GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

Exploiting Network Coding in Lossy Wireless Networks

vorgelegt von

Fang-Chun Kuo

aus

Pingtung, Taiwan

Dissertation

zur Erlangung des Doktorgrades

der Mathematisch-Naturwissenschaftlichen Fakultäten

Juni 2009

Referent: Prof. Xiaoming Fu

Korreferenten: Prof. Dieter Hogrefe, Prof. Jon Crowcroft

Tag der mündlichen Prüfung: 16. Februar 2009

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

ABSTRACT

MATHEMATISCH-NATURWISSENSCHAFTLICHEN FAKULTÄTEN

Doctor of Philosophy

by Fang-Chun Kuo

Users increasingly depend on Wireless LANs (WLANs) for business and entertainment. It is well-known that wireless links are error-prone and require retransmissions to recover from errors and packet losses. Previous work proposed to use network coding (NC) for more efficient MAC-layer retransmissions in WLANs. However, their design is independent from underlying physical layer and MAC capabilities and algorithms. These independent design may result in inefficient utilization of network coding gain, or even impair system performance.

This dissertation presents a practical network coding-aided MAC layer retransmission scheme, namely XOR Rescue (XORR). Unlike previous independent network coding design, XORR provides a global approach by integrating the utilization of network coding, adaptation to time-varying wireless channel, fairness, and multi-rate transmission in wireless networks. The main characteristic of XORR is *opportunism*: each node relies on local information to detect the best transmission/retransmission and exploits the benefits provided by both network coding and multiuser diversity whenever they arise.

The contributions of this dissertation are multifold. First, it builds a practical link layer retransmission architecture by integrating network coding and wireless physical and MAC design. Specifically, the system presented in this dissertation is the first to accommodate network coding into complex wireless environments, e.g. time-varying link quality. Second, the work presents novel algorithms and introduces new concepts which may be applicable to other wireless network coding protocols. A Bayesian-learning based estimation scheme for evaluating reception status can providing substantially coding opportunities without extra overheads. A framework of an network coding aware fair opportunistic scheduling is designed with the objective of maximizing the system goodput as well as maintaining fairness.A new coding metric, namely expected goodput, is devised for exploiting the

gain of network coding and multiuser diversity. The concept of network coding fairness is proposed, where not only the fair resource share is guarantee but also the performance for *every* wireless station is improved compared to non-coding scheme. Finally, we present theoretical analysis and extensive simulations. Our results show that XORR outperforms the non-coding fair opportunistic scheduling and 802.11 by 25% and 40%, respectively.

GEORG-AUGUST-UNIVERSITÄT GÖTTINGEN

ZUSAMMENFASSUNG

MATHEMATISCH-NATURWISSENSCHAFTLICHEN FAKULTÄTEN

Doctor rerum naturalium

von Fang-Chun Kuo

Benutzer sind für Geschäfts- und Unterhaltungsanwendungen zunehmend abhängig von Wireless LANs (WLANs). Es ist bekannt, dass drahtlose Verbindungen anfällig für Fehler sind und erneute Ubertragungen benötigen um Fehler und Paketverluste auszugleichen. In früheren Arbeiten wurde vorgeschlagen, Netzwerk-Codierung (NC) zu verwenden, um erneute MAC-Layer Übertragungen in WLANs effizienter zu gestalten. Allerdings ist das vorgeschlagene Design unabhängig von den unterliegenden physikalischen Schichten und den Fähigkeiten und Algorithmen der Medienzugriffskontrolle (MAC). Dieses unabhängige Design kann zu einer ineffizienten Verwendung des Zugewinns durch Netzwerk-Codierung führen oder sogar die Systemleistung beeinträchtigen.

Diese Dissertation präsentiert ein praktisches Schema für wiederholte, durch Netzwerk-Codierung unterstützte, MAC-Layer Übertragungen, das sogenannte XOR Rescue (XORR). Anders als vorhergehende, unabhängige Netzwerk-Codierungs-Konzeptionen bietet XORR einen umfassenden Ansatz durch die Integration der Anwendung von Netzwerk-Codierung, der Anpassung an den zeit-veränderlichen drahtlosen Kanal, Fairness und Mehrfach-Geschwindigkeits-Übertragung in drahtlosen Netzwerken. Das Hauptmerkmal von XORR ist Opportunismus: Jeder Knoten stützt sich auf lokale Informationen um die beste Übertragung bzw. erneute Übertragung zu erkennen und nutzt die Vorteile, die Netzwerk-Codierung und Mehrbenutzer-Diversität bieten, wann immer sie entstehen.

Die Beiträge dieser Dissertation sind vielfältig. Erstens wird eine praktische Architektur für erneute Übertragungen der Verbindungsschicht konstruiert. Dafür werden Netzwerk-Codierung und das Design der drahtlosen physikalischen und Medienzugriffskontrolle integriert. Insbesondere ist das in dieser Dissertation vorgestellte Verfahren das erste welches Netzwerk-Codierung in komplexe drahtlose Umgebungen aufnimmt. Zum Beispiel Umgebungen mit einer über die Zeit

wechselnden Verbindungsqualität. Zweitens stellt diese Arbeit neue Algorithmen und Konzepte vor, die in anderen Netzwerk-Codierungsprotokollen angewandt werden können. Ein Schema für die Beurteilung des Empfangszustandes, das auf Bayeschen-Lernen basiert, kann wesentliche Kodierungsmöglichkeiten ohne zusätzlichen Overhead bieten. Ein Framework für faires opportunistisches Scheduling, welches Network-Kodierung nutzt, wird entworfen mit dem Ziel den Goodput zu maximieren und Fairness zu gewährleisten. Eine neue Kodierungsmetrik, nämlich der erwartete Goodput, wird entworfen um den Zugewinn durch Netzwerk-Codierung und Mehrbenutzer-Diversität auszunutzen. Das Konzept der Fairness von Netzwerk-Codierung wird vorgeschlagen. Dabei wird nicht nur das faire Teilen von Ressourcen garantiert sondern auch die Leistung für jede drahtlose Station verbessert im Vergleich zu Systemen, die keine Kodierung verwenden. Schliesslich präsentieren wir eine theoretische Analyse und ausgiebige Simulationen. Unsere Ergebnisse zeigen, dass XORR nicht-codierendes, faires, opportunistisches Scheduling sowie 802.11 mit 25% bzw. 40% übertrifft.

To Chien-Ho Kuo and Su-Lien Wu

Acknowledgements

I would like to show my sincerest gratitude to my co-supervisors, Prof. Xiaoming Fu and Prof. Dieter Hogrefe for giving me copious amounts of insightful guidance, constant encouragement, constructive criticism, and expertise on every subject that arose throughout all these years. Their enthusiasm and dedication to their students are truly inspiring; it is my very privilege to have been one of them. I would also like to thank Prof. Jon Crowcroft for being on my advisory committee and for all his helps and suggestions during my PhD study. I would like to thank Prof. Carsten Damm, Prof. Bernhard Neumair and Prof. Stephan Waack for serving on my thesis committee and their invaluable suggestions.

I first got the opportunity to explore the topic of wireless network coding when I was an intern at the Microsoft Research, Beijing under the supervision of Dr. Kun Tan and Prof. Xiang-Yang Li in the summer of 2007. I am very grateful to them for introducing me to this exciting topic and for the enjoyable collaborations. I thank the researcher in Microsoft, Jiansong Zhang,for his implementation and testbed setup of XORR. The experimental results presented in this thesis are based on his work. I also would like to thank my other colleagues in Microsoft, Dr. Chuanxiong Guo, Yunxin Liu, Chunyi Peng, Dr. Jacky Shen, Dr. Haitao Wu, Dr. Yongqiang Xiong, Dr. Fan Yang and Dr. Yongguang Zhang, whose suggestions have been very valuable. It has been a pleasure collaborating with and learning from these very smart research colleagues.

I am very thankful for many friends and colleagues I have had at Göttingen. Special thanks to Sven Anderson, Christian Dickmann, Jan Demter, Jun Lei, Ralf Lübben, Deguang Le, Fabian Meyer, Niklas Neumann, Henning Peters, Lei Shi, Niklas Steinleitner, Florian Tegeler, Hannes Tschofenig and Swen Weiland for many stimulating discussions and warm friendship. I would also like to thank many other friends, especially Klaus Alten, Tsai-Wen Chen, Tachao Kao, Bei-Jung

Lin, Chen-Yu Lin, Stefan Leonhardt, Kevin Ivory, Haruko Kariya and Benjamin Schelwis; my years at Göttingen would not have been the same without them.

