[t_1, t_2)$, we have

$$|\alpha_i(t_1,t_2) - \alpha_j(t_1,t_2)| \le \max_t \frac{L_i^t}{r_i^t} + \max_t \frac{L_j^t}{r_j^t} + 2U_{max},$$

where $\alpha_i(t_1,t_2)$ is the service time assigned to the station u_i during the time interval $[t_1,t_2)$, L_i^t and r_i^t are the frame size and the transmission rate of u_i at time t, respectively.

Utility function. The utility represents the coding benefit and also affects the fairness bound as shown in Theorem 3. Therefore, the utility function of a coding-set g is defined as an increasing function of the expected goodput of g and is bounded as shown in Eq. (7).

$$U_g = \beta \cdot T_{\max} \cdot \left(1 - e^{-\frac{\chi_g^*}{r_{\max}}}\right), \tag{7}$$

where $T_{\max} = \frac{\max_i L_i}{\min_i r_i}$ is the maximum transmission time of a coded frame, $r_{\max} = \max_i r_i$ is the maximum transmission rate and χ_g^* is the *biased* expected goodput. Obviously, U_g is upper-bounded by βT_{max}, where β is called *utility scaling factor*, which balances the opportunistically improved system performance and fairness [9].

D. Fair service time assignment

In the non-coding fair scheduling, the goodput of a station is linearly determined by the service time assigned to the station. Accordingly, temporal fairness implies certain goodput fairness among stations. However, with NC, such implication becomes tricky because the time for transmitting a coded frame can be arbitrarily assigned. Our simulations show that even though the service time is evenly allocated by the scheduler, an improper service time assignment algorithm may cause some stations to perform worse with an NC-aware fair scheduler than they would with a non-NC fair scheduler. It is called that such stations have *coding loss*. For clarifying fairness in the NC-aware scheduler, we define *NC-fairness* as:

Definition 2 (NC-fairness): An NC-aware scheduler achieves *NC-fairness* if it maintains fairness, i.e., the service time is evenly allocated to all stations, and no station has *coding loss* when applying it.

Therefore, for achieving *NC-fairness*, the service time assignment algorithm has to be designed carefully. Two terms are used for explaining our assignment algorithm.

Definition 3 (Relative coding edge): The relative coding edge ψ_i^g of u_i in a coding-set g is the ratio of the expected goodput of u_i using network coding to that without coding, i.e., $\psi_i^g = \frac{\chi_{i,g}}{r_i \cdot \gamma_i}$.

Definition 4 (Effective goodput): The effective goodput λ_i of u_i in a coding-set g is the expected decoded bits divided by the assigned service time, i.e., $\lambda_i = \frac{\chi_{i,g} \cdot T_g}{\mathcal{A}(g,i)}$.

The following theorem gives a time assignment strategy. It guarantees that the effective goodput of every station in the coding-set is no less than that if its native frame is transmitted alone.

Theorem 4: $\forall u_i \in g$. If the service time is assigned proportionally to the *relative coding edge* of each station in the coding-set, i.e.,

$$\mathcal{A}(g,i) = T_g \cdot \frac{\psi_i^g}{\sum_{j \in g} \psi_j^g}, \tag{8}$$

we have $\lambda_i \ge r_i \gamma_i$.

More specifically, Theorem 4 implies that in each scheduling, transmitting the coded frame improves the instant goodput for every station, compared to transmitting its native frame alone.

Theorem 4 only considers the instant goodput in one scheduling time. However, the scheduling patterns for NC and non-NC schedulers are different during a period of time. Theorem 4 does not answer whose accumulated goodput is better over a period of time. The following theorem shows that the XORR always achieves NC-fairness in any duration.

