XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks

Fang-Chun Kuo* Kun Tan† XiangYang Li‡ Jiansong Zhang † Xiaoming Fu*

ABSTRACT

Users increasingly depend on WLAN for business and entertainment. It is well-recognized that wireless links are prone to errors. Previous work, ER, proposed to use network coding (NC) for providing more efficient MAC-layer retransmission scheme in WLAN. However, it uses inefficient and costly reception report scheme and does not consider the effect of heterogenous and time-varying wireless conditions and fairness. These issues are critical for getting full benefits of network coding. We show that, without addressing them, NC may even cause negative effect on the system. In this paper, we present a novel MAC-layer retransmission scheme, namely XORR, which uses reception estimation without extra overhead and adopts NC-aware opportunistic scheduling with maintaining temporal fairness in WLAN. We prove our NC-aware scheduling algorithm is *fair* and it will *always* improve the expected goodput for *each* wireless clients. We further verify XORR with extensive simulation as well as experiment studies and find that our scheme outperforms traditional opportunistic scheduling (without NC) and 802.11 about 25% and 40%, respectively.

1. INTRODUCTION

The proliferation of 802.11 wireless networking products have encouraged the development of wireless local-area network (WLANs). However, wireless networks are notorious for error-prone natures, because of time-varying channel fading, interference or collisions. Recent measurement on IEEE 802.11-based WLAN has revealed that many wireless links suffer from moderate to severe frame losses (20-60%) [1, 4]. For the sake of reliable communication, contemporary WLANs use automatic repeat request (ARQ), a MAC-layer retransmission mechanism, to recover the corrupted frames. Nevertheless, when the channel's quality deteriorates for a long period, ARQ-based retransmissions triggered by a missing link-layer ACK become ineffective and wasteful. In this paper, we focus on utilizing network coding (NC) for efficient retransmission in single-hop wireless networks, where many mobile stations are communicating directly with an access point (AP).

*Institute of Computer Science, University of Göttingen, Germany, Email: {fkuo, fu}@cs.uni-goettingen.de
†Microsoft Research Asia, Beijing, China, Email: {kuntan,jianzhang}@microsoft.com
‡Dept. of Computer Science, Illinois Institute of Technology, Chicago, IL, USA, Email: xli@cs.iit.edu

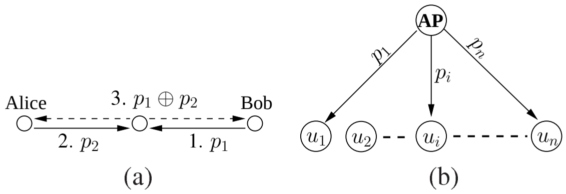


Figure 1: Illustration of Network Coding. (a) The basic Alice-and-Bob scenario. Here a number in front of a packet denotes the time-instance when the packet is sent. (b) Network coding reduces the number of retransmissions in single-hop wireless network.

Network coding (NC) is an emerging technology to increase the utilization of both wired and wireless networks. It was firstly proposed in the context of multicast in wired networks by Ahlswede et.al. [2]. They showed that having the intermediate nodes mix information in different messages can achieve maximum multicast capacity. Recently, network coding has been adapted to support unicast applications in wireless networks by exploiting the broadcast nature of the wireless medium [3, 9, 15, 18, 19]. This can be illustrated using the simple example in Alice-and-Bob scenario [18], as shown in Fig. 1(a). In this scenario, Alice wants to send frame p_1 to Bob and Bob wants to send p_2 to Alice. They need a *relay* in the middle to exchange the frames. The traditional strategy needs total 4 transmissions. With network coding, the *relay* XORs the two frames and broadcasts the mixed frame. Accordingly, both Alice and Bob can decode the needed frame when receiving *coded frame*. Thus, 3 transmissions are needed. Network coding improves the network throughput compared to traditional schemes by reducing the required transmissions from four to three (33% coding gain).

The above wireless network coding schemes mainly focus on multi-hop wireless networks. In such coding scheme, there are no coding opportunities for single-hop path [13]. However, it has been shown in [13, 20] that network coding can be used for combating frame losses in single-hop wireless networks. As shown in Fig. 1(b), assume the AP transmits frame p_1 to u_1 and p_2 to u_2. Both frames to u_1 and u_2 are lost, but u_1 overhears p_2 and u_2 overhears p_1. After getting reception report from the users, the AP can retransmit a *coded frame*, $p_1 \oplus p_2$, instead of any single frame. Thus, NC can reduce the retransmission by utilizing opportunistic listening, i.e. a broadcast wireless medium creates many opportunities for nodes to overhear packets so that a node may use overheard frames for decoding. Note that the opportunistic listening is not necessary for NC in multi-hop wireless network, but is essential for NC in single-hop wireless network. Without opportunistic listening, the retransmitted coded frames cannot be decoded.

Although previous work [13, 20] has demonstrated that NC can be used to enhance MAC-layer retransmission, three critical design issues due to the effect of wireless characteristics have not been considered:

- **How to avoid/reduce the burden of reception report**? Since the wireless medium is a scare resource, the signaling overhead of sending reception report is expensive. Our simulation in Section 5 also shows that the reception report scheme is inefficient.
- **How to adapt NC to heterogenous and time-varying wireless environment**? To this end, a more recent work [3] has presented a joint NC and scheduling scheme. However, their scheduling scheme is not suitable for NC in single-hop wireless networks. Because [3] omits the effect of opportunistic listening from their scheduling, but opportunistic listening is essential for applying NC in single-hop wireless network.
- **How to maintain fairness when using NC**? Network coding schemes designed only to maximize the overall throughput could be unfairly biased. In wireless networks, one important fairness concept is *the temporal fairness* [10, 11, 16]. However, it is non-intuitive to assign service time for users when they are mixed in a coded frame. To the best of our knowledge, this is the first attempt to identify the issue of time assignment when a temporal fairness is applied to an NC scheduler.

By addressing above issues, we present a novel MAC-layer retransmission scheme, namely XOR Rescue (XORR), which exploits opportunistic scheduling and network coding in WLAN under temporal fairness constraint. Our main contributions are summarized as follows.

1. The Bayesian learning technique is used for the estimation of the reception status, which incurs no extra signaling overhead.
2. An opportunistic NC scheduling tailored to MAC-layer retransmission in single-hop wireless network is designed.
3. We design a NC-aware fair scheduler with a novel *fair service time assignment* algorithm for network-coded frames. We theoretically prove that our scheduling guarantees *temporal fairness* and will *always* improve the expected throughput for *each* wireless client.
4. We theoretically characterizing the potential network coding gain for lossy wireless environment.
5. We evaluate XORR with extensive simulations. Our results show that XORR can significantly reduce the number of retransmissions (10-60%) in various situations with heterogenous and time varying channels. Thus, XORR improves the network capacity by $10-30\%$ compared to the traditional opportunistic scheduling without coding, and over 40% regarding to existing IEEE 802.11 network.
6. We implement a prototype of XORR and conduct experiments in a real five-client wireless test-bed. Our results show XORR improves goodput by 8% for UDP traffic, and 14.5% for TCP traffic compared to the traditional opportunistic scheduling without coding.

The rest of the paper is structured as follows. We review the related work in Section 2. We present an overview of the design approaches of XORR in Section 3 and detailed design of XORR in Section 4. The simulation and experimental studies of our protocols are presented in Section 5 and Section 6, respectively. Finally, we conclude our paper with future works in Section 7.

2. RELATED WORK

XORR builds on prior work on loss resilience, wireless network coding, and opportunistic scheduling.

Loss Resilience. ARQ, a MAC-layer retransmission mechanism, is commonly used to handle unreliability. Due to the inefficiency of ARQ in lossy wireless networks, previous work, like PPR [21], SOFT [23], and Hybrid ARQ [22], has been proposed to exploit partially correct receptions to recover the corrupted frames. XORR takes a different approach from the above work: instead of applying FEC or physical layer information to frame retransmission, it utilizes network coding to provide a more efficient retransmission scheme.

Wireless Network Coding. Recently, network coding has been found as an innovative means to enhance the wireless communication by taking advantage of the broadcast nature of the wireless medium [9, 18]. In particular, COPE [9] develops a practical network coding scheme for unicast in multi-hop wireless networks. It utilizes network coding for the original transmissions. However, in such coding scheme, there are no coding opportunities for single hop wireless networks [13]. In contract, XORR uses network coding to make the MAC-layer retransmission more efficient in single hop scenario.

Opportunistic scheduling. Opportunistic scheduling has been proposed to improve the network performance by exploiting *multiuser diversity* [8, 10, 11]. It has been demonstrated in [3] that network coding should be jointly considered with scheduling to maximize the network capacity. However, Ref. [3] mainly focuses on characterizing the capacity region with joint NC and scheduling. No practical algorithms are presented. Furthermore, due to omitting opportunistic listening from the scheduling [3], it is not suitable for the NC-aided retransmission in single-hop wireless networks. XORR presents not only a primary design of an opportunistic NC scheduling for retransmissions but also a heuristic algorithm for efficient coding selection.