I want to thank my fiance, Xiang Liu, for his love, support, encouragement, sense of humor, and for being an intelligent colleague as well. I could not have accomplished this without him. Words cannot express my gratitude to my sisters and their families, Yun-Hsin Kuo, Sen-Jan Yang, Yi-Shan Yang, Cheng-Chun Yang, Chia-Fen Kuo, Yu-Kuo Tsai, Winnie Tsai, Chuan-Shao Kuo, Ching-Sheng Teng, Wei-Ren Teng and Chien-Yin Kuo, for their constant encouragement, support and love from overseas. And finally, yet most importantly, my parents, Chien-Ho Kuo and Su-Lien Wu. Without their love, care and encouragement, I would not have come this far. This thesis is dedicated to them.

List of Publications

1. Fang-Chun Kuo, Kun Tan, Xiang-Yang Li, Jiansong Zhang, and Xiaoming Fu, “XOR Rescue: Exploiting Network Coding in Lossy Wireless Networks,”6th IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON 2009), Rome, Italy, June 2009.

2. Fang-Chun Kuo, Xiaoming Fu, Hannes Tschofenig, and Dieter Hogrefe, ”Performance Analysis of Key Exchange Methods for Transport Layer Security (TLS)”, submitted to Elsevier Computer Communications Journal, 2008.

3. Fang-Chun Kuo and Xiaoming Fu, “Probe-Aided MulTCP: An Aggregate Congestion Control Mechanism,” ACM SIGCOMM Computer Communication Review, volume 38, pages 19-28, 2008.

4. T.D. Nguyen, Fang-Chun Kuo, Lie-Liang Yang, and Lajos Hanzo, “Amalgamated Generalized Low Density Parity Check and Luby Transform Codes for the Wireless Internet,” in Proceedings of IEEE VTC2007-Spring, Dublin, Ireland, 22-25, 2007.

5. Fang-Chun Kuo, Hannes Tschofenig, Fabian Meyer, and Xiaoming Fu, “Comparison Studies between Pre-Shared and Public Key Exchange Mechanisms for Transport Layer Security,” in Proc. of the 9th IEEE Global Internet Symposium, in conjunction with IEEE INFOCOM 2006, Barcelona, Spain, IEEE, 2006.

6. Fang-Chun Kuo and Lajos Hanzo, “Symbol-Flipping Based Decoding of Generalized Low-Density Parity-Check Codes Constructed over GF(q),” in Proc. of the IEEE Wireless Communications and Networking Conference 2006, Las Vegas, NV USA, IEEE, 2006.

7. Ronald YS Tee, Fang-Chun Kuo and Lajos Hanzo, "Generalized Low-Density Parity-Check Coding Aided Multilevel Codes," in Proc. of IEEE VTC2006-Spring, Melbourne, Australia, 7-10, 2006.

8. Ronald YS Tee, Fang-Chun Kuo and Lajos Hanzo, "Multilevel Generalized Low-Density Parity-Check Codes," in IEE Electronics Letters, Vol. 42, pp. 167 - 168, 2006.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Wireless, in its various forms, is a growing prevailing communication medium. It provides a means for mobility, city-wide Internet connectivity, distributed sensing as well as other communication purposes. Most wireless networks offer one-hop wireless connectivity through access points (APs), e.g. Wireless local area networks (WLANs). They are being deployed at an accelerated pace both in private networks, such as campus and corporate networks, and in public areas, such as homes, offices, airports, malls, hotels, parks, and arenas, providing seamless, high-speed connectivity to the Internet. The reason for the rapid growth is the availability and low cost of IEEE 802.11 wireless networking products. Consequently, network access is almost instantaneous to any user for emails, instant messaging, file transfers, and web browsing. Furthermore, for sharing data across the network, consumer electronic devices such as digital photo cameras, and music players are being incorporated with WLAN interfaces. More specialized WLAN communication devices, such as mass storage, portable voice-over-IP (VoIP) phones, video-conferencing stations, gaming consoles, have also emerged.

The ability of wireless LANs to cater to large number of users having these exciting applications with fast and reliable network performance is increasingly important. The increasing appetite for wireless performance has in turn spurred extensive research effort to provide high data rates at the physical (PHY) layer. Current WLAN devices such as IEEE 802.11a [1] and IEEE 802.11g [2] are capable of delivering data at high raw bit rates up to 54 Mbps. Nevertheless, the performance achieved in today's WLANs is measured to be far lower than the highest achievable rate.

Temporary transmission failures in IEEE 802.11 WLANs are inevitable. Recent measurement studies [3, 4, 5] have shown that random losses are common and the loss ratio is distributed over a large spectrum. Many wireless links have medium loss ratios (e.g. $20\% - 60\%$). A significant cause for poor performance is data corruption during transmission over the wireless medium. The complex behavior of wireless signal propagation, particularly indoors, is due to noise, attenuation, interference, multipath, user mobility, depending on the transmission path traversed between an AP and a client station. These properties lead to transmission errors at the link layer, which in turn results in packet losses, low throughput, and higher and more variable packet latencies at higher layers.

Modern wireless networks deploy automatic repeat requests (ARQs) to shield transmission errors by retransmitting the corrupted frames. For example, in IEEE 802.11 standard, the AP would continue to retransmit a frame until the frame is successfully acknowledged or a limit of retransmissions is reached (i.e. four for data frames). Such retransmissions consume a significant portion of wireless capacity.

Motivated by these observations, in this dissertation, we propose a new design to build more reliable wireless networks. The key idea underlying our design is to provide the nodes with the ability to encode the multiple frames into one single "coded" frame before retransmitting them, i.e. to perform network coding for retransmissions.

Network coding (NC) is an emerging technique to improve the network capacity. It was first proposed in the context of wired networks [6] and subsequently applied to wireless multihop networks [7, 8, 9, 10]. Our work share the same concept with ER [11] and MU-ARQ [12], which employ network coding for efficient retransmissions in wireless networks, namely NC-aided ARQ. The potential coding gain of NC-aided ARQ has been demonstrated in ER [11] and MU-ARQ [12].

However, the design in ER and MU-ARQ only focuses on the coding aspect, independent from the characteristics of underlying physical layer (e.g. time-varying wireless links) and MAC capabilities and algorithms (e.g. multi-rate function in IEEE 802.11). A fundamental trait of wireless channels is that they exhibit time-varying fading effects, due in part to mobility and other user interference. As a result of this time-variation, a user's channel suffers not only periods of severe decay, but also periods when the channel gain is stronger than average. Owing to the multi-rate function in IEEE 802.11 MAC, the flows may be granted with higher transmission rates as their channel conditions are good. When many users are present, different users will experience peaks in their channel quality at different times. This effect has been called *multiuser diversity* [13]. It has been demonstrated in [7] that the network coding design should consider not only the coding opportunities but also multiuser diversity; otherwise, it would even degrade the network throughput.

This dissertation provides a practical NC-aided ARQ, namely XOR Rescue (XORR), for wireless networks. It assumes no synchronization or prior knowledge of senders or receivers, any of which may vary at any time. The key difference from past work is to provide a *global* approach to the design of a retransmission method by taking into account the following aspects:

1. Throughput improvement.
2. Short-term fairness in link-layer in order to satisfy QoS requirements in higher layers.
3. Adaptation to time-varying channel condition.
4. Rate adaptation by exploiting the multi-rate capability at the physical layer.
5. Utilization of coding opportunity.

The main characteristics of our approach is opportunism: each node relies on local information to detect and exploit the opportunities provided by not only network coding but also multiuser diversity whenever they arise. XORR has four components in its operation: reception estimation, coding metric calculation, NC-aware fair opportunistic scheduling, and NC-fair assignment.

1.1 Contributions

This dissertation contributes a novel link layer retransmission architecture for adopting a global approach in wireless networks. It develops XORR, a practical NC-aided ARQ scheme that focuses on not only the utilization of coding opportunity, but also the throughput improvement, short-term fairness in link-layer, adaptation to time-varying channel condition, and handling multiple transmission rates. It estimates the reception status without extra overheads and devises a new coding metric, which accommodates the effects of the frame size and the channel condition. An NC-aware fair opportunistic scheduling is designed, which is theoretically proven to achieve *NC-fairness*, i.e. not only the service time is evenly allocated, but also it improves the performance for *every* wireless station. Moreover, the dissertation studies the performance of XORR, providing theoretical analysis and extensive simulations and a real wireless testbed. A complete description of our contributions is in Chapter 7.