Theorem 5: Given any scheduling policy $\mathcal{L}$ that achieves temporal fairness, let λ_i^{XORR} and $\lambda_i^{\mathcal{L}}$ denote the goodput of u_i with and without XORR, respectively. If the service time assignment strategy defined in Eq. (8) is applied to XORR, XORR achieves NC-fairness i.e.,

$$E(\lambda_i^{XORR}) \geq E(\lambda_i^{\mathcal{L}}).$$

E. Scheduling flow

Fig. 4 outlines the scheduling flow in XORR. The *scheduling* function loop searches the best scheduling candidate sets in $TxGroup$ and $RetxGroup$, respectively. The result of the selection is fed into the encoder (Line 8). At the end of scheduling, the credits are updated by the function *UpdateCredit* (whose pseudo-code is shown in Fig. 3).

Let Ψ denote the set of stations in $RetxGroup$. The following theorem shows that exhaustively searching the best coding-set in $RetxGroup$ is NP-hard. In other words, it is computationally expensive and is not feasible for practical use.

Theorem 6: Finding an optimal coding-set g at time t is NP-hard and cannot be approximated within $|\Psi|^{1-\epsilon}$ unless NP=ZPP, for arbitrary small $\epsilon > 0$.

Thus, a heuristic algorithm is described in Fig. 5. The algorithm starts with finding the station in $RetxGroup$ which maximize utility minus deficit credit. This station is added into the coding-set. Then, the algorithm tries to search again in the remaining stations of $RetxGroup$ and find another station to form a better coding-set. This process continues until no more stations can be added or all stations in $RetxGroup$ are selected. Therefore, the complexity of the heuristic algorithm is $O(|\Psi|^2)$. The algorithm is evaluated by simulations in our report [15]. It shows that our heuristic selection algorithm only slightly degrades the performance, compared to the optimal selection (computed by exhaustive searching).

```
 1: function scheduling
 2: loop
 3:     $g_{tx} \leftarrow \arg\max_{j \in TxGroup} U_j - \Delta_j$
 4:     $g_{rx} \leftarrow$ SelectCodingSet $(RetxGroup)$
 5:     if $U_{g_{tx}} - \Delta_{g_{tx}} < U_{g_{rx}} - \Delta_{g_{rx}}$ then
 6:         $g_{tx} \leftarrow g_{rx}$
 7:     end if
 8:     EncodeAndTransmit($g_{tx}$)
 9:     UpdateCredit $(g_{tx})$
10: end loop
11: end function
```

Fig. 4. Pseudo-code for XORR scheduling.

VI. PERFORMANCE EVALUATION

A single-hop wireless network having an AP and N stations is considered in our simulations. The transmission rate of each link between the AP and a station can be 1, 2, 5.5 or 11 Mbps, as specified in IEEE 802.11b. The size of data frames is 1500 bytes. Both ACK and feedback frames have a size of 50 bytes and are always transmitted at the base rate of 2 Mbps. The effects of varying the utility scaling factor β and the deferring retransmission factor θ in XORR is extensively investigated in our report [15]. Using the results in [15], the utility scaling factor is set as $\beta = 50$ for balancing between fairness and goodput performance, while the deferring retransmission factor is set as $\theta = 2$ for achieving sufficient coding opportunities. Unless otherwise mentioned, by default the number of stations is $N = 10$, the transmission rate is $r = 5.5Mbps$ and the simulation time is 100 seconds.

```
function SelectCodingSet(RetxGroup)
$g \leftarrow \emptyset$
repeat
    $\hat{j} \leftarrow \arg\max_{j \in RetxGroup \backslash g} U_{g \bigcup j} - \Delta_{g \bigcup j}$
    $\hat{g} \leftarrow g \bigcup \hat{j}$
    if $U_{\hat{g}} - \Delta_{\hat{g}} < U_g - \Delta_g$ then
        break
    else
        $g \leftarrow \hat{g}$
    end if
until $(g == RetxGroup)$
return $g$
end function
```

Fig. 5. Pseudo-code for heuristic coding-set selection

XORR is compared with the following schemes in the context of both static and time-varying channels:

1) *Opportunistic scheduling* (labeled as *Opp*). It uses a similar scheduling strategy as that in XORR, except that there are only native frames but no coded frames to be scheduled.