One work most related to XORR is ER [13], which extends the COPE algorithm [9] to retransmission scheduling. ER shares the same idea of using NC for retransmission with XORR, but with three differences. First, ER relies on periodical reception report, while XORR embraces a probabilistic regime to estimate the reception status without additional control frame. Second, ER uses a simple scheduling policy that defers retransmission in a retransmission queue up to a predefined threshold. However, XORR builds up an opportunistic NC scheduler to exploit the gain of multi-user diversity as well as network coding. Third, XORR ensures the temporal fairness among users and has a proven property to always improve the performance of each user in unreliable networks.

A work is recently proposed about the reliability gain of NC for reliable multicast in multi-hop wireless networks [5]. Ref. [5] 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-hop wireless networks with NC-based ARQ. XORR shows significant performance improvement even for unicast traffic.

3. OVERVIEW

We introduce XORR, a new retransmission scheduling based on network coding for single-hop wireless networks. Table 1 summarizes the terms used in the rest of the paper.

Term	Definition
Native frame	A non-coded frame
Coded frame	A frame that is XORed from multiple native frames
Original frame	A native frame that is first being transmitted
Retransmitted frame	A native that is being retransmitted or a coded frame that contains only lost native frames
Coding set	A set of native frames that are encoded in a Coded frame

Table 1: Definitions of terms in this paper.

3.1 System model

In this paper, we mainly consider single-hop wireless networks. As shown in Fig. 1(b), there is a wireless access point (AP) and a set of clients $\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ that are associated to the AP. Clients can only communicate to the AP directly. Each link between the AP and a client u_j is denoted as l_j, which has associated a time-varying transmission rate r_j and a link reliability γ_j. We consider a packet network, where the AP transmits data to each user in unit of frame. We assume that the AP maintains a per client queue, and we denote p_j as the HOL (head-of-line) frame to user u_j. We denote L_j as the size of p_j. Therefore, when the AP selects p_j to transmit, the transmission time is $T_j = \frac{L_j}{r_j}$.

We assume all wireless links are independent, i.e., the variables r_i $(1 \leq i \leq N)$ are independent and γ_i $(1 \leq i \leq N)$ are independent. Due to the broadcast nature of the wireless medium, all clients may overhear the transmissions of other nodes in the network. When the AP transmits a frame to a client u_i, another client u_j can successfully receive the frame with a probability of γ_j. When a client successfully decodes a frame targeting at it, the client would send the AP an acknowledgement (ACK) to confirm the reception of the frame. Upon receiving the ACK from the client, the AP will remove the frame from the transmission queue and a new frame will move to the front of the queue.

In this paper, we focus on the downlink traffic, because that most of the traffic in a wireless network is downloading. Later, we will show in Section 4 that XORR can also be beneficial in two-way traffic scenario.

3.2 Coding and decoding

In this work, encoding a *coding-set* g, a group of frames, is to perform XOR operations on these frames and generate a coded frame $p_g = \bigoplus_{i \in g} p_i$. The resulted frame p_g has a frame length equal to the maximal frame length of the coding-set L_g, or $L_g = \max_{i \in g} L_i$.

XORR employs a few principles in coding and decoding. First, we employ an *immediate decoding principle*. When receiving a *coded frame*, the client tries to decode any native frame immediately from its past received frames. If the decoded frame is not for the client itself, it is stored in a buffer and can be used to decoded the following coded frames. If the client fails to decode, the coded frame is discarded silently. The second principle we adopt is *never encode an original frame*. A frame is called *original* if it is a native frame that has never been transmitted before. So, if an original frame is XORed with other frames, no client can decode this coded frame. As a consequence, in XORR, network coding only happens among the frames which need retransmission. The final principle is that we *never XOR two frames to the same client.* This is because a coded frame XORs of these two lost frames can not be decoded by that client. Therefore, XORR only encodes retransmissions of the HOL frames of different clients. Thus, each client in XORR only needs to keep up to $(N-1)$ native frames for decoding, where N is the number of active clients in the network.

3.3 Understanding coding gain

One nature question on XORR is how beneficial it is. This section provides some insight into the expected performance gain of XORR and the factors that affect it. Here, we present some insights using a simple model where all users in the network have the same transmission rate r and link reliability γ. Assume that all frames are of same size and all users are always backlogged.

We define the *coding gain* as the ratio of goodput achieved by XORR, to that by the current non-coding approaches.

Clearly, the coding gain of XORR depends on the wireless link reliability. If the links are perfect reliable, then XORR may provide no coding gain since no retransmission is needed. Obviously, the coding gain of any scheme is bounded from above by $\frac{1}{\gamma}$.

Let λ denote the expected goodput with the current non-coding approach. We have $\lambda = \gamma \cdot r$. Now let's consider the expected goodput with XORR. Here, we assume the AP has the perfect knowledge on which native frames have been received by each user. Hence the AP only selects a coding-set, which is decodable at every targeted receiver when the client receives it. The AP starts to recover lost frames when the number of users waiting for retransmission reach the threshold, denoted by $\mathfrak{N}$; otherwise, the AP sends the original frames. Let $\mathbf{K}$ denote the expected coding size (*i.e.* number of native frames used to get a coded frame).Theorem 1 characterizes the coding gain of XORR. The proof is outlined in Appendix.

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

According to the previously defined coding and scheduling policy, an upper and a lower bound on the expected *coding-set* size, $\mathbf{K}$, is presented in Lemma 2. We prove Lemma 2 in Appendix.

LEMMA 2. *The expected coding-set size* $\mathbf{K}$ *satisfies,*

$$\sum_{\kappa=1}^{\mathfrak{N}} 1-(1-\gamma^{(\kappa-1)\kappa})^{\lfloor \frac{\mathfrak{N}}{\kappa} \rfloor} \leq \mathbf{K} \leq \sum_{\kappa=1}^{\mathfrak{N}} 1-(1-\gamma^{(\kappa-1)\kappa})^{m} \quad (1)$$

where $m = \binom{\mathfrak{N}}{\kappa}$.

As shown in Lemma 2, a larger $\mathfrak{N}$ results in a larger $\mathbf{K}$. In other words, a better scheduler can opportunistically defer the retransmission to create more coding opportunities, so that when doing loss-recovery, it can potentially encode more frames into one retransmission.

Table 2 presents some numerical results of the expected lower and upper bounds of XORR coding gain with respect to different $\mathfrak{N}$. We can see with a moderate number of $\mathfrak{N}$, XORR can effectively reduce the retransmissions and thus improve the system performance.

	$\mathfrak{N}=10$		$\mathfrak{N}=100$		$\mathfrak{N}\to\infty$	
γ	Lower	Upper	Lower	Upper	Lower	Upper
0.9	1.08	1.09	1.09	1.11	1.10	1.11
0.8	1.15	1.19	1.18	1.23	1.21	1.23
0.7	1.21	1.29	1.26	1.37	1.32	1.38
0.6	1.26	1.39	1.35	1.52	1.45	1.55
0.5	1.29	1.49	1.41	1.70	1.60	1.78
0.4	1.29	1.54	1.47	1.88	1.73	2.60
0.3	1.24	1.57	1.55	2.08	1.88	2.40
0.2	1.14	1.58	1.59	2.15	2.09	2.78
0.1	1.04	1.32	1.34	1.93	1.84	3.08

Table 2: Numerical results of coding gain.

3.4 Scheduling

The core part of XORR is the scheduling. Whenever a transmission opportunity occurs, the *scheduling discipline* selects a native frame or coded frame to transmit. Previous schedulers combining with network coding [3, 19] are designed only to maximize the overall throughput. This could be unfairly biased, especially when there are users with widely disparate distances from the AP. Therefore, the task of the scheduling discipline in XORR is to optimize the system performance under certain fairness constraint. In wireless networks, one important fairness concept is *the temporal fairness* [10,11,16]. Under temporal fairness scheduler, each user may allocate an equal service time (air-time) instead of throughput.

Our XORR follows the temporal fairness concept. Unlike prior work which only selects a single frame to transmit, XORR may select a set of users and XOR their HOL frames into one transmission. We denote U_g^t as the utility if a set, g, is selected to code and transmit at time t, $|g| \geq 1$. We assume $U^t(.)$ is non-negative and bounded. We denote $\vec{U}^t = \{U_g^t\}, g \in 2^{\mathcal{U}}$ the performance vector of all possible sets at time t. XORR scheduling can be defined in Eq. (2).

$$\hat{g}^t = \arg\max_{g} U_g^t, \ s.t. \, \alpha_i = \alpha_j, \, i \neq j, \quad (2)$$

where α_i is the average service time allocated to user i, and the $\arg\max$ means $\hat{g}$ is the set that maximizes the utility. Note that the traditional scheduling disciplines can be regarded as a special case of the preceding model, where g always contains only one user.

Before delving into the details of XORR in Section 4, we briefly outline three issues that challenge the design of XORR, which are not mentioned or fully addressed in previous NC approaches.

(a) Reception Estimation: Obtaining reception information is crucial for the success of coding because the AP needs to ensure a high likelihood that the coded frame can be decoded by multiple clients. In ER the AP knows explicit reception status by getting periodic reports from users. However, through our simulations, such reception report mechanism causes the following problems: 1) The signaling overhead for reception report offsets the coding gains. 2) Using larger period of report can alleviate the burden of signaling overhead, but results in less coding efficiency. 3) The period is difficult to adjust because the optimal period depends on the sending rate, the link quality and the number of users, which are normally time-varying in wireless networks. The mis-chosen period may even degrade performance severely. Therefore, in this work, we propose a Bayes-based approach to estimate the up-to-date reception status without additional feedback.