1.2 Dissertation Organization

We now briefly describe the organization of the dissertation. In Chapter 2, we provide the backgrounds related to this dissertation, which include technologies of link reliability enhancement, wireless scheduling algorithms, and introduction as well as design challenges of network coding. We also describe previous work in the areas. In Chapter 3, we describe in more details the problems facing NC-aided ARQ when the characteristics of wireless media (e.g. time-varying) and the algorithms in MAC (e.g. multi-rate) are considered. Chapter 4 forms the core of this dissertation, where we present XORR, a practical NC-aided ARQ scheme including four components in its operation: reception estimation, coding metric calculation, NC-aware fair opportunistic scheduling, and NC-fair assignment. The potential coding gain of XORR is theoretically characterized in Chapter 5. Using extensive simulations and a real wireless testbed, we show in Chapter 6 that XORR outperforms the IEEE 802.11 retransmission scheme, traditional wireless opportunistic scheduling and previous NC-aided ARQ. Finally, we conclude our work and outline future work in Chapter 7.

Chapter 2

Background and Related Work

Wireless networks have been designed using the wired network as the blueprint. Existing topology control algorithms assume underlying wireless links are static, either connected or disconnected. However, wireless medium is fundamentally different from wired links:

1. **High loss rate**. While wired network links are relatively reliable and predictable, wireless links are subject to high bit error rate caused by interference, noise, and fading.

2. **Time-varying and Heterogeneity**. The wireless link characteristics could change dramatically even over time durations lasting just milliseconds. Furthermore, in wireless networks, signals are transmitted over channels having different characteristics and distance. Thus the conditions of wireless links are time-varying and heterogeneous.

3. **Broadcast**. Wired links are unicast links, but the majority of wireless links (with omni-directional antennas) are broadcast links. Therefore, transmissions in a wired network do not interfere with each other, whereas interference is common case for the wireless medium.

Current wireless networks are based on the design rationale for wired networks, which does not work well with the characteristics of the wireless medium. As a result, they suffer suffer low throughput and dead spots where the wireless connection fades in and out, or drops off completely. The characteristics of wireless networks may all seem disadvantageous at the first sight, but a different perfective reveals that some of them can be used as an advantage, albeit with a fresh design.

In this chapter, we introduce the reader to the basic concepts and related work which utilize wireless characteristics to enhance wireless performance. We begin in Section 2.1 with the discussion of how to enhance link reliability in wireless networks. Section 2.2 presents the scheduling algorithms in wireless networks. Finally, Section 2.3 introduces network coding. It also further describes the applications and challenges of network coding in wireless networks.

2.1 Link Reliability Enhancement

In wireless communications, frames can be lost due to errors, collisions and hidden nodes. Even with the advent of a variety of physical techniques such as spread-spectrum and OFDM modulation [14], and channel coding (e.g. Turbo Codes [15], LDPC codes [16], and RS codes [17]), current systems still rely heavily on link-layer retransmissions to recover from bit errors and achieve high capacity. The following gives a review of approaches for addressing wireless link reliability issue.

- **Current MAC-layer Retransmission Mechanism**

IEEE 802.11 standard [18] represents the MAC (Medium Access Control) solution and provides a reliable link layer by handling the packet delivery problems using a Stop and Wait ARQ (Automatic Repeat Request) scheme. With Stop and Wait ARQ, each transmitted frame must be acknowledged before the next frame can be sent. If either the frame or its acknowledgment is lost, the frame is retransmitted in its original form by using a binary exponential back-off algorithm, where its contention window is doubled every time after a failed transmission until it reaches its maximum value of the window size. The advantages of this scheme are the high reliability of data delivery and the ease of implementation. However, such an ARQ scheme is inefficient because of the waste of wireless capacity.

- **Adaptive FEC Scheme**

Forward error correction (FEC) is often applied for reducing the bit error rate (BER). The efficiency of applying FEC relies on the knowledge of the current BER. Ahn et al. [19] propose an adaptive FEC algorithm which dynamically adjusts the amount of FEC coding per packet based on the presence or absence of receiver acknowledgments. However, it is very difficult to adjust the FEC redundancy based on the current channel error rate because the channel quality changes very quickly and unpredictable.

- **FEC-aided Retransmission Scheme**

The idea of recovering a frame by combining it with a retransmitted version was first proposed in [20] and then further analyzed in [21, 22]. Hybrid ARQ is an extension of this technique, which combines FEC and retransmission to recover unsuccessful transmission. Hybrid ARQ is a way of combining FEC and ARQ. Type I hybrid ARQ schemes [23] retransmit the same coded data in response to receiver NACKs. Chase combining [24] improves this strategy by storing corrupted frames and feeding them all to the decoder. Type II hybrid ARQ schemes [23] forego aggressive FEC while the channel is quiet, and send parity bits on retransmissions, a technique called incremental redundancy [25]. Metzner [26] and later Lin and Yu [27] have developed type II hybrid ARQ schemes based on incremental redundancy.

- **Partial Retransmission Scheme**

Retransmitting an entire frame works well over wired networks where bit-level corruption is rare and a frame loss implies that all the bits of the packet were lost. Over wireless medium, however, it is very often that only partial bits in a frame are corrupted. Therefore, it is wasteful to retransmit the whole frame. For more efficient retransmissions, Seda [28] and PPR [29] propose to partially retransmit the corrupted frames. Seda splits the frame into fragments and sends multiple checksums per frame. If bit errors are concentrated in only a few bursts, then entire fragments will checksum correctly, and the receiver would then only have to recover the erroneous fragments from the sender. Instead of using per-fragment checksum, PPR uses physical information for the higher layer to detect the bit errors.

- **Spatial Diversity**

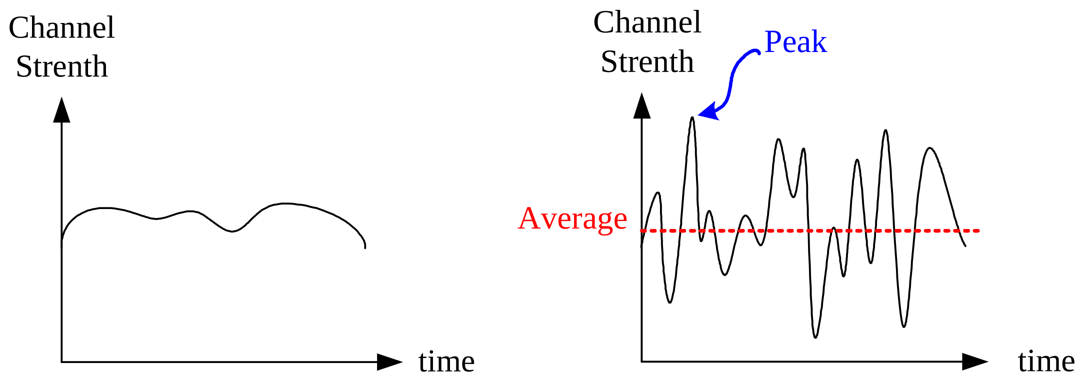
The broadcast nature of wireless provides an opportunity to deal with its unreliability; when a node broadcasts a packet, it is likely that at least one nearby node receives it, which can then function as the next-hop and forward the packet. This is totally opposite to the present wireless design, where there is single designated next-hop, and when it does not receive the packet, the previous hop has to retransmit it. The property is call *spatial diversity* and has been explored in the literature for recovering faulty frames, e.g. MRD [30], SOFT [31] and SPaC [32]. While MRD and SOFT share the same idea of combining multiple erroneous versions of frames simultaneously received at multiple location-distributed access points, SOFT further utilizes physical layer information to recovery the packet. In SPaC, the nodes in the multi-hop wireless network may buffer the overheard corrupted frames (corrupted or correct). When two or more corrupted frames have been received, the frame can be recovered by combining corrupted frames without retransmit redundantly.

- **Collision Avoidance**

One of the major sources of packet losses in 802.11 wireless networks comes from the problem of packet collisions and hidden terminals [33, 34, 35, 36, 37]. Measurements from a production WLAN show that 10% of the sender-receiver pairs experience severe packet loss due to collisions [33]. Current 802.11 WLANs rely on carrier sense (CSMA) to limit collisions. In other words, senders sense the medium and abstain from transmission when the medium is busy. Karn proposed the RTS/CTS [38] mechanism to combat both hidden and exposed terminals. We now briefly describe the RTS-CTS mechanism for unicast transmission. When a node wishes to transmit, it sends an RTS frame to the intended recipient; this RTS frame contains the length of the proposed transmission. If the recipient hears the RTS, it replies immediately with a CTS; the CTS also contains the length of the imminent data transmission. Upon hearing the CTS, the initiator goes ahead with the transmission. Any node overhearing the CTS defers for a period of the oncoming data transmission. After a data frame is received, the recipient provides link-level ARQ feedback, by means of an ACK. However, in practice it proves to be overly conservative.