2) *IEEE 802.11-based WLAN.* (labeled as *802.11*) This is a baseline for existing WLANs, where a shared FIFO queue is used for all stations and a frame is retransmitted immediately once its loss is detected.

3) *ER.* This is a prior NC-aided MAC-layer retransmission scheme [7]. Unlike XORR, ER neither employs opportunistic scheduling nor considers temporal fairness. In addition, ER relies on feedbacks from stations for obtaining reception status. We implement their *sort-by-time* coding algorithm and use 25 as the threshold for the retransmission queue.

A. Performance metrics

XORR is evaluated from two aspects, system performance and fairness. For evaluating system performance, *goodput gain* and *reduced retransmission ratio* are defined. The baseline scheme for evaluating system performance is *802.11.*

$$\text{Goodput gain} = \frac{\text{goodput of the scheme}}{\text{goodput of 802.11}} - 1.$$

$$\text{Reduced retransmission ratio} = \frac{\textit{ReTxRatio} \text{ of the scheme}}{\textit{ReTxRatio} \text{ of 802.11}} - 1,$$

where *ReTxRatio* is the ratio of total number of retransmissions to that of transmissions.

For evaluating fairness, two metrics are defined, *fairness index* and *coding improvement ratio*. Fairness index was defined in [16] and its value ranges between 0 and 1. If it equals 1, it means the service time is allocated evenly. Coding improvement ratio is used for checking if the individual station has *coding loss* when compared to *Opp*:

$$\text{Coding improvement ratio} = \frac{\lambda_i^{XORR}}{\lambda_i^{Opp}} - 1,$$

where λ_i^{XORR} and λ_i^{Opp} are the goodput of u_i using XORR and using *Opp*, respectively. If coding improvement ratio is less than zero, it means the station suffers from *coding loss*. Note that under all simulations, the calculated *fairness index* is close to 1 for *Opp* and XORR.

B. Static channels

(a) Impact of link reliability Fig 6 and Fig. 7 show the impact of different link reliabilities on the goodput gain and the reduced retransmission ratio for XORR, *Opp*, and ER having a report period of 10, 50 and 200 ms (labeled as ER-10ms, ER-50ms, ER-200ms). All links have the same reliability, which varies from 0.2 (least reliable) to 0.9 (most reliable).

As shown in Fig. 6, *Opp* has the same goodput gain as that of *802.11* in the context of homogeneous networks, because all stations have exactly the same transmission rate and reliability, and thus no multi-user diversity gain can be utilized.

Both XORR and ER employ network coding for reducing retransmissions. As shown in Fig. 7, when the link reliability is higher than 0.8, over 60% of retransmissions are saved. On the other hand, when the link reliability is low, their reduced retransmission ratios are also small, because stations receive less native frames, which results in less coding opportunities.

The coding efficiency of ER heavily depends on feedback information carried by reception reports. Thus ER having a shorter report period may reduce retransmissions further, as shown in Fig. 7. The reduced retransmission ratio of ER-10ms is even higher than that of XORR. However, Fig. 6 shows that the coding gain of ER-10ms is worse than that of XORR, and even worse than that of *802.11* when the link reliability is high. This is because signaling overheads incurred by frequent reports degrade the goodput performance severely. When the number of the station is 10, ER-50ms performs best with the most link reliabilities than other ER schemes. So unless otherwise mentioned we use 50 ms as the report period for ER when the number of the user is around 10.

By contrast, XORR estimates the reception status, which mitigates signaling overheads, thus outperforms ER. As the link reliability decreases, more native frames are lost and need to be retransmitted. XORR improves the goodput by reducing retransmissions as depicted in Fig. 6. When the reliability is around 0.5, XORR's goodput gain peaks near 25%. However, when the reliability decreases further, its goodput gain bends down, because XORR relies on reception estimation to select coding-set for retransmissions. If there are significant losses, the estimation accuracy decreases due to less ACK received.