(b) Coding-set Selection: Previous design, ER, simply chooses a set of frames that contains maximum frames and is decodable by all destinations. Such coding-set selection may not be optimal or even degrade the performance because different frames may have different size; instantaneous wireless links also have heterogeneous reliability and transmission rate, referred to as *multiuser diversity*. Similar observation has also been reported in [3]. However, the scheduling in [3] cannot be applied to XORR directly because opportunistic listening, the essential source of network coding opportunities for retransmission in single-hop wireless networks, is omitted from the scheduling. Thus we design a scheduling algorithm for selecting coding-sets which considers varying link rate, link reliability and the impact of frame size. Furthermore, owing to our estimation scheme for reception status, the AP only has the probability of the reception status. Hence, the coding-set selection in XORR should be designed based on this probabilistic framework.

The following example explains how network coding can be beneficial with this probabilistic framework. Assume all have a data rate $r = 1$, a frame size 1, and a delivery probability of $\gamma_i = 70\%$. Assume the AP transmits frame p_1 to u_1 and p_2 to u_2 and both frames to u_1 and u_2 are lost. Instead of getting feedback from u_1 and u_2, the AP estimates that u_2 and u_1 may overhear p_1 and p_2 with a probability of 70%, respectively. Then the probability that u_1 can recover its frame when sending a coded frame, $p_1 \oplus p_2$, is $(0.7)^2$, because it should receiver both p_2 and $p_1 \oplus p_2$. Same happens to u_2. Thus, the expected goodput by sending this coded frame is $(0.7)^2 + (0.7)^2 = 0.98$, while the expected goodput by sending any native frame is 0.7. Hence, sending the coded frame, $p_1 \oplus p_2$, has about 40% coding gain.

(c) Fairness: With temporal fairness, one important task of AP is to assign equal service time to all clients. However, with network coding, it becomes non-intuitive to count *service time* for clients when they are mixed in a coded frame. For example, if two frames p_1 and p_2 are encoded and transmitted, how much time is served for u_1 and u_2, respectively. One naive solution may evenly distribute the transmission time between these two clients. But in fact, this strategy causes significant unfairness, because different users may have different decoding ability. Consider an example as follows. The link reliability of u_1 and u_2 is 0.8 and 0.6, respectively. AP transmits p_1 and p_2 and both of them get lost. Then, AP XORs p_1 and p_2 to opportunistically recover both losses. This is reasonable since the expected recovered frames are $0.8^2 + 0.6^2 = 1$, larger than any single retransmission. But u_1 and u_2 may have different probability to decode the frame, since u_1 has higher probability to have p_2 than u_2 has p_1. Consequently, equally distributing the transmission time of the XORed frame among these two users brings unfairness to u_2. Thus, u_2 may even have less goodput than in non-coding cases, or even be starved in some cases as verified in our simulations.

4. XORR DESIGN

4.1 Opportunistic scheduling framework

XORR maximizes the network performance under temporal fairness constraint. To provide a bounded short-term temporal fairness among all clients, XORR follows a credit-based approach as in [11] that assigns a state variable, *credit*, to control the fairness property, but XORR extends the framework to support network coding. Denote K_i as the credit of client u_i.

As aforementioned, when a coded frame is transmitted, it needs a *service time assignment* to distribute the actual transmission time to all users that are encoded. Assume $\mathcal{A}$ is a service time assignment algorithm that determines the fraction of service time for each member when a coded frame is served. Let g be the coding-set. $\mathcal{A}\ (g, i) = \delta_i$ is the fraction of assigned service time for u_i, $i \in g$. Algorithm $\mathcal{A}$ should satisfy $\sum_{i \in g} \delta_i = T_g$, where T_g is the overall transmission time for that coded frame p_g.

The "deficit" of each user in the coding-set g can be calculated as $\Delta_i = \delta_i - K_i$. We further define the "deficit" of the coding-set g as the maximum of all its member users, or $\Delta_g = \max_{i \in g} \Delta_i$.

Then, XORR scheduler is defined as follows:

$$\hat{g}^t = \arg\max_{g} U_g^t - \Delta_g^t, \tag{3}$$

where U_g^t is the utility to transmit the coding-set g.

Eq. (3) balances the transmission utility and the fairness constraint. It tries to select a coding-set (possible with only one frame) that maximizes the utility while has minimal service time "deficit" to ensure the fairness. A user accumulates its credit when it is not scheduled (selected in a coding-set). Following the scheduling decisions, all backlogged users update their credits as described via pseudo-code in Fig. 2.

```
1: function UpdateCredit(g)
2: for u_j ∈ g do
3:     K_j ← K_j − δ_j
4: end for
5: if Δ_g > 0 then
6:     for all u_j ∈ U do
7:         K_j ← K_j + Δ_g
8:     end for
9: end if
```

Figure 2: Pseudo-code to update credits.

After a set g is selected and the coding frame is transmitted, all members in g will decrease their credits by the fraction of service time assigned to them (Line 2-4). If any user has deficit ($\Delta_g > 0$), then all users will adjust its credits by adding Δ_g (Line 6-8). Through this, other unscheduled users may accumulate their credits and all users get non-negative credit values.

Let $\alpha_i(t_1, t_2)$ be the *allocated* service time of client u_i during time interval $[t_1, t_2)$. We show with following Theorem 3 that the scheduling discipline defined in Eq. (3) achieves bounded temporal fairness. We outline its proof in Appendix.

THEOREM 3 (TEMPORAL FAIRNESS). *Under XORR scheduler, for any two users u_i and u_j that are continuously backlogged over any interval $[t_1, t_2)$, we have*

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

where L_i^t is the frame size of u_i at time t and r_i^t is the transmission rate of u_i at time t.

Recall that, for traditional non-coding transmission scheduler, the goodput of a client is linearly decided by the service time received by the client. Thus, service time fairness implies certain goodput fairness among clients. We later will design a service-time allocation scheme such that temporal fairness in the network-coding transmission scheduler also implies goodput fairness. To the best of our knowledge, this is the first attempt to address this issue in the literature.

4.2 Reception Estimation

In XORR, the AP does not require its clients to explicitly acknowledge every native frame that they have overheard, instead the AP estimates the probability that a client has certain native frames, based on its link reliability. Note that a client may still acknowledge the reception of its *own* frame if it successfully receives one. In our analysis, we always assume that ACKs will never get lost. The AP maintains a statistic on the reliability[1] γ_i^t to each client u_i. This information is already available for most of existing wireless networks. In XORR, the AP also maintains a score-table $\mathcal{Y}$ that has $N \times N$ entries. Each entry $y_{i,j}$ records the probability for u_i to have the HOL native frame of client u_j. We then discuss in detail how the AP updates $\mathcal{Y}^{t+1}$ for current time $t+1$ from the information ($\mathcal{Y}^t$ and results of transmission) at time t.

The table initially contains all zeros. When the AP transmits an original frame p_j (and p_j is lost), it will update the column j of its score-table accordingly,

$$y_{i,j} = \gamma_i, 1 \leq i \leq N.$$

Such estimation will be updated once a frame is sent (either a native frame p_j or a coded frame that contains p_j).

If p_j is directly sent, then the probability that u_i does not have p_j after the transmission is the union of two events: u_i has no p_j before the transmission and u_i does not receive the retransmission neither. Thus, we have

$$y_{i,j}^{t+1} = 1 - (1 - y_{i,j}^t)(1 - \gamma_i), \; i \neq j. \tag{4}$$

However, when a coded frame p_g is sent, the estimation of $y_{i,j}$ is a little bit subtle. We consider two cases separately. A client is in the set g or not. For the first case that clients are not contained in the set, i.e. u_i, $i \notin g$, they may decode a native frame in g, if possible. Then, the probability that u_i ($i \notin g$) does not have p_j after the transmission is the union of two events: u_i has no p_j before the transmission and u_i fails to decode p_j. Therefore, we have

$$y_{i,j} = 1 - (1 - y_{i,j}) \cdot \left[(1 - \gamma_i) + \gamma_i (1 - \prod_{q \in g \setminus j} y_{i,q}) \right]. \tag{5}$$

The equation enclosed in the square-bracket reflexes the probability that u_i fails to decode p_j, which happens either due to fail of reception (the first term) or because it misses enough native frames to decode the coded frame (the second term).

The second case is to update $y_{k,j}$ for the clients that are in the set, i.e. u_k, $k \in g$. Certainly, 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 should have all other native frame p_q, $q \in g \setminus k$. Thus, the AP will update

$$y_{k,q}^{t+1} = 1.$$

Otherwise, u_k fails to acknowledge. It may be because either u_k fails to receive the transmission or it does not have all needed native frames to decode. Thus, the new $y_{k,j}$, $j \in g \setminus k$ is estimated based on the Bayes-law. We denote $\overline{y_{k,j}} = 1 - y_{k,j}$ and $\mathbf{Pr}\left(\overline{ACK_k^{t+1}}\right)$ the probability that u_k does not acknowledge at current 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,

$$y_{k,j}^{t+1} = 1 - \frac{1 - y_{k,j}^t}{(1 - \gamma_k) + \gamma_k (1 - \prod_{q \in g \setminus j} y_{k,q}^t)}. \tag{6}$$

When a client successfully decodes its own frame, it should immediately send an ACK to the AP. Upon receiving the ACK of p_i, the AP removes p_i from the head of queue for u_i and is ready to transmit the next frame to u_i. Since p_i will never be sent again, its estimations are removed from the score-table. The corresponding column is cleaned to zero to record the estimations of the next frame to u_i. Note that with this ACK, a client can piggyback the information of its received native frames to further facilitate XORR recovery. Then, if the AP receives an ACK that piggybacks such information, it may update the corresponding estimation to one.