Nevertheless, experimental results show that enabling RTS-CTS significantly reduces the overall throughput [34, 37, 36, 39]. Accordingly, WLAN deployments and access point (AP) manufacturers disable RTS-CTS by default [40, 41]. Recent work [42, 43] advocates the use of successive interference cancellation (SIC) and joint decoding to resolve 802.11 collisions. Additionally, prior works have studied wireless interference [33, 34, 35, 36, 37, 44, 45] , and proposed MAC modifications to increase resilience to collisions [46, 47, 38, 48, 49]. Rather than avoiding collisions or proposing a new MAC, ZigZag [50] is a new form of interference cancellation that iteratively decodes strategically picked chunks, exploiting asynchrony across successive collisions and works within the 802.11 MAC.

2.2 Scheduling in Wireless Networks



(a) Wired Constant Channel

(b) Wireless Time-varying Channel

FIGURE 2.1: Wired and wireless channels.

Effective transmission over wireless channels is a key requirement of wireless communication. To achieve this one must address a number of issues specific to the wireless environment. As depicted in Figure 2.1, in contrast to wired constant channels, a fundamental trait of wireless channels is that they exhibit time-varying fading effects, due in part to mobility and other user interference. As a result of this time-variation, a user's channel suffers not only periods of severe decay, but also periods when the channel gain is stronger than average, as shown in Figure 2.1(b). When many users are present, different users will experience peaks in their channel quality at different times. This effect is called *multiuser diversity* [13]. It can be exploited by scheduling transmissions when a user has favorable channel conditions. Multiuser diversity gains can be achieved because when users experiencing good channels are selected, it enables the system to potentially operate close to its peak rather than average performance.

Multiuser diversity has its roots in the work of Knopp and Humblet [13], where they presented a power control scheme for maximizing the information theoretic network capacity of the uplink of a single cell with time-varying channels. Given the channel gain of each user, it is shown that capacity is maximized by allowing only the user with the best channel to transmit at any time. It has been further described from an information theoretic viewpoint in [51]. It underlies much of the recent work [52, 53] on "opportunistic" scheduling design for time-division downlink optimization in Code Division Multiple Access (CDMA) cellular networks, such as 1xEV-DO High Data Rate (HDR) [54] and High Speed Downlink Packet Access (HSDPA) [55] systems. A channel-aware ALOHA protocol is proposed in [56] to exploit multiuser diversity gains. In this work, all users base their transmission probability on their channel gain, assuming each user knows his own channel gain as well as the distribution of other users' channel gains. Recent work [57] has proposed leveraging the benefits of rate adaptation schemes by aggressively exploiting multiuser diversity in wireless LANs.

However, the main drawback of previous approaches is that individual quality of service (QoS) requirements cannot be taken into account in designing the scheduling policy. More specifically, in a networking context, the difficulty with previous approaches is that while it maximizes the overall throughput, it could result in significant unfairness among the users. For example, under such a scheme, stations that are close to the AP may always be favored over those that are further away, resulting in potentially poor performance for certain stations in the network. Since link layer fairness mechanisms serve as the basis for achieving network layer QoS, the scheduling algorithms must support some notion of "weighted fairness", wherein flows with larger weights receive correspondingly better service in accordance with a system-wide fairness model.

Accordingly, achieving fair allocation is an important goal for wireless networks. In [58, 59] the authors extend wireline scheduling policies to wireless networks and present wireless fair scheduling policies which ensure short term and long term fairness bounds. While these approaches provide fairness guarantees, the multiuser diversity is not exploited in those fair scheduling algorithms. In [60] the authors present an fair opportunistic scheduling called WCFQ (Wireless Credit-based Fair Queuing) to utilize multiuser diversity gain while providing (only) temporal fairness among the users with statistical fairness bounds. Their approach is based on CBFQ (Credit Based Fair Queuing) [61], a scheduler for wired systems. WCFQ trades off the fairness and throughput to exploit the channel time variations by mapping channel condition into a cost function. Another fair opportunistic scheduling approach was proposed in [62]. The objective in [62] is to develop a continuous channel scheduling scheme that maximizes system throughput subject to fairness constraints.

The IEEE 802.11a [1] and 802.11b [18] media access protocols provide a physical-layer multi-rate capability. With the multi-rate enhancement, transmission can take place at a number of rates according to channel conditions. Throughput fairness in wireless networks has received a great deal of attention [58, 59, 63, 64, 65, 66]. However, these approaches do not consider the effect of multi-rate in the wireless networks. Normalizing flow throughput in a multi-rate network would result in significant inefficiency and mitigate the gains of the multi-rate physical layer, as poor-channel flows would consume disproportionately more time and system resources [67]. It was shown that this phenomenon results from applying max-min fairness to the case of competing hosts with different bit rates [68]. Therefore, temporal fairness for wireless networks has been proposed [69, 70]. Their argument of using temporal fairness is while throughput fairness and temporal fairness are equivalent to each other in single-rate wireless networks, the distinction between temporal fairness and throughput fairness is critical in multi-rate networks. By using temporal fairness as a performance objective in wireless networks, two pathological situations can be eliminated: (i) performance anomaly in which the rate of a slower host limits the throughput of a fast host and (ii) starvation of slow hosts that may occur if an access point does not allow switching to a lower bit rate. In the method called Opportunistic Auto Rate, Sadeghi et al. proposed to grant hosts the same temporal-share of channel access as under single-rate 802.11 DCF [69]. Tan et al. reconsidered performance objectives in 802.11 DCF by proposing temporal fairness that focuses on time shares instead of rate shares [70].

2.3 Network Coding

2.3.1 Overview

Much of this section is inherited from [71].

- **General Principle**

Communication networks today are based on the assumption that the information is separate. Hence, information, e.g. packets over the Internet or signals in a phone network, is transported in the same way as cars share a highway or fluids share pipes. In other words, independent data streams may share network resources, but the information itself is separate. Routing, data storage, error control, and generally all network functions are based on this assumption.

Network coding (NC) is a recent field in information theory that breaks with this assumption. Network coding was proposed by Ahlswede et al. [6]. The core idea is that the forwarding nodes should merge data contained in distinct data in such a way that allows for recovery at the destination. This is in contrast with the traditional scheme which treats each data as a distinct object that must be delivered to the destination intact.

Network coding can be best illustrated through the butterfly example as shown in Figure 2.2. In the butterfly network, there are two sources, S_1 and S_2, and two two destination nodes, D_1 and D_2. Both packets P_1 and P_2 are delivered to both D_1 and D_2. Assume each link can transmit a packet in each time slot. If we only used routing, then the central link will be the bottleneck since it would be able to carry either P_1 or P_2, but not both. Suppose we send P_1 through the central link; then the D_1 would receive P_1 twice and not know P_2 at all. Similarly, sending P_2 poses the same problem for D_2. We say that routing is insufficient because no routing scheme can transmit both P_1 and P_2 simultaneously to both D_1 and D_2. On the other hand, as shown in the figure, both destinations can receive both packets simultaneously by sending the coded packet $P_1 \oplus P_2$ through the central link. Thus network coding can obtain a multicast throughput of two packets per time slot, strictly better than the routing approach which can at best achieve 1.5 packets per time slot.

- **Linear Network Coding**

Linear network coding [72], is in general, similar to the previous example in Figure 2.2, with the difference that the bitwise XOR operation is replace by a linear combination of the original packets, interpreted as numbers over some finite field [1] $\mathbb{F}_{2^m}$. The reason for choosing a linear framework is that the algorithms for coding and decoding are well understood. Note that linear combination is not concatenation, i.e. the length of the coded packet is still L after two packets with length L are combined.

I. **Encoding**

Assume $P_1, \cdots, P_n$ are original packets generated by one or several sources. In linear network coding, each coded packet X in the network is associated

[1] A finite field is a field with a finite field order (i.e. number of elements), also called a Galois field. The order of a finite field is always a prime or a power of a prime.