(b) Impact of the number of stations

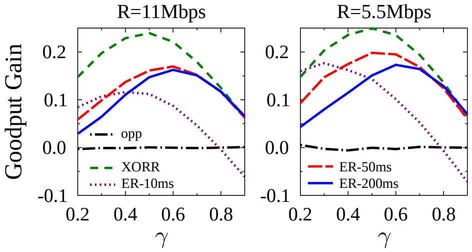


Fig. 6. Goodput gain with various link reliability in static channels.

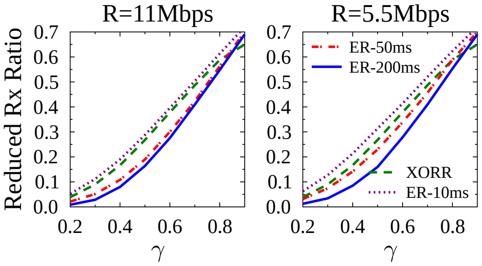


Fig. 7. Reduced retransmission ratio with various link reliability in static channels.

Fig. 8 illustrates the impact of different number of stations, N, on the goodput gain of XORR, ER-10ms, ER-50ms and ER-200ms, while the link reliability, γ, is 0.2, 0.5 or 0.8, respectively. As N increases, the goodput gain also increases for both XORR and ER. This is expected since there are more coding opportunities. However, the goodput gain of ER bends down when $N > 10$. This is due to overheads of reception reports. More stations introduce more feedback frames, which overwhelms the coding gain in ER. In addition, although ER outperforms XORR in few conditions as shown in Fig. 6 and Fig. 8 (e.g. ER-10ms with $\gamma = 0.2$ and $N \leq 10$ in Fig. 8), it is difficult to adjust the report period for ER. Because the optimal period depends on the data transmission rate, link quality and the number of stations, which in practice are typically time-varying. For example, as shown in Fig. 8, ER-10ms performs the best when $\gamma = 0.2$ and $N < 20$, while 50ms is the optimal period when $\gamma = 0.2$ and $N \geq 20$.

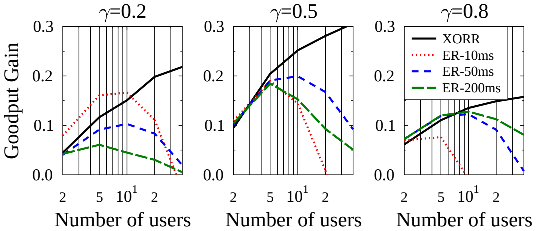
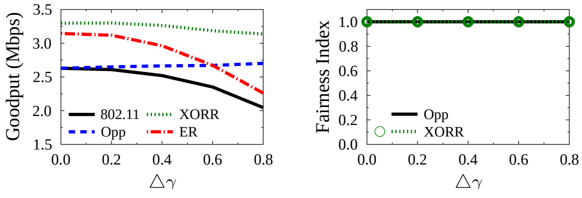


Fig. 8. Goodput gain with different number of stations in static channels.

(c) Heterogeneous networks

In Fig. 9(a), the goodput of four schemes are evaluated in the context where links may have heterogeneous reliabilities. How to model channel conditions realistically is beyond the scope of this paper. For the sake of simplicity, each link's reliability is randomly chosen from $[\gamma_{min}, \gamma_{max}]$. The mean of each link's reliability is $E(\gamma) = 0.5$. The interval length $\Delta\gamma = \gamma_{max} - \gamma_{min}$ varies from 0 to 0.8.

When $\Delta\gamma$ is large, the goodput of *802.11* drops dramatically, while *Opp*'s goodput remains unchanged. This is be-



(a) (b)

Fig. 9. (a) Goodput in heterogeneous and static channels, where $E(\gamma) = 0.5$. (b) Fairness index.