4.3 Coding-set Selection

Unlike transmitting a native frame p_j, whose expected goodput is $r_j \gamma_j$, the goodput achieved by transmitting a coded frame p_g depends on how well the coded frame can be decoded by its clients.

DEFINITION 1 (DECODING ABILITY). *The decoding ability A_i^g, $i \in g$ is the probability that user u_i can decode its frame p_i from a coded frame p_g over a set g of frames.*

[1] For simplicity, we may omit the dependence on time t when no confusion occurs.

The *decoding ability* can be estimated from the score-table maintained at the AP. For a user u_i, A_i^g can be calculated using the following equation, $A_i^g = \prod_{j \in g \setminus i} y_{i,j}$, where $y_{i,j}$ is an element of the score-table that records the estimated probability that u_i has the native frame p_j.

We then derive the expected goodput (i.e., the expected number of bits, targeted to it, are received per unit of time) of a client in a coding-set g. We denote p_g as the coded frame for $p_j, j \in g$. Let L_g be the size of p_g. Then, we have $L_g = \max_{j \in g} L_j$. Let r_g be the transmission rate to send the coded frame, p_g. We have $r_g = \min_{j \in g} r_j$. Thus, to transmit frame p_g, the air time consumed is $T_g = \frac{L_g}{r_g}$. Then, for every $u_i \in g$, the expected goodput χ_i^g is

$$\chi_i^g = \frac{L_i}{T_g} \cdot \gamma_i \cdot A_i^g, \quad (7)$$

where γ_i is the reliability to u_i, and A_i^g is the decoding ability of u_i for the coded frame p_g. Then, the expected goodput of the coding-set g is

$$\chi^g = \sum_{i \in g} \chi_i^g. \quad (8)$$

From the above derivation, one can find the expected goodput of a coding-set can be low if it fails to be decoded by many members. In XORR, we only focus on the coding-set that improves the system goodput. In other words, we only select the coding-set that has higher expected goodput compared to transmission of any of its members. We call a coding-set g satisfied this condition as a *valid* coding-set. Hereafter, unless otherwise mentioned, a coding-set is referred as a *valid* coding-set. The proof of the following theorem simply follows the definition of the valid coding-set.

THEOREM 4. *Let g be a valid coding-set. $\chi^g \geq r_i \cdot \gamma_i$, $i \in g$.*

To find the optimal coding-set that has the maximal expected goodput is complex. Denote Ψ as the set of users that are waiting for retransmission. We then have the following theorem, whose proof is in Appendix.

THEOREM 5. *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, we propose a heuristic coding-set selction algorithm in Section 4.5.

4.4 Service Time Assignment

When transmitting a coded frame, the transmission time is shared among the members in the coding-set. It is critical to distribute this transmission time among them. If not properly assigned, some users may have significant performance degradation. Our goal is to design a *service time assignment* algorithm, so that for any user, it will have performance improvement with XORR (compared with non-coding). We call any service time assignment that achieves this goal as *fair service time assignment.*

We define the relative coding edge ψ_i^g of u_i in a coding-set g as the ratio of the expected goodput of u_i when being coded to that without coding. That is $\psi_i^g = \frac{\chi_i^g}{r_i \cdot \gamma_i}$. We can see that $0 < \psi_i^g \leq 1$. This is because for a user u_i, when being encoded, it may take longer time to transmit ($T_g \geq T_i$) and it may still not be able to decode even if it received the coded frame.

We propose that the distribution of the transmission time of T_g among the members of the set should be proportional to the *relative coding edge* of each member. We denote δ_j^g as the service time that is assigned to u_j, when p_j is encoded in a set g. Then we have

$$\delta_i^g = T_g \cdot \frac{\psi_i^g}{\sum_{j \in g} \psi_j^g}. \quad (9)$$

THEOREM 6. *Given a coding-set g. Define λ_j, the effective goodput of u_j served in this transmission of in a coding-set g, as*

$$\lambda_j = \frac{\chi_j^g \cdot T_g}{\delta_j}. \quad (10)$$

If the service time assignment strategy is defined as Eq. (9), for every user in a coding-set, its effective goodput should be no less than that when its native frame is being transmitted alone, i.e. $\forall u_i \in g$. $\lambda_j \geq r_i \gamma_i$.

PROOF. By definition, $\lambda_j = \frac{\chi_i^g}{\psi_i^g} \cdot \sum_{j \in g} \psi_j^g = r_i \cdot \gamma_i \cdot \sum_{j \in g} \psi_j^g$. Let $\hat{j} = \arg\max_{j \in g} r_j \cdot \gamma_j$. We have

$$\sum_{j \in g} \psi_j^g = \sum_{j \in g} \frac{\chi_j^g}{r_j \cdot \gamma_j} \leq \sum_{j \in g} \frac{\chi_j^g}{r_{\hat{j}} \cdot \gamma_{\hat{j}}} = \frac{1}{r_{\hat{j}} \cdot \gamma_{\hat{j}}} \cdot \sum_{j \in g} \chi_j^g$$

From Theorem 4, we have $\chi^g = \sum_{j \in g} \chi_j^g \geq r_{\hat{j}} \cdot \gamma_{\hat{j}}$. Therefore, $\sum_{j \in g} \psi_j^g \geq 1$. Thus, the theorem is proven. □

Next theorem (whose proof is in Appendix) shows that our scheme achieves better expected goodput. Note our scheduling achieves temporal fairness.

THEOREM 7. *Given any scheduling discipline $\mathcal{L}$ that achieves temporal fairness, denote λ_i^{XORR} and $\lambda_i^{\mathcal{L}}$ the goodput of u_i with and without XORR. If the service time assignment strategy is defined as Eq. (9),*

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

4.5 Put Everything Together

We first present the scheduling flow of XORR. And then we present a few remaining components for our NC-aware scheduling.

XORR scheduling flow. The pseudo-code of XORR operation is shown in Fig. 3. In XORR, the AP classifies the users into two groups: $TxGroup$ and $RetxGroup$. If the HOL frame of a user u_i is an original frame, u_i is put into the $TxGroup$ queue. If a frame is detected to be lost, the

AP moves its corresponding user to the $RetxGroup$ queue. A queue in $RetxGroup$ is moved back to $TxGroup$ once its HOL frame is acknowledged. The *scheduling* function takes a loop which examines the best scheduling candidates in $TxGroup$ and $RetxGroup$. In $TxGroup$, the candidate set is always one original frame; while in $RetxGroup$, a coding-set may contain multiple frames. As we shown in Theorem 5 that finding an optimal coding-set is NP-hard, we will propose a heuristic algorithm for function *Select-CodingSet* below. According to Eq. (3), the scheduler selects the best set to encode (if needed) and transmit (Line 9). The pseudo-code of function *UpdateCredit* is shown in Fig. 2.

Coding opportunity. XORR introduces a new coding opportunity to reduce the retransmission. Therefore, it is reasonable to defer the recovery of a lost frame for a while to pursue the potential coding possibility. To do so, we artificially bias for choosing an original frame to transmit by multiplying a factor on the expected goodput. More specifically, we calculate χ_g^* using following equation,

$$\chi_g^* = \begin{cases} \theta\chi_i, & \text{if } p_i \text{ is an original frame} \\ \chi_g, & \text{otherwise} \end{cases} \tag{11}$$

where $\theta > 1$ is a tunable parameter that gives bias to scheduling original frames to increase the coding opportunities. Then, the utility of the selected set is calculated using χ_g^* in Eq. (12).

Utility function. We define the utility of a coding-set g as a function of the expected goodput of g. The utility U_g is an increasing function of χ_g and is bounded, as defined in Eq. (12).

$$U_g = \beta \cdot T_{\max} \cdot (1 - e^{-\frac{\chi_g}{r_{\max}}}), \tag{12}$$

where $T_{\max} \leq \frac{\max_i L_i}{\min_i r_i}$ is the maximum transmission time of a frame, and $r_{\max} = \max_i r_i$ is the maximum possible transmission rate. Obviously, U_g is upper-bounded by βT_{max}. β is a parameter that can be tuned to balance the opportunistically improved system performance and the fairness bound [11]. This definition also has a nice property that U_g is maximized when χ_g is maximized by choosing a set g, *i.e.*, g maximizes the expected goodput of all clients.

Heuristic coding-set selection. Function *SelectCodingSet* as shown in Fig. 4 uses a greedy algorithm to select the best coding-set in $RetxGroup$. The algorithm starts to search the coding-set with only one user and find the one which has the maximal utility minus time deficit. This user is selected in the group. Then, the algorithm tries to search again in the remaining users in $RetxGroup$ and find another user to form a better coding-set. This process continues until no more such users can be found or all users in $RetxGroup$ is selected.