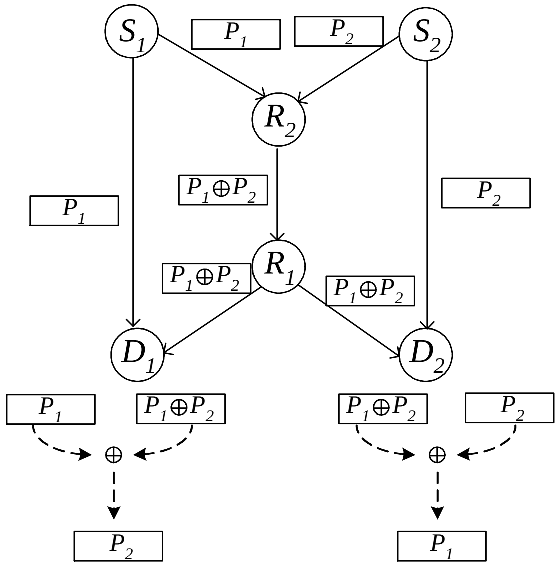


FIGURE 2.2: Butterfly Network: a scenario showing how network coding improves throughput.

with a vector of coefficients $\mathbf{c} = (c_1, \cdots, c_n)$ in $\mathbb{F}_{2^m}$, namely *encoding vector*. The coded packet can be derived as

$$X = \sum_{i=1}^{n} c_i \cdot P_i$$

Encoding can be performed recursively, namely, with already coded packets. Consider a node that has received and stored a set $(\mathbf{c^1}, X^1), \cdots, (\mathbf{c^k}, X^k)$ of coded packets, where $\mathbf{c^j}$ is the encoding vector of the jth coded packet X^j. This node may generate a new coded packet $(\mathbf{c}', X')$ by picking a set of coefficients $h_1, \cdots, h_k$ and computing the linear combination:

$$X' = \sum_{j=1}^{k} h_j X^j.$$

The corresponding encoding vector $\mathbf{c}'$ is not simply equal to h, since the coefficients are with respect to the original packets $P_1, \cdots, P_n$; in contrast, straightforward algebra shows that it is given by

$$c_i' = \sum_{j=1}^{k} h_j \cdot c_i^j.$$

This operation may be repeated at several nodes in the network.

II. **Decoding**

A node has received the coded packets, $X_1, \cdots X_k$ with the corresponding encoding vectors $\mathbf{c}_1, \cdots, \mathbf{c}_k$. In order to retrieve the original packet, $P_1, \cdots, P_n$, the system of equations need to be solved:

$$\begin{cases} X_1 = \sum_{i=1}^{n} c_i^1 P_i \\ X_2 = \sum_{i=1}^{n} c_i^2 P_i \\ \cdots \\ X_k = \sum_{i=1}^{n} c_i^k P_i, \end{cases}$$

where the unknowns are the original set of packets P_i. This is a linear system with k equations and n unknowns. When $k \geq n$ and there are at least n linear independent combinations, then system of equations can be solved to retrieve the n original packets, $P_1, \cdots, P_n$.

In practice, the linear equations can be solved as follows. A node stores the encoding vectors it receives as well as its own original packets, row by row, in a so-called *decoding matrix*. Initially, the matrix is empty. When a coded packet is received, it is inserted as the last row into the decoding matrix. The matrix of coefficients is transformed to triangular matrix by performing Gaussian elimination. A received coded packet is called *innovative* if it increases the rank of the matrix. If a coded packet is non-innovative, it is reduced to a row of 0s by Gaussian elimination and is ignored. As long as the encoding vector part of matrix contains a row of the form (e_i, X)[2], this node knows that the original packet P_i is equal to X. This occurs at the latest when n innovative coded packets are received. Note that it is not necessary to perform the decoding process at all nodes, but only at the receivers.

III. **Selection of Linear Combinations**

The problem of network code design is to select what linear combinations each node of the network performs in order to ensure the destination node receives at least n linear independent combinations from which it can decode the original packets. A simple algorithm is that each node in the network selects uniformly at random the coefficients over the field $\mathbb{F}_{2^m}$, in a completely independent and decentralized manner [73, 74]. With random network coding, there is a certain probability of selecting linearly dependent combinations [73]. This probability is related to the field size 2^m. Simulation results

[2] e_i is a unit vector with a single one at the ith position.

indicate that even for small field size (e.g. $m = 8$) the probability becomes negligible [75].

Alternatively, the deterministic algorithms can be used to design network codes, The polynomial-time algorithm for multicast in [76] sequentially examines each node of the network, and decodes what linear combinations each node performs. Since each node uses fixed linear coefficients, the packets only need to carry the information vector. There also exist deterministic decentralized algorithms that apply to restricted families of network configurations [77].

For all practical purposes, the size of the matrices with which network coding operates has to be limited. This is straightforward to achieve for deterministic network codes, but more difficult with random network coding. This is because the random network coded packets are usually grouped into so-called generations, and only packets of the same generation can be combined [78]. Size and composition of generations may have significant impact on the performance of network coding [79]. Similar considerations hold for the size of the finite field. Both parameters allow to trade off performance for lower memory requirements and reduced computational complexity.

- **Theoretical Gains**

Network coding achieves the optimal network capacity for multicast flows [6]. More specifically, consider a network that can be represented as a directed graph (typically, it is a wired network). The vertices of the graph correspond to terminals, and the edges of the graph correspond to channels. Assume there are M sources, each sending information at some given rate, and N receivers. All receivers are interested in receiving from all sources. Ahlswede et al. [6, 72] showed that network coding provides the following guarantee:

Assume that the source rates are such that, without network coding, the network can support each receiver in isolation (i.e. each receiver can decode all sources when it is the only receiver in the network). With an appropriate choice of linear coding coefficients, the network can support all receivers simultaneously.

In other words, when the N receivers share the network resources, each of them can receive the maximum rate it could hope to receive, even if it were using all the network resources by itself. Thus, network coding can help to better share the available network resources.

Furthermore, network coding benefits not only multicast flows but also other traffic patterns, such as unicast. Consider again the butterfly example in Figure 2.2 but assume now that source S_1 transmits to destination R_2 and S_2 to R_1. With network coding the sending rate is 1 packet per time slot, while without it, the sending rate is only 0.5 per time slot to each receiver.

- **Subsequent Work on Network Coding**

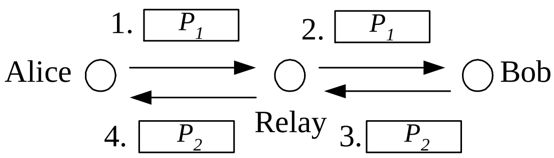
As discussed above, prior work shows that network coding achieves the multicast capacity of the network. The results have two practical implications. First, the combination of [72, 80, 81] shows that, for multicast traffic, linear network codes achieve the maximum capacity bounds, and coding and decoding can be done in polynomial time. Second, Ho et al. show that the above is true even when the routers pick random coefficients [73]. This enables distributed network coding, where routers do not need to coordinate with each other on the choice of codes. Network coding has been applied to many areas including wireless networks [82, 8], energy [83, 84], secrecy [85, 86], content distribution [87], reliability in DTN (Delay tolerance Network) [88], and distributed storage [89].

The classical network coding research is theoretical and hardly practical, which assumes multicast traffic, and ignores traffic burstiness and application requirements [6, 72, 80, 81, 73, 90, 91, 92]. In contrast, real-world packet networks are asynchronous and subject to random losses and delays. Recently, a few papers have employed network coding in wireless protocols to improve their throughput or reliability [8, 93, 94, 44, 95]. These papers are more focused on implementable protocols and practical issues than the early theoretical foundation.

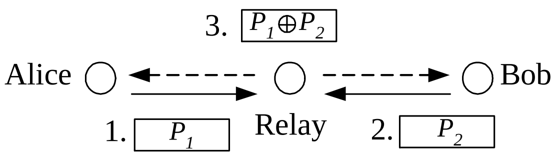
2.3.2 Wireless Network Coding

Because of the intrinsic characteristics of wireless links that complicate routing, namely, their unreliability, broadcast nature, and interference, wireless networks offer a natural space for utilizing network coding as an alternative approach for efficient transmissions. For example, the nature of wireless links further enriches coding possibility, as broadcasting can be achieved without power penalty. In other words, wireless networks exhibit high redundancy because a broadcast packet is heard by multiple nearby nodes. Network coding can exploit this redundancy to perform in-network compression of the data, thereby increasing the wireless throughput [8].

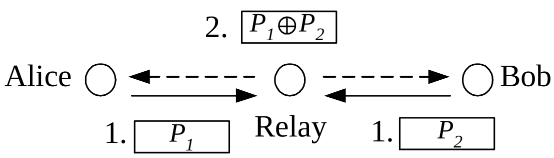
This can be illustrated using the simple example in Alice-and-Bob scenario [10], as shown in Figure 2.3. In this scenario, Alice wants to send packet P_1 to Bob and Bob wants to send P_2 to Alice. The radio range does not permit them to communicate directly and thus they need a *relay* in the middle to forward the packets. In the current design as shown in Figure 2.3(a), Alice sends her packet to the relay, which forwards it to Bob, and Bob sends his packet to the relay, which forwards it to Alice. Hence, the traditional strategy needs total 4 transmissions. With network coding as shown in Figure 2.3(b), the relay XORs the two packets and broadcasts the mixed packet, $P_1 \oplus P_2$. Accordingly, both Alice and Bob can decode the needed packets when receiving the coded packet, $P_1 \oplus P_2$. More specifically, Alice recovers P_2 by XOR-ing the coded packet with P_1, and Bob recovers P_1 in the same way. Thus, only 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 process exploits the existing redundancy in the network to compress the information, delivering two packets in a single transmission, and improving the throughput.