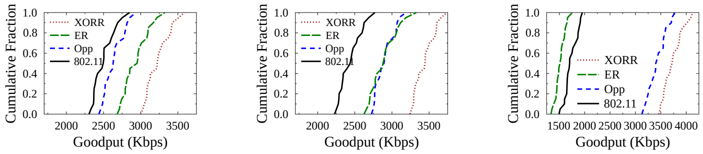
cause that *802.11* allocates more channel time to stations with worse channel conditions [10], while *Opp* always allocates equal service time to all stations. Therefore, *Opp* performs better than *802.11*.

Due to the similar reason to *802.11*, ER's system goodput also decreases as $\Delta\gamma$ increases. On the other hand, XORR performs best with all values of $\Delta\gamma$ because it not only effectively reduces retransmissions with NC but also maintains temporal fairness as *Opp* does. The goodput of XORR drops slightly as $\Delta\gamma$ increases. This is because when $\Delta\gamma$ is large, most retransmissions are for the stations with low link reliabilities. Hence, there are less effective coding opportunities. Furthermore, Fig. 9(b) shows the fairness index for *Opp* and XORR. It demonstrates that the fairness index $\simeq 1$, i.e. the service time is evenly allocated in both *Opp* and XORR. Note that in all simulations, the calculated fairness index is close to 1. Hence, we omit it in the following results.

C. Time-Varying Channels

In practice, wireless channels are time-varying. The varying speed of channel conditions is typically characterized by the *coherence time*, within which the channel may be considered as "static" [14]. The coherence time of a wireless channel depends on the station's mobility and its surrounding environment. Assume that the channel is stationary. ($\bar{\gamma}_i$, $\sigma_{\gamma,i}$) and ($\bar{r}_i$, $\sigma_{r,i}$)[3] are used for characterizing the time-varying reliability and transmission rate of link l_i, respectively. The mean of each link's reliability is randomly chosen from [0.3,0.7]. The coherence time is 24.45 ms, corresponding to a fast walking speed of 5 m/s [14].

Fig. 10 shows the Cumulative Distribution Function (CDF) of the goodput when the transmission rate is 5.5Mbps while the variance of the link reliabilities is $\sigma_\gamma = 0.1$ and $\sigma_\gamma = 0.01$, respectively. In Fig. 10(a), *Opp* performs slightly better than 802.11. This is because the channel variance is very small, so that little multi-user diversity can be exploited. By contrast, when the channel varies more drastically, as shown in Fig. 10(b), *Opp* scheduling improves the system goodput more significantly by serving the stations with better channel conditions. ER does not use opportunistic scheduling for exploiting multi-user diversity, so its goodput does not change when the channel variance varies. XORR exploits not only coding-gain but also multi-user diversity. As a consequence, it outperforms ER by $10-25\%$, *Opp* by $20-25\%$, and *802.11* by $30-40\%$, respectively.



(a) $\sigma_\gamma = 0.01$ (b) $\sigma_\gamma = 0.1$ (c)

Fig. 10. Goodput in time-varying channel. $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$. Coherence time is 24.57 ms.(a) (b) static and homogeneous $r = 5.5$Mbps. (c) time-varying r, $\sigma_\gamma = 0.1$

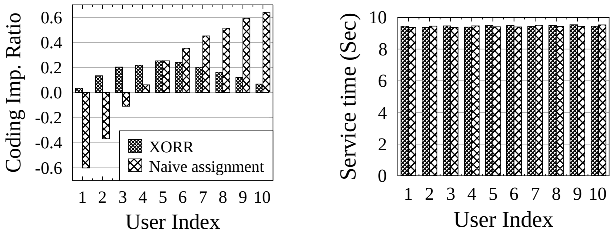
Fig. 10(c) shows the goodput performance with time-varying transmission rates. Each station's transmission rate varies among 2, 5.5 and 11Mbps. As shown in Fig. 10(c), ER performs worst because its coding gain is offset by the inappropriate coding scheduling, which does not consider the link condition. This was also illustrated in [4]. Thus ER not only fails to exploit multi-user diversity, but also loses the coding gain. *Opp* achieves higher goodput than *802.11* by exploiting multi-user diversity. XORR, performs best by exploiting both multi-user diversity and the network-coding.