4.6 Discussion

XORR can be easily extended to support proportional temporal fairness, where each user may have a weight ϕ_i such that $\frac{\alpha_i}{\phi_i} = \frac{\alpha_j}{\phi_j}$ for $i \neq j$. Here α_i is the expected service time for user i. Denote $\Phi = \{\phi_1, \phi_1, \cdots, \phi_n\}$. Then, XORR

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**

Figure 3: Pseudo-code for XORR scheduling.

function SelectCodingSet(RetxGroup)
$g \leftarrow \emptyset$
repeat
  $\hat{j} \leftarrow \arg \max_{j \in RetxGroup \setminus g} U_{g \bigcup j} - \Delta_{g \bigcup j}$
  $\hat{g} \leftarrow g \bigcup \hat{j}$
  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

Figure 4: Pseudo-code for heuristic coding-set selection

scheduler is defined by

$$\hat{g}(t) = \arg\max_g U_g(t) - \Delta_g^*(t), \tag{13}$$

where $\Delta_g^*(t) = \frac{\Delta_g(t)}{\phi_g}$. Here ϕ_g is the weight of the coding-set g, which is defined as $\phi_g = \min_{i \in g} \phi_g$. Accordingly, the credit updates defined in line 7 in Fig. 2 also changes to scale with each user's weight as $K_j \leftarrow K_j + \phi_j \cdot \Delta_g^*$. The scheduling policy defined in Eq. (13) achieves the proportional temporal fairness among any two users in the network $\frac{\alpha_i}{\phi_i} = \frac{\alpha_i}{\phi_i}$. For simplicity, we omit the proof in this paper.

XORR is also applicable with two-way traffic to further reduce the retransmissions, assuming clients can overhear each other (*e.g.* WLAN). For example, when u_i retransmits an up-link frame p_i^u, it can apply XORR to recover another lost down-link frame p_j if u_i overhears the prior transmission of p_j by transmitting $p_i^u \oplus p_j$. If u_j happens to overhear p_i^u, it can decode p_j from the coded frame. Note that the AP can always decode the uplink frame as it already has all downlink frames already. Scheduling two-way traffic follows the same scheduling policy as presented. Due to the space limitation, we omit the details.

5. PERFORMANCE EVALUATION

5.1 Simulation setup

In our simulations, we generate network topologies consisting of an AP and a varying number of users. The transmission rates of each link can be 1, 2, 5.5 and 11 Mbps, as specified by IEEE 802.11b. In all simulations, the size of the data frame is 1500 bytes. The ACK and feedback frames are transmitted with the base rate (2 Mbps) with the size of 50 bytes. We first present the results under static wireless channel, where the channel condition does not change over time. Then, we evaluate the XORR performance under more realistic time-varying channels.

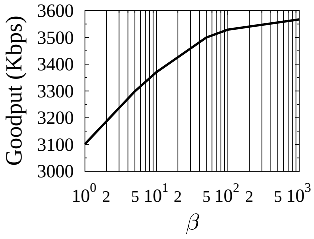
In our simulation, we compare the performance difference with four schemes: 1) *XORR*. 2) *Opportunistic scheduling (labeled Opp)*. It takes the same scheduling strategy as XORR, except each time only one native frame is allowed to be selected to transmit. 3) *IEEE 802.11-based WLAN*. This is a base-line for existing WLAN, where a shared FIFO queue is used for all users and retransmission immediately happens once a loss is detected. 4) *ER*. This is a prior NC-aided MAC-layer retransmission scheme [13]. Unlike XORR, ER does not adopt the opportunistic scheduling with temporal fairness constraint. Furthermore, ER relies on the receiver to send feedback for obtaining the reception status. We implement their *sort-by-time* coding algorithm and use 25 as the threshold for the retransmission queue. Unless otherwise mentioned, the default number of users is 10 and the transmission rate is 5.5Mbps.

We use *goodput*, *reduced retransmission ratio* and *goodput gain* to qualify the performance of different retransmission schemes. Goodput is defined as the total data successfully transmitted over time. The reduced retransmission ratio is defined as the difference of the retransmission rate [2] of target retransmission scheme and that of *802.11* divided by that of *802.11*. The goodput gain is defined as the difference of the goodput of target retransmission scheme and that of *802.11* divided by that of *802.11*.

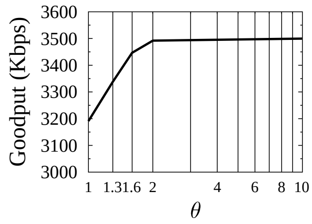
5.2 Impact of parameters

XORR has two tunable parameters, β and θ. β is used to balance the fairness bound and the system performance gain [11]; while θ decides how much priority should be given to original frames than retransmissions. Fig. 5(a) shows the goodput of XORR with respect to different β. All users are transmitted using 5.5Mbps and the reliability is randomly chosen from 0.4 to 0.6. As expected, the network goodput is increased with β. With a larger β, the scheduler trades more fairness for performance by assigning more service time to better users. Such tradeoff has been sufficiently discussed in literature [11]. In the following evaluation, we choose $\beta = 50$, which achieves good balance in our scenario. Fig. 5(b) depicts the goodput of XORR with the increase of θ. A larger θ gives the AP more favor to transmit original frames and defer the retransmission for potential coding opportunity. As shown in Fig. 5, a small θ would be enough for coding opportunity. In the following evaluation, we set $\theta = 2$.

[2] The retransmission rate is defined as the ratio of total number of retransmissions to that of transmissions.



(a) Impact of β



(b) Impact of θ

Figure 5: (a) Goodput with different values of β. $\theta = 2$(b) Goodput with different values of θ. $\beta = 50$.

5.3 Impact of heuristic selection

We have proven that finding the optimal coding-set is NP-hard in Theorem 5. Therefore, a practical heuristic coding-set selection algorithm is proposed in section 4.5. In order to verify the effectiveness of our heuristic algorithm, we compare it with an exhaustive search algorithm, which is guaranteed to give an optimal solution but computationally very expensive. As shown in Fig. 6, our heuristic algorithm is efficient because it only slightly degrades the performance but reduces drastically the complexity of the exhaustive search.

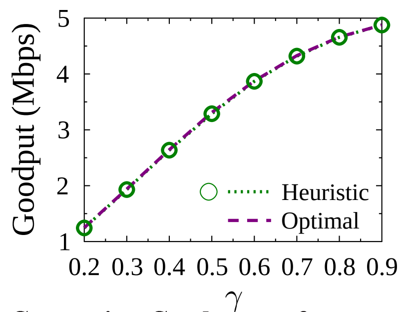


Figure 6: Comparing Goodput performance with various reliability between heuristic and exhaustive coding-set seleciton.

5.4 Static channel

(a) Impact of link reliability XORR improves the network throughput by effectively reducing the retransmissions. Therefore, in the first study, we evaluate XORR's performance under different link reliability scenarios. We let all users have the same reliability, which varies from 0.9 (most reliable) to 0.1 (most unreliable). Fig. 7 and Fig. 8 show the goodput gain and reduced retransmission ratio over *802.11* with different link reliability for XORR, *Opp*, and ER with the report period of 10, 20, 50 and 200 ms. We make the following observation:

1) As shown in Fig. 7, in the homogeneous case, *802.11* has the same goodput as *Opp*. This is reasonable since all users have exactly same transmission rate and reliability and thus no multi-user diversity gain can be utilized.

2) XORR and ER take advantage of network coding to reduce the retransmissions, as demonstrated in Fig. 8. When the link reliability is larger than 80%, over 60% of retrans-

missions are saved. When the link reliability is low, the percentage of saved retransmission by the coding schemes is reduced. This is because when the link reliability is low, each station will have less native frame received, which results in less coding opportunities.

3) With the decrease of the reliability, more frames are lost. XORR thus increasingly improves the network goodput by reducing more retransmissions as depicted in Fig. 7. When the reliability is around 0.5, the goodput gain of XORR is near 25%. However, when the reliability further decreases, the improvement bends down. This is because XORR relies on reception estimation to select coding-set when retransmitting. When there are significant losses, the accuracy of the estimation decreases due to less feedback piggybacked in ACKs.

4) The coding efficiency of ER heavily depends on the feedback information carried by the reception report. Thus ER with shorter period of reception report can reduce more retransmissions as shown in Fig. 8. However, when considering the signaling overhead, frequently sending report degrades the goodput performance severely. In Fig. 7, the goodput of ER with the period of 10 ms is even worse than that of 802.11 when the link reliability is high. Furthermore, the period is difficult to adjust because the optimal period depends on the data transmission rate and link quality, which are normally time-varying. In contrast to ER, XORR estimates the reception status without signaling overhead and thus outperforms ER.

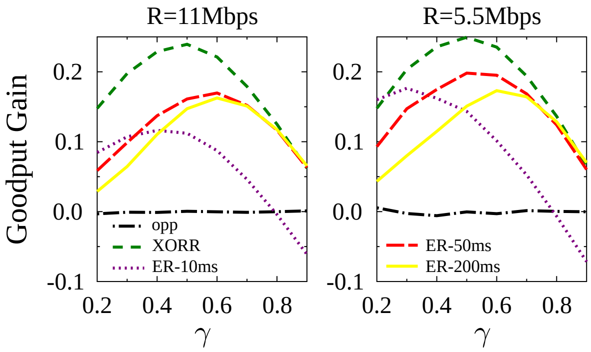


Figure 7: Goodput gain with various link reliability in static channels.