(a) Current Approach



(b) Packet-level Network Coding



(c) Analog Network Coding

FIGURE 2.3: The basic Alice-and-Bob scenario: illustration of wireless network coding. Here a number in front of a packet denotes the time-instance when the packet is transmitted.

Furthermore, while interference has traditionally been considered harmful, network coding allows the nodes to exploit interference strategically, and perceives it as a special code which compresses the number of transmissions and improves throughput [94]. This can be demonstrated with the same Alice-and-Bob example in Figure 2.3. As discussed above, Alice and Bob wish to exchange a pair of packets, and they require four time slots with current architecture and three time slots with packet-level network coding scheme. But applying network coding in analog signal is even better; it accomplishes the exchange in two time slots. As shown in Figure 2.3(c), Alice and Bob transmit their packet *simultaneously*, allowing their transmissions to interfere at the relay. This requires only one time slot. Due to interference, the relay receives the sum of Alice's and Bob's signals, which it cannot decode. The relay simply amplifies and forwards the received interfered signal. Accordingly, there are totally two time slots needed. Thus, compared to the traditional approach, analog network coding reduces the required time slots from four to two, doubling the throughput.

In summary, the synergy between the characteristics of the wireless medium and network coding coupled with the fact that wireless networks are more amenable to innovative designs than their wired counterparts, opens up many opportunities for successful research.

2.3.2.1 Challenges

A new network architecture that employs network coding and exploits the broadcast nature of the wireless medium would require the research community to rethink the network stack. Most of current medium access control, routing, and transport protocols are imported from the wired domain, with minor modifications. They are designed for working over point-to-point links, assume a single predetermined path and a layered architecture. The cost of redesigning our network stack is non-negligible. Nevertheless, the wireless medium is a scarce resource, which warrants efforts to investigate more efficient architectures. Furthermore, the wireless environment is more amenable to new deployments than the wired environment; usually such deployments can rely solely on software updates.

- **Challenges of the Broadcast Nature of Wireless Networks**

The benefits of network coding lie in that it utilizes the broadcast nature of the wireless medium to simultaneously transmit a single packet to multiple receivers. Most of the new challenges are not caused by network coding, but are rather a side product of relying on the broadcast channel, which has implications on MAC, routing, and transport protocols.

I. **MAC**: The standard access mode of 802.11 and similar MACs is a Distributed Coordination Function (DCF) combining carrier sense multiple-access (CSMA) with collision avoidance (CA). A node wishing to transmit has to first listen to the channel for a predetermined amount of time so as to check for any activity on the channel. If the channel is sensed *idle* then the node is permitted to transmit. If the channel is sensed as *busy* the node has to defer its transmission.

   In the 802.11 *unicast* mode, if the sender does not receive an acknowledgment within a specific period after transmitting, it assumes that there was a collision and selects a random backoff timer uniformly distributed within a contention window. The contention window doubles for every failed transmission in order to reduce the probability of collisions.

   In contrast, the 802.11 *broadcast* mode does not provide both reliability and backoff. A broadcast frame has many intended receivers, and it is unclear how to add this functionality without creating significant complexity and the potential for ACK implosion. In the absence of the ACKs, the broadcast mode offers no retransmissions and consequently very low reliability. Additionally, a broadcast source cannot detect collisions, and thus does not back off. If multiple backlogged nodes share the broadcast channel, and each of them continues sending at the highest rate, the resulting throughput is then very poor due to high collision rates. Therefore, the wireless network coding schemes like [8] need to compensate the MAC broadcast issue using other mechanism. Katti et al. [8] proposed *pseudo-broadcast*, which piggybacks on 802.11 unicast and benefits from its reliability and backoff mechanism.

   Recent years have seen the growing popularity of multi-rate wireless network devices that can exploit variations in channel conditions and improve overall network throughput. Many rate-adaptation schemes haven developed that selectively increase data transmissions on a link when it offers good channel quality. It has been shown in [7] that wireless network coding should consider the underlying physical and MAC layer, rather than being just designed as an autonomous layer. Otherwise, network coding may not improve, but even reduce the network throughput.

II. **Routing**: Traditional routing protocols impose a point-to-point abstraction on wireless networks, and reduce routing in a wireless network to a shortest path computation on these directed links, as in wired networks. However, with broadcast, multiple nodes could simultaneously receive a packet and one or more of them might choose to transmit as a result. This changes the notion of routing from a single shortest path to a multi-path problem, where decisions are made after packet reception, rather than at the time of transmission. Multi-path routing can be formulated as a linear program (LP). Nevertheless, the difficulty arises from the broadcast nature of the wireless medium, which gives the LP formulation an exponential number of constraints [96]. The MORE protocol presents a practical low-complexity heuristic that addresses this issue [93].

III. **Transport**: As mentioned previously, the 802.11 MAC used in broadcast mode does not perform the usual link layer functionality of congestion avoidance and reliability. The resulting high loss rate needs to be addressed; otherwise it would be mis-interpreted as a signal of congestion by transport protocols like TCP, causing them to reduce the transmission rate unnecessarily.

- **Coding Challenges**

The coding challenges arise from the desire to combine several attractive properties, such as low complexity, delay and memory requirements, high achievable rates, and adaptability to unknown channel conditions.

I. **Coding Complexity**: Network coding requires intermediate nodes in the network to perform operations over finite fields in real-time. While the cost of XOR coding is usually low, the general linear codes over large finite field could be computationally expensive. Decoding operations have quadratic complexity, which becomes too slow for high throughput applications. Further encoding operations are also complicated since they involve multiplications in large finite fields. This makes their use in high throughput applications questionable. Encoding and decoding algorithms should have linear complexity for practical implementation. Therefore, the bitwise XOR is particularly suitable for wireless routers that can afford only the fastest operations. There is an increased effort to design lower complexity encoding and decoding algorithms inspired by low density codes, but this effort is still at its first steps.

II. **Coding Opportunity**: Since network coding exploits the broadcast nature of wireless medium through opportunistic network coding. The coding operation is only performed when the coding opportunities arise. Hence, the coding improvement depends on how many coding opportunities the coding scheme can offer and how efficient the coding scheme can exploit the coding opportunities. Sengupta et al. [9] show that a route selection strategy that is aware of network coding opportunities leads to higher end-to-end throughput when compared to coding-oblivious routing strategies.

2.4 Chapter Summary

Unlike wired links, the characteristics of the wireless medium include 1) high loss rate, 2) time-varying and heterogeneity, and 3) broadcast. Most of current medium access control, routing, and transport protocols in wireless network are extended or modified based on the design rationale for wireline domain. For achieving effective transmission over wireless channels, those characteristics must be addressed or even exploited. In this chapter, three perspectives which are most related to this dissertation for improving wireless performance are discussed, namely link reliability enhancement, wireless scheduling, and network coding.

Because of the inefficiency of current link layer retransmission, technologies of combining FEC with retransmissions, partial retransmission scheme, utilizing spatial diversity resulted from wireless broadcast property and avoiding collision are explored for enhancing wireless link reliability.

Opportunistic and fair scheduling is proposed for use in wireless networks by exploiting time-variation and heterogeneity of wireless links. Especially, current MAC protocols [1, 18] provide multi-rate functionality, which further enlarges the heterogeneity among links. Therefore, temporal fairness is advocated for eliminating the performance anomaly, where the “bad” users (with bad links) limit the performance of the “good” users (with good links), as well as the starvation of bad users caused by no fairness guarantee.

Network coding is a recent field in information theory that allows mixing of data at intermediate network nodes. Applying network coding in wireless networks is proposed to improve wireless throughput by taking advantage of broadcast and interference natures of the wireless medium. Wireless Network coding enables more efficient, scalable, and reliable wireless networks. These opportunities come with some challenges for rethinking MAC, routing, and transport protocols in order to integrate network coding into the wireless network design.

Chapter 3

Problem Statement and Challenges

It is well recognized that wireless links are prone to errors. Many wireless links have medium loss ratios (e.g. $20\% - 60\%$). As mentioned in Chapter 2, current link layer retransmission (ARQ) is inefficient. Network coding has been found as an innovative means to enhance unicast performance in wireless networks. For example, ER [11] and MU-ARQ [12] suggested that network coding is applicable for efficient retransmissions in wireless networks, namely NC-aided ARQ, and demonstrated the potential coding gain of NC-aided ARQ. However, the design in ER and MU-ARQ is independent from the underlying physical layer and MAC capabilities and algorithms. As mentioned in Chapter 2, the design for wireless networks has to take into account the special characteristic of the wireless medium. Without considering them, these independent coding designs may reduce the performance.