D. Service time assignment

As aforementioned, service time assignment is critical for designing a fair NC scheduling. In order to understand the impact of the service time assignment, we compare our algorithm in Eq. (8) with a naive service time assignment, which evenly distributes the transmission time among all stations in the coding-set. The link reliability for each wireless link is randomly chosen from [0.2,0.9]. Fig. 11 shows the coding improvement ratio and allocated service time of each station using both time assignment algorithms in XORR. All stations are sorted by their link reliabilities in the figure. Although the overall service time over a period of time is evenly allocated to all stations in both algorithms, when the naive service time assignment is used in XORR, the stations with less reliable links have *coding loss*. By contrast, XORR improves the goodput of all stations while maintaining fairness.

E. Impact of estimation error

XORR relies on reception estimation for selecting coding-sets. To estimate the reception of native frame for each user, the AP needs to estimate the link reliability. Many existing wireless system already maintains such statistics (*e.g.* WLAN [13]). We now evaluate XORR in the case that reliability

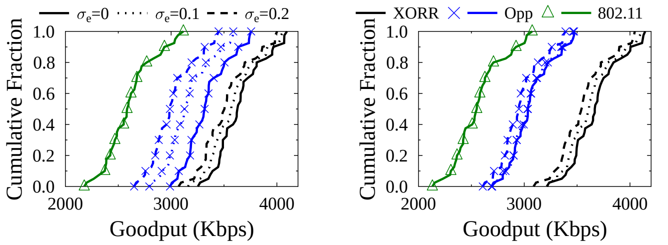


(a) Coding improvement ratio (b) Allocated service time

Fig. 11. Starvation of XORR with naive service time assignment.

---

[3] In network like IEEE 802.11, there are only a small set of transmission rate that can be used. Therefore, we actually use the transmission rate index instead of transmission rate directly.



(a) Coherence time 6.14 ms. (b) Coherence time 24.57 ms.

Fig. 12. Goodput with estimation error in time-varying channel model where, $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$.

	XORR/802.11	Opp/802.11	XORR/Opp
UDP	10.7%	2.5%	8.0%
TCP	15.7%	1.0%	14.5%

TABLE II

GOODPUT IMPROVEMENT IN TEST-BED EXPERIMENTS. XX/YY MEANS THE GOODPUT IMPROVEMENT OF XX OVER YY.

estimations have errors. To model this, we artificially add a noise in the link reliability estimation, $\gamma_e = \gamma_c + e$, where e is a random variable following the normal distribution $\mathcal{N}(0, \sigma_e)$. Fig. 12 summarizes the CDF of the network goodput under two different coherence times, 6.14 (vehicle speed) and 24.57ms (walking speed), respectively. When there exists an estimation error, the network goodput of XORR does have slightly degradation. But overall, the impact of estimation error is limited, especially when the channel does not change very fast. It is interesting to note that XORR is less sensitive to estimation errors compared to *Opp*, as shown in Fig. 12(a). It may be because the network coding actually could average the errors out. Therefore, it is less significant if a user misses the transmission due to estimation errors, as it may get a coding opportunity later in the waiting queue.

VII. TEST-BED EXPERIMENTS

We have prototyped XORR and preliminarily evaluated its performance on a real wireless test-bed. Our implementation is based on Atheros AR5212 wireless NICs on the Windows platform. We use broadcast to emulate all transmissions and rely on software to generate ACKs. The test-bed contains 6 VIA EPIA mini-ITX boxes, each of which has a Netgear WAG511 802.11a/b/g card. One machine works as an AP that directly communicates with 5 other machines. We conduct the experiments in a typical office environment. We fix the transmission rate at 11Mbps. The links between the AP and stations have an average reliability of 80%. Table II shows a summary of goodput gain on our test-bed with both UDP and TCP flows. Note that *Opp* does not have much gain compared to *802.11* because in our environment the channel condition is rather stable. The results show that XORR does improve the network goodput compared to both *802.11* and *Opp*. The coding gain XORR obtained over *Opp* is 8.0% with UDP flows and 14.5% with TCP flows, respectively. It is interesting to note that XORR has more performance gain with TCP. It is because TCP is more sensitive on frame losses due to its congestion control scheme. As XORR significantly reduces frame losses, it improves TCP performance more significantly.