(b) Impact of the number of users

Fig. 9 and 10 demonstrate the goodput gain and reduced retransmission ratio over *802.11* with different number of users for XORR and ER with the report period of 50 ms, when the link reliability is 0.2, 0.5 and 0.8, respectively. As the number of users increases, the reduced retransmission ratio also increases in XORR and ER. This is a predictable behavior since the coding opportunity increases as the number of the users increases. However, the slope of the increase quickly slows down with only moderate number of users (*e.g.* 10). This actually suggests the network coding opportunity is already significant when there are only moderate number of users and further increasing the users does

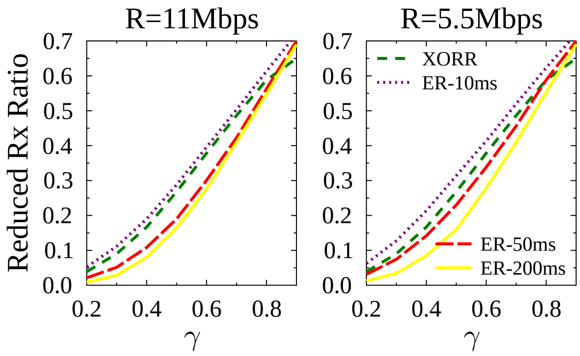


Figure 8: Reduced retransmission ratio with various link reliability in static channels.

not increase the coding opportunity greatly. Since reducing retransmissions improves the goodput, the goodput gain of XORR also increases as the number of users increases, as shown in Fig. 9. However, the goodput gain of ER bends down when the number of users is greater than 10. This is due to the overhead for sending reports. More users introduce more feedback frames, which overwhelms the coding gain in ER.

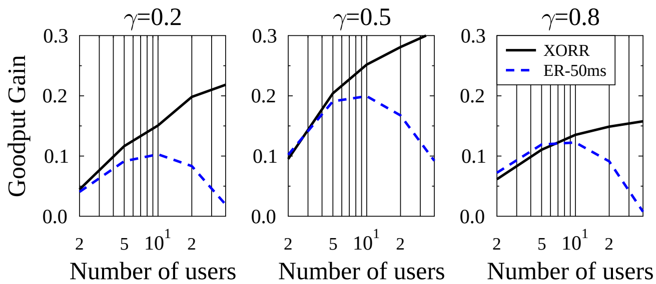


Figure 9: Goodput gain with different number of users in static channels.

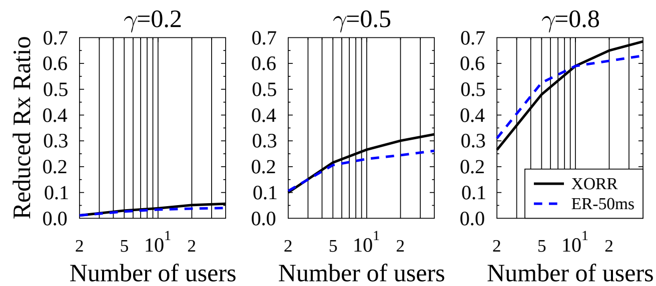
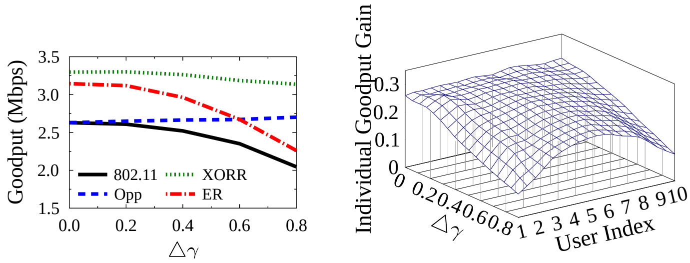


Figure 10: Reduced retransmission ratio with different number of users in static channels.

(c) Heterogeneous wireless links

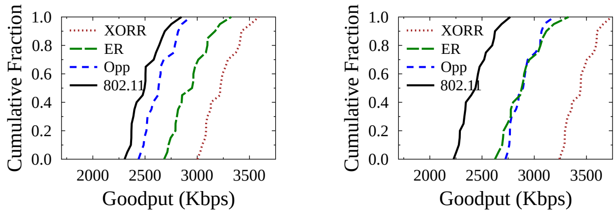
We evaluate XORR with more general scenario where users may have heterogeneous reliability. To realistically model channel condition is beyond the scope of the paper. In this simulation, we randomly choose user's reliability from $[\gamma_{min}, \gamma_{max}]$.

Fig. 11(a) compares the goodput of four schemes. In this



(a) (b)

Figure 11: (a) Goodput in heterogeneous and static channels, where $E(\gamma) = 0.5$. (b) Individual goodput gain.



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

Figure 12: Goodput in time-varying channel with static and homogenous transmission rates. $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$. Coherence time is 24.57 ms. $\bar{r} = 5.5Mbps$. $\sigma_r = 0$.

simulation, we fix the mean of the link reliability as $E(\gamma) = 0.5$, while change $\Delta\gamma = \gamma_{max} - \gamma_{min}$ from 0 to 0.8. When different users have different channel conditions, *Opp* yields better system performance than *802.11*. When $\Delta\gamma$ is large, the system goodput of *802.11* dramatically decreases. This is because *Opp* always allocates equal service time to all users. Its total throughput remains unchanged with the varying $\Delta\gamma$. However, *802.11*, which maintains goodput-based fairness, allocates more channel time to users with worse channel conditions [16].

ER also adopts goodput-based fairness as *802.11* does. Therefore, its system goodput performance also decreases as $\Delta\gamma$ increases. On the other hand, XORR maintains temporal fairness as *Opp* does. In all cases, XORR improves the overall system goodput as it effectively suppresses retransmissions. The goodput of XORR drops slightly with the increase of $\Delta\gamma$. This is because when $\Delta\gamma$ is large, most of retransmissions are targeted to low reliability users. Hence, the effective coding opportunity is reduced. Fig. 11(b) gives a close view of goodput gain of each individual user with XORR in the same simulation case. The user index is sorted with their link reliability. We can see XORR indeed improves the goodput of all users (from 10% to 25%) compared to *Opp*.

5.5 Time-Varying Channel

In practice, wireless channel is time-varying. The speed of the channel condition change is generally characterized with the *channel coherence time*, within which the channel may be considered as "static" [6]. The coherence time is related to the mobility of users or surrounding environment. We assume the channel is stationary. Therefore, we use ($\bar{\gamma}_i$, $\sigma_{\gamma,i}$) and ($\bar{r}_i$, $\sigma_{r,i}$)[3] to characterize the time-varying reliability and transmission rate of link l_i, respectively. In our simulation, the mean of the reliability of each user is randomly chosen from $[0.3, 0.7]$. The coherence time is 24.45 ms, corresponding to a fast walking speed (5 m/s) [14]. The transmission rate is fixed to 5.5 Mbps.

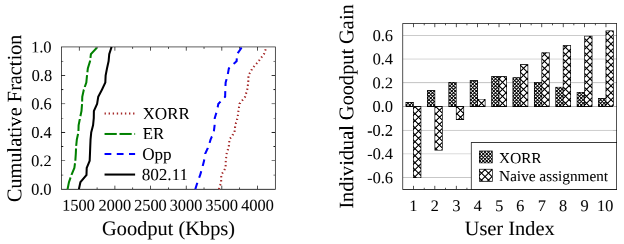
Fig. 12 plots the Cumulative Distribution Function (CDF) of the network goodput when $\sigma_\gamma = 0.1$ (large variance) and $\sigma_\gamma = 0.01$ (small variance), respectively. In Fig. 12(a), *Opp* has slight improvement compared to *802.11*. This is because the channel variance is very small, so that little *multi-user diversity* can be exploited. In contrast, when the channel varies largely, as shown in Fig. 12(b), *Opp* scheduling improves the system throughput greatly by opportunistically scheduling users with better channel condition. Since ER does not use opportunistic scheduling to exploit multi-user diversity, the goodput of ER is not improved when the channel varies largely. Nevertheless, XORR provides not only coding-gain, but also *multi-user diversity*. As a consequence, XORR outperforms ER and traditional opportunistic scheduling about $10-25\%$ and $20-25\%$, respectively. Moreover XORR has around $30-40\%$ performance gain compared to *802.11*.

As shown in Fig. 13(a), we further demonstrate the goodput performance when multiple transmission rates are used and the transmission rate of user varies over time. Each user has a random mean of the transmission rate among 2, 5.5 and 11Mbps, and the transmission rate varies between one level around the mean value. In this case, *Opp* achieves higher performance gain as the channel condition varies very large in terms of both reliability and the transmission rate. On the other hand, ER performs worse than *802.11*. As illustrated in [3], the coding gain is deducted because of the inappropriate coding scheduling which does not consider the link condition. Thus ER not only cannot exploit the gain of *multi-user diversity*, but also loses the coding gain. However, XORR again, by exploiting both multi-user diversity and the network-coding, outperforms all the other schemes.