The purpose of this chapter is to provide the reader the problem statement and challenges of our design. We start by providing a few definitions in Section 3.1. Then the system model and an example for explaining NC-aided ARQ are described in Section 3.2. Finally, we talk about the challenges for designing a practical NC-aided ARQ in wireless networks in Section 3.3.

3.1 Definitions

Below are a few definitions that we use throughout this dissertation.

(a) **Native frame**: A non-network-coded frame.

(b) **Coded frame**: A frame that is network-coded from multiple native frames.

(c) **Original frame**: A native frame that has never been transmitted.

(d) **Retransmitted frame**: A native frame that is being retransmitted or a coded frame that contains only lost native frames.

(e) **Coding-set of a coded frame**: A set of native frames that are network-coded in a coded frame.

(f) **Decodable set**: A coding-set whose corresponding stations can retrieve their native frames by decoding the coded frame. For example, the station u_1 has p_2 and the station u_2 has p_1. A coded frame, $p_1 \oplus p_2$, with a coding-set $\{1, 2\}$ can be decoded by the stations u_1 and u_2. Therefore, the coding-set $\{1, 2\}$ is a decodable set and the size of the set is 2.

(g) **Maximal decodable set**. A decodable set with maximal size among all deccodable sets. For example, $\{1, 2, 3\}$ is the maximal decodable set among the decodable sets $\{1, 2\}$, $\{1, 3\}$, $\{2, 3\}$ and $\{1, 2, 3\}$.

(h) **Coding metric**: A metric that evaluates the benefit of a coding set.

(i) **Coding loss**: A node has *coding loss* if it performs worse with an NC scheduler than it would with a non-NC scheduler.

(j) **NC-fairness**: An NC scheduling achieves *NC-fairness* if it maintains certain fairness constraint (e.g. time or throughput) and no node has *coding loss* when applying it.

(k) **Goodput**: The number of bits per unit of time successfully received by the destination.

3.2 NC-Aided ARQ

3.2.1 System Model

In this thesis, we mainly consider single-hop wireless networks as shown in Figure 3.1. There is a wireless access point (AP) and a set of N stations, i.e.

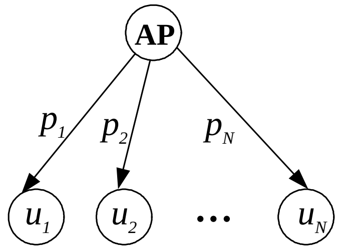


FIGURE 3.1: System model

$\mathcal{U} = \{u_1, u_2, \cdots, u_N\}$ which are associated with the AP. Stations only communicates with the AP directly. We denote r_j and γ_j the transmission rate and reliability of the link between the AP and a station u_j, respectively. Assume that all wireless links are mutually independent, i.e., the variables r_i $(1 \leq i \leq N)$ and γ_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 may overhear the frame with a probability of γ_j. Note that in this thesis, we mainly focus on downlink traffic, because most of the traffic in a WLAN is downloading. Later, we will show how network coding can also be beneficial in two-way traffic scenario in Section 4.6.

3.2.2 Example

As shown in Figure 3.2, the following example illustrates how network coding is employed for reducing retransmissions in WLANs. Assume that the AP transmits the frames p_1 to u_1. p_1 is not received by u_1, but overheard by u_2 and u_2 stores the overheard p_1 in its local pool. Later the AP transmits p_2 to u_2. p_2 is not received by u_2, but overheard by u_1 and u_1 stores the overheard p_2 in its local pool. By certain means, the AP obtains the reception information, i.e. u_1 possesses p_2 and u_2 possesses p_1. Based on the reception information, the AP uses the pre-defined coding metric to evaluate the benefit of different coding-sets. Accordingly, the AP schedules a coded frame, $p_1 \oplus p_2$, because two lost frames, p_1 and p_2, can be recovered in one retransmission if the coded frame is received by both stations. Therefore, one retransmission could be saved.

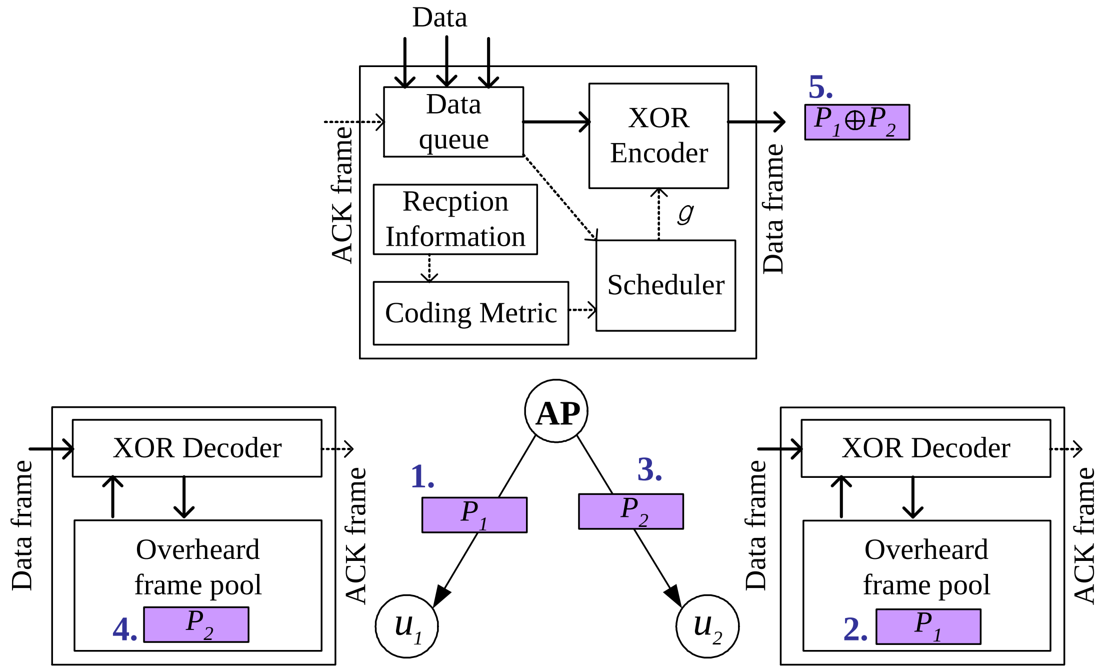


FIGURE 3.2: Example of NC-Aided ARQ. Here a number in front of a frame denotes the time-instance when the frame is processed

3.2.3 Cooperation with 802.11 MAC

For the sake of replacing the MAC-layer retransmission in 802.11, the sender disables the default MAC-layer retransmission by setting the MAC retry count[1] to 0. The new retransmission scheme retransmits the frames above the MAC layer until its receiver acknowledges the packet or the retry count is reached. In order to provide the same level of reliability, the retry count in new retransmission scheme is set to the original MAC retry count [11].

Since wireless network coding scheme broadcasts encoded frames, the natural approach would be to use broadcast mode in 802.11 MAC. Unfortunately, as we mentioned in Section 2, this does not work because of poor reliability and lack of backoff. Pseudo-broadcast, which piggybacks on 802.11 unicast, was proposed in [8] for benefiting from its reliability and backoff mechanism. With pseudo-broadcast, the link-layer destination field is set to the MAC address of one of the intended recipients. An XOR-header is added after the link-layer header, listing all recipients of the frame.

[1]MAC retry count is the maximum number of retransmissions at MAC layer.

Furthermore, all nodes are set in the promiscuous mode, and thus they can overhear frames not addressed to them. When a node receives a frame with a MAC address different from its own, it checks the XOR-header to see if it is a recipient. If so, it processes the frame further, else it stores the frame in a buffer as an opportunistically received frame. As all frames are transmitted using 802.11 unicast, the binary exponential backoff can help reduce collision losses under high load.