VIII. CONCLUSION

In this paper, we presented XOR Rescue (XORR), an efficient NC-aware scheduling for MAC retransmissions. We conducted extensive simulations and test-bed experiments to investigate the performance of XORR. Our results showed that, by exploiting both multi-user diversity and network coding, XORR has a consistent improvement over the non-coding schemes (802.11 and traditional opportunistic scheduling); while the existing NC-aided ARQ retransmission scheme sometimes even causes negative effect and thus performs worse than 802.11. Furthermore, as shown by our theoretical proof and simulations, while maintaining fairness, XORR scheduler achieves a better goodput for *every* station in the system, compared with traditional opportunistic schedulers. Our scheme can be generalized for uplink traffic, or a mixture of downlink and uplink traffic. The detailed scheme is omitted due to space limit.

REFERENCES

[1] D. Aguayo, J. Bicket, S. Biswas, G. Judd, and R. Morris. Link-level Measurements from an 802.11b Mesh Network. In *ACM SIGCOMM*, August 2004.

[2] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris. A HighThroughput Path Metric for MultiHop Wireless Routing. In *ACM MOBICOM*, 2003.

[3] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung. Network Information Flow. *IEEE Transactions on Information Theory*, 46(4):1204–1216, July 2000.

[4] P. Chaporkar and A. Proutiere. Adaptive network coding and scheduling for maximizing throughput in wireless networks. In *ACM Mobicom*, pages 135–146. 2007.

[5] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Medard, and J. Crowcroft. XORs in the air: Practical wireless network coding. In *ACM SIGCOMM*, June 2006.

[6] Y. Wu, P. A. Chou, and S.-Y. Kung. Information exchange in wireless networks with network coding and physical-layer broadcast. Technical Report MSR-TR-2004-78, Microsoft, August 2004. http://www.princeton.edu/∼yunnanwu/.

[7] E. Rozner, A. P. Iyer, Y. Mehta, and L. Qiu. ER: Efficient Retransmission Scheme for Wireless LANs. In *CoNext*, Dec 2007.

[8] P. Larsson and N. Johansson. Multi-User ARQ. In *IEEE Vehicular Technology Conference 2006-Spring*, pages 2052 - 2057, 2006.

[9] Y. Liu, S. Gruhl, and E. W. Knightly. WCFQ: an opportunistic wireless scheduler with statistical fairness bounds. In *IEEE Transcation on Wireless Communication*, September 2003.

[10] G. Tan and J. Guttag. Time-based Fairness Improves Performance in Multi-rate Wireless LANs. In *The USENIX Annual Technical Conference*, 2004.

[11] S. Lin and D. Costello. Error Control Coding. Prentice Hall, Uppser Saddle River, NJ, 2004.

[12] M. Ghaderi, D. Towsley, and J. Kurose. Reliability gain of network coding in lossy wireless networks. In *INFOCOM* , 2008.

[13] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan. Robust rate adaptation for 802.11 wireless networks. In *ACM MobiCom*, pages 146–157, 2006.

[14] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly. Opportunistic media access for multirate ad hoc networks. In *ACM MobiCom*, pages 24 – 35, 2002.

[15] F.-C. Kuo, K. Tan, X.-Y. Li, and X.-M. Fu XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks Available at http://www.cs.iit.edu/∼xli/paper/Submitted/xorr-full.pdf

[16] Raj Jain, D. Chiu, and W. Hawe A Quantitative Measure Of Fairness And Discrimination For Resource Allocation In Shared Computer Systems. DEC Research Report TR-301, September, 1984