5.6 Service time assignment

As aforementioned, when transmitting an encoded frame, the service time assignment is critical. A naive assignment will cause significant unfairness and some users would have much worse performance with network-coding compared to that without coding. With simulation, we validate that XORR not only maintains the fairness but also improves the performance of each user. We randomly choose the link reliability for each wireless link from $[0.2, 0.9]$. Fig. 13(b) shows the individual goodput gain of each user over *Opp* for XORR and XORR with a naive service time assignment, which evenly distributes the transmission time among the

[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) time-varying channel (b) service-time assignment

Figure 13: (a) Goodput in time-varying channel with time-varying transmission rates. Each user has a random mean transmission rate among 2, 5.5 and 11 Mbps. $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$. The coherence time is 24.57 ms. The number of users is 12. (b) Starvation of XORR with naive service time assignment.

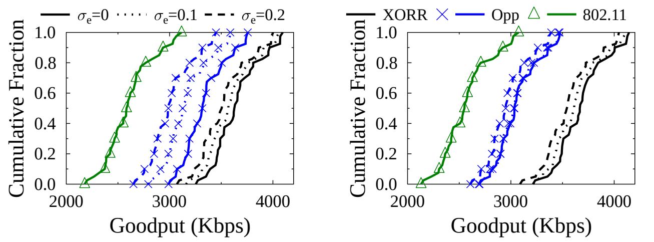
coded users. We sort the user with their link reliability. Clearly, the naive service time assignment causes significant performance degradation for less reliable users with network coding, although the overall system performance is improved. However, XORR ensures the performance improvement for every individual user.

5.7 Impact of estimation error

XORR relies on reception estimation to select coding-sets. To estimate the reception of native frame for each user, the AP further needs to estimate the link reliability. Many existing wireless system already maintains such statistics (*e.g.* WLAN [17]). We now evaluate XORR if reliability estimation contains error. 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 normal distribution $\mathcal{N}(0, \sigma_e)$. Fig. 14 summarizes CDF of network goodput under two different coherence times, 6.14 (vehicle speed) and 24.57ms (walk speed), respectively. When there presents 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 is not varying very fast. It is interesting to note that XORR is less sensitive to estimation error compared to *Opp* as shown in Fig. 14(a). It may be because the network coding actually could average this error out. Therefore, it is less significant a user misses the transmission due to estimation error, as it may get a coding opportunity later in the waiting queue.

5.8 Impact on delay

In XORR, the scheduler may defer the retransmission for potential coding opportunity. This might cause additional delay in frame delivery. We define the *frame transmission delay* (FTD) as the interval between a frame coming to the head of the queue and the time when it is successfully received. Fig. 15 plots CDF of frame delay measured with different scheduling policy. *802.11* has relative long FTD as the AP will continue retransmitting a lost frame even the channel is bad, so that it causes HOL blocking. In contrast,



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

Figure 14: Goodput with estimation error in time-varying channel model where, $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$.

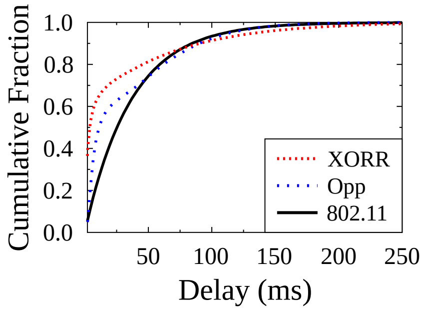


Figure 15: CDF of frame transmission delay in time-varying channel model where, $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.4, 0.8]$. Coherence time is 24.57 ms.

Opp does not have this HOL blocking issue as the scheduler always tries to select the user in a good channel condition irrespective of the retransmission states. Therefore, the curve of *Opp* has a shift to the left. A large portion of frames in XORR has an even shorter FTD, *i.e.* 70% of frames has FTD less than 5ms. This is because XORR significantly reduces the number of retransmissions. Note that XORR has slight longer tail compared to *Opp*. This is because XORR favors transmissions on original frames and such induces more delay for retransmission.

6. TEST-BED EXPERIMENTS

We have prototyped XORR and preliminarily evaluated its performance on real wireless test-bed. Our implementation is based on Atheros AR5212 wireless NIC in 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 to 11Mbps. The links between the AP and stations have an average reliability of 80%. Table 3 shows a summary of goodput gain in 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

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

Table 3: Goodput improvement in test-bed experiments. XX/YY means the goodput improvement of XX over YY.

Opp. The coding gain XORR obtained over *Opp* is 8.0% with UDP flows and 14.5% with TCP flows. 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 the frame losses, it improves TCP performance more significantly.

7. CONCLUSION

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

8. 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] 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.

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

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

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

[6] A. Goldsmith. *Wireless Communication*. Cambridge University Press, 2005.

[7] J. Hastad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, pages 105–142, 1999.

[8] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram. Downlink scheduling in cdma data networks. In *ACM MobiCom*, pages 179–190, 2000.

[9] 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.

[10] X. Liu, E. K. P. Chong, and N. B. Shroff. Opportunistic transmission scheduling with resource-sharing constraints in wireless networks. *IEEE JSAC*, 19, october 2001.

[11] 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.

[12] P. McKinley, C. Tang, and A. Mani. A study of adaptive forward error correction for wireless collaborative computing. *IEEE Trans. on Parallel and Distributed Systems*, pages 936–947, 2002.

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

[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] S. Sengupta, S. Rayanchu, and S. Banerjee. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *IEEE Infocom*, 2007.

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

[17] 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.

[18] 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/.

[19] H. Yomo and P. Popovski. Opportunistic Scheduling for Wireless Network Coding. In *IEEE ICC*, pages 5610–5615, 2007.

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

[21] K. Jamieson and H. Balakrishnan. PPR: Partial packet recovery for wireless networks. In *ACM SIGCOMM*, 2007.

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

[23] G. Woo, P. Kheradpour, and D. Katabi. Beyond the bits: Cooperative packet recovery using phy information. In *ACM MobiCom*, Montreal, Canada, 2007.

APPENDIX

A. Proof of Theorem 1

PROOF. After sending a coded frame with average coding size $\mathbf{K}$, the average number of ACKs is: $\alpha_r = \gamma \cdot \mathbf{K}$. Since the ACKed frames are removed from retransmission queue, there are α_r vacancies in the retransmission queue. Consequently, the AP sends original frames for α_r users individually until the average retransmission queue reaches $\mathfrak{N}$. Thus the average number of sent original frames for α_r users is $X = \alpha_r \times \sum_{i=1}^{\infty} \gamma^{i-1}(1-\gamma) \cdot i = \frac{\gamma \mathbf{K}}{1-\gamma}$. Since α_r frames are not ACKed after sending X original frames, it can be inferred that the number of consequent ACKs is $\alpha_n = X - \alpha_r = \frac{\gamma^2 \mathbf{K}}{1-\gamma}$. In summary, the total transmissions in one retransmission period include one retransmission and X original frame transmissions, i.e., $1 + X$. And the total ACKs should be $\alpha_r + \alpha_n$. Therefore the expected goodput with XORR is $\lambda^{XORR} = \frac{1+X}{\alpha_r+\alpha_n} \cdot r = \frac{\gamma \mathbf{K}}{1-\gamma+\gamma \mathbf{K}} \cdot r$. So the coding gain of XORR is $B = \frac{\mathbf{K}}{1-\gamma+\gamma \mathbf{K}}$. □

B. Proof of Lemma 2

PROOF. Assume $\hat{g}$ is the selected decodable set. Then the average size of the set is $\mathbf{K} = \sum_{\kappa=1}^{\mathfrak{N}} \kappa \cdot Pr(|\hat{g}| = \kappa) = \sum_{\kappa=1}^{\mathfrak{N}} Pr(|\hat{g}| \geq \kappa)$.

Assume g_i^κ is a decodable set with κ users, where $|g_i^\kappa| = \kappa$. On the other hand, $\overline{g_i^\kappa}$ is an undecodable set with κ users. Note that g_i^κ may be part of larger set with more than κ users. Accordingly, it can be inferred that $Pr(\overline{g_i^\kappa}) = 1 - Pr(g_i^\kappa) = 1 - \gamma^{(\kappa-1)\kappa}$. Thus, we have $Pr(|\hat{g}| \geq \kappa) = Pr\left(\bigcup_{i=1}^{m} g_i^\kappa\right) = 1 - Pr\left(\bigcap_{i=1}^{m} \overline{g_i^\kappa}\right)$, where m is total number of sets with κ users, $m = \binom{\mathfrak{N}}{\kappa}$. Since there are overlaps among undecodable sets, the joint probability can be bounded as $Pr\left(\bigcap_{i=1}^{m}\overline{g_i^\kappa}\right) \geq \prod_{i=1}^{m} Pr(\overline{g_i^\kappa}) = (1-\gamma^{(\kappa-1)\kappa})^m$. Thus the upper bound is proven.

Assume there is an inefficient XORR scheme that the AP groups $\mathfrak{N}$ users into several coding-sets so that there is no overlap among the grouped sets, i.e. there are $\lfloor \frac{\mathfrak{N}}{\kappa} \rfloor$ sets, and κ users in each set, where $\kappa = 1, 2, \cdots, \mathfrak{N}$. the AP later selects a decodable set among those un-overlapped sets as a coding set. It can be inferred that $\mathbf{K} \geq \mathbf{K_1}$, where $\mathbf{K_1}$ is the expected size of the coding-set in the inefficient XORR scheme. Since there is no overlap among sets in the inefficient scheme, the joint probability is $Pr\left(\bigcap_{i=1}^{m'}\overline{g_i^\kappa}\right) = (1-\gamma^{(\kappa-1)\kappa})^{m'}$, where $m' = \lfloor \frac{\mathfrak{N}}{\kappa} \rfloor$. Therefore, the lower bound is proven. □

C. Proofs of Theorem 3

LEMMA 8. *For any flow i and for any schedule time t, the credit counter value is always bounded as*

$$0 \leq K_i \leq \max_t \frac{L_i^t}{r_i^t} + U_{max},$$

where U_{max} is the maximal value of utility function.

PROOF. According to the credit update in Figure 2, $K_i \geq 0$.

For the right part of the inequality, the proof is separated into two cases according to Δ_g^t is greater to zero or not. When $\Delta_g^t > 0$ and $i \notin g^t$, then from Eq. (3) we have $U_i^t - (T_i^t - K_i^t) \leq U_g^t - \Delta_g^t$. According to the credit update, we have $K_i^{t+1} = K_i^t + \Delta_g^t \leq T_i^t + U_g^t - U_i^t \leq \max_t \frac{L_i^t}{r_i^t} + U_{max}$. Similarly, when $\Delta_g^t > 0$ and $i \in g^t$, we have $K_i^{t+1} = K_i^t - \delta_i^t + \Delta_g^t \leq \max_t \frac{L_i^t}{r_i^t} + U_{max}$. On the other hand, when $\Delta_g^t \leq 0$ and $\forall i$, the updated credit is $K_i^{t+1} \leq K_i^t \leq \max_t \frac{L_i^t}{r_i^t} + U_{max}$. Considering both cases, Lemma 8 holds. □

Proof of Theorem 3

PROOF. It has been shown in [11] that, for any flow i continuously backlogged during $[t_1, t_2)$, $K_i^{t_1} - K_i^{t_2} = \alpha(t_1, t_2) - \sum_{t=t_1}^{t_2-1} \max\left(0, \Delta_g^t\right)$. Accordingly, the service discrepancy can be derived as following

$$\begin{aligned} |\alpha_i(t_1,t_2) - \alpha_j(t_1,t_2)| &= \left|K_i^{t_1} - K_i^{t_2} - (K_j^{t_1} - K_j^{t_2})\right| \\ \leq \quad & \left|\max\left(K_i^{t_1}, K_i^{t_2}\right)\right| + \left|\max\left(K_j^{t_1}, K_j^{t_2}\right)\right| \\ \leq \quad & \max_t \frac{L_i^t}{r_i^t} + \max_t \frac{L_j^t}{r_j^t} + 2U_{max} \end{aligned}$$

Thus Theorem 3 is proven. □

D. Proofs of Theorem 7

Assume our system is stationary process and γ_i^t, r_i^t and L_i^t are independent random variables. In order to prove Theorem 7, we first present the supporting lemmas.

LEMMA 9. *Given any two scheduling disciplines $\mathcal{L}$ and $\mathcal{N}$ that achieve temporal fairness, where the service discrepancy in any time interval is bounded by $\theta^{\mathcal{L}}$ and $\theta^{\mathcal{N}}$, respectively. For any group of users that is continuously backlogged over interval (t_1, t_2), we have*

$$\alpha_i^{\mathcal{N}}(t_1,t_2) - (\theta^{\mathcal{L}} + \theta^{\mathcal{N}}) \leq \alpha_i^{\mathcal{L}}(t_1,t_2) \leq \alpha_i^{\mathcal{N}}(t_1,t_2) + (\theta^{\mathcal{L}} + \theta^{\mathcal{N}}),$$

where $\alpha_i^{\mathcal{L}}(t_1,t_2)$ and $\alpha_i^{\mathcal{N}}(t_1,t_2)$ are the service time for any user i in the group, respectively.

PROOF. Assume a group of users, U, is continuously backlogged in (t_1, t_2). Thus for any two users i and j in the group U, we have $\frac{t_2-t_1}{|U|} - \theta^{\mathcal{L}} \leq \alpha_i^{\mathcal{L}}(t_1,t_2) \leq \frac{t_2-t_1}{|U|} + \theta^{\mathcal{L}}$. This can be proven as follow: if $\exists \alpha_i^{\mathcal{L}}(t_1,t_2) > \frac{t_2-t_1}{|U|} + \theta^{\mathcal{L}}$, then $\forall j, \alpha_j^{\mathcal{L}}(t_1,t_2) \geq \alpha_i^{\mathcal{L}}(t_1,t_2) - \theta^{\mathcal{L}} = \frac{t_2-t_1}{|U|}$. Accordingly, $\sum_{i\in U} \alpha_i^{\mathcal{L}}(t_1,t_2) > t_2 - t_1$, which contradicts to our assumption. We can have similar bound for for the fair scheduler $\mathcal{N}$. Combining both bounds, the lemma is proven. □

LEMMA 10. *Given two scheduling disciplines $\mathcal{L}$ and $\mathcal{N}$ that achieve temporal fairness, with and without network coding. If in each scheduling time the coding effective goodput can satisfy*

$$\frac{\chi_j^g T_g}{\delta_i} \geq \gamma_i r_i, \tag{14}$$

then the expected goodput of u_i in (t_1, t_2) is

$$E[\lambda_i^{\mathcal{L}}(t_1,t_2)] \geq E[\lambda_i^{\mathcal{N}}(t_1,t_2)] - \epsilon,$$

where $\epsilon = \frac{E[\gamma_i]E[r_i](\theta^{\mathcal{L}}+\theta^{\mathcal{N}})}{t_2-t_1}$ and $\theta^{\mathcal{L}}$ and $\theta^{\mathcal{N}}$ are the fairness bounds for the scheduling disciplines $\mathcal{L}$ and $\mathcal{N}$, respectively.

PROOF. Let $Q_i^{\mathcal{L}}$ and $Q_i^{\mathcal{N}}$ be the set of scheduling time for u_i in (t_1, t_2) in the scheduler with and without coding, respectively. Then, $E[\lambda_i^{\mathcal{L}}(t_1,t_2)] = \frac{\sum_{t\in Q_i^{\mathcal{L}}} E[\chi_i^g(t)\cdot T_g^t]}{t_2-t_1} = \frac{E[L_i]E[\gamma_i]\sum_{t\in Q_i^{\mathcal{L}}} E[A_i^g(t)]}{t_2-t_1}$. In addition, $E[\lambda_i^{\mathcal{N}}(t_1,t_2)] = \frac{\sum_{t\in Q_i^{\mathcal{N}}} E[\gamma_i^t L_i^t]}{t_2-t_1}$. On the other hand, based on Eq. (14), we have $\alpha_i^{\mathcal{L}}(t_1,t_2) = E\left[\sum_{t\in Q_i^{\mathcal{L}}} \delta_i^t\right] \leq \frac{E[L_i]}{E[r_i]} \times \sum_{t\in Q_i^{\mathcal{L}}} E[A_i^g(t)]$. By applying Lemma 9 to the above equation, we get $\frac{E[L_i]}{E[r_i]}\sum_{t\in Q_i^{\mathcal{L}}} E[A_i^g(t)] \geq \alpha_i^{\mathcal{N}}(t_1,t_2) - (\theta^{\mathcal{L}}+\theta^{\mathcal{N}})$. Since $\alpha_i^{\mathcal{N}}(t_1,t_2) = \sum_{t\in Q_i^{\mathcal{N}}} E\left[\frac{L_i^t}{r_i^t}\right]$, we have

$$E[L_i]E[\gamma_i]\sum_{t\in Q_i^{\mathcal{L}}} E[A_i^g(t)] \geq \sum_{t\in Q_i^{\mathcal{N}}} E[\gamma_i^t L_i^t] - E[\gamma_i]E[r_i](\theta^{\mathcal{L}}+\theta^{\mathcal{N}}).$$

Hence, the lemma is proven. □

Proof of Theorem 7

PROOF. Since XORR coding scheme adopts the service time assignment in Eq. (9), Theorem 6 can satisfy the assumption of Lemma 10. Consequently, the results of Lemma 10 can be applied to XORR coding scheme. Therefore, Theorem 7 is proven. □

E. Proof of Theorem 5

PROOF. We will reduce the NP-complete clique problem to the problem of finding optimal coding-set maximalizing the expected goodput. For a graph $G = (V, E)$, we define a coding-set selection problem as follows. The set of clients is V. Let $\gamma_i = 1$, $L_i = 1$ and $r_i = 1$ for each $i \in V$. Furthermore, at current time t, a client i has the packet p_j (i.e., $y_{i,j} = 1$) iff edge $(v_i, v_j) \in E$. It is easy to showt that finding an optimal coding-set in such setting is equivalent to solving the maximum clique problem in G. In addition, since maximum clique is not approximable within $O(|V|^{1-\epsilon})$ for any $\epsilon > 0$ unless NP=ZPP [7], coding-set selection problem is also not approximable within $O(|\Psi|^{1-\epsilon})$. □