3.3 Challenges for NC-Aided ARQ

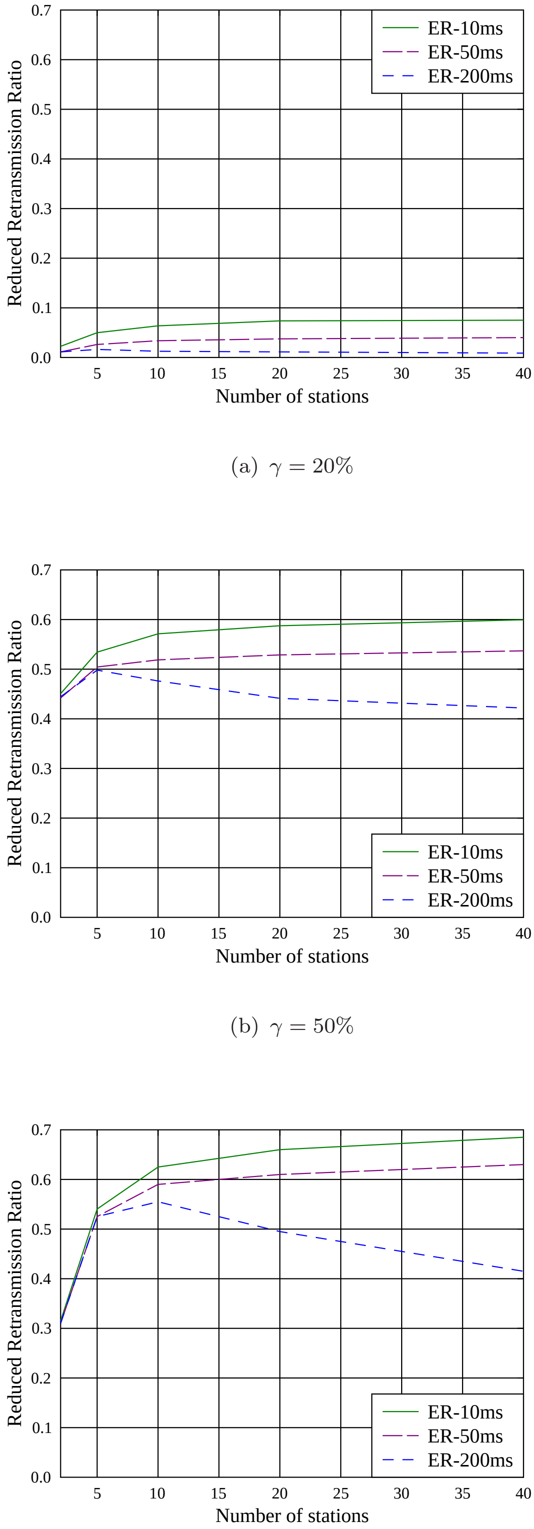
To enable a practical application of network coding to MAC-layer retransmission scheme, one needs to address the following challenges: learning reception information, coding metric, and NC-aware scheduling.

3.3.1 Learning Reception Information

As explained in the previous example, the AP requires the knowledge that which frames have been received by each station for making the coding decision. In previous works, this reception information is explicitly reported to the AP by each station, per frame [12] or periodically [11]. However, such a reception report scheme causes significant overheads which may be even larger than the coding benefit.

We run simulations of ER [11] to investigate the effect of the reception reports on the coding scheme. The static homogeneous link model is used in the simulations. The details of simulation setup for ER and the explanation of performance metrics can be found in Section 6.1. In ER, the stations send the reception reports periodically. We vary the period of the report transmission and evaluate the performance under different link conditions and the number of stations in the network.

Figure 3.3 and Figure 3.4 show the goodput gain and reduced retransmission ratio compared to traditional IEEE 802.11 MAC retransmission scheme, respectively. According to the results, we have the following observations. First, the coding efficiency of ER heavily depends on feedback information carried by reception reports. Hence, as shown in Figure 3.3, ER having a shorter report period may further reduce retransmissions, especially when the link reliability is higher.

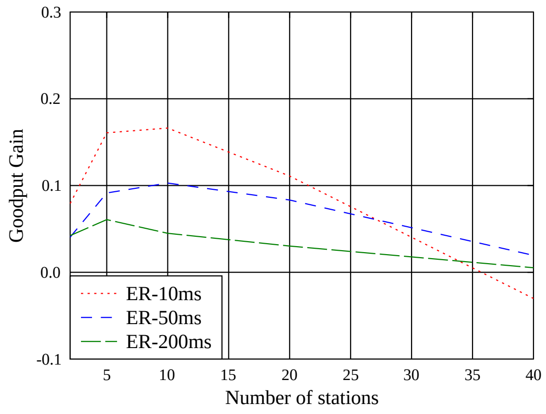


(a) $\gamma = 20\%$

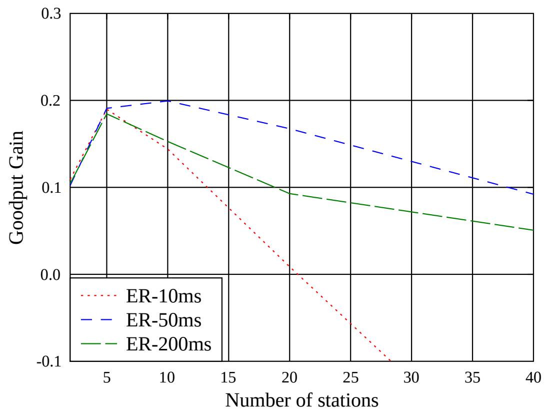
(b) $\gamma = 50\%$

(c) $\gamma = 80\%$

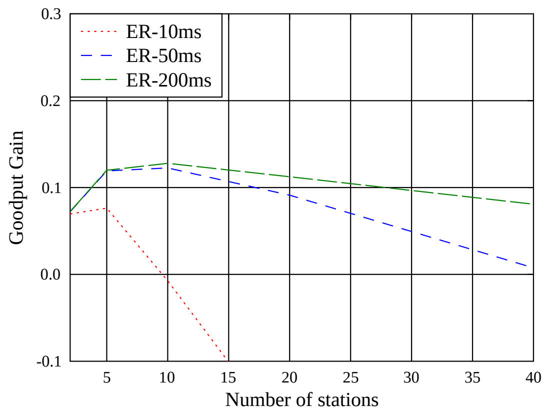
Figure 3.3: Reduced retransmission ratio with different numbers of stations in static channels.



(a) $\gamma = 20\%$



(b) $\gamma = 50\%$



(c) $\gamma = 80\%$

FIGURE 3.4: Goodput gain of ER with different numbers of stations in static channels.

However, signaling overheads incurred by frequent reports degrade the goodput performance severely. As shown in Figure 3.4, the goodput of ER having a report period of 10 ms is ever worse than that of *802.11* when the link reliability is high.

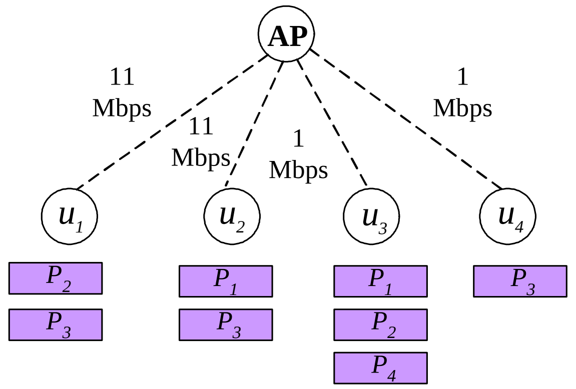
Second, as depicted in Figure 3.3, the reduced retransmission ratio degrades if the report period is too long, especially with larger number of stations and high link reliability. Therefore, although overheads may be reduced by choosing a longer feedback interval, it will cause insufficient reception information at the AP. Accordingly, the coding benefit may vanish due to less coding chances.

Third, in Figure 3.4, when $\gamma = 0.2$ and the number of stations is less than 20, ER with 10 ms period seems performs the best. However, when $\gamma = 0.2$ and the number of stations is greater than 20, 50 ms seems to be the optimal period. These show that it is difficult to make such a trade-off as it depends on many essentially dynamic parameters such as the number of stations and wireless channel conditions, which in practice are typically time-varying. The mis-chosen period may even degrade performance severely. In summary, the design goal is to efficiently learn reception information while avoiding or at least reducing the burden of reception reports.

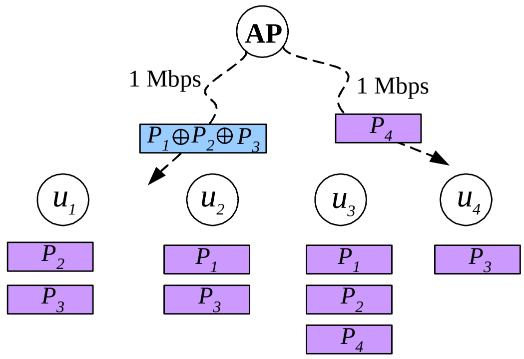
3.3.2 Coding Metric

The coding metric used in the previous example is the size of the decodable set. Sending the coded frame with the maximal decodable set means that more stations can retrieve their frames within this retransmission and hence the number of retransmissions is reduced. Existing NC schemes (e.g. [11]) are designed assuming that the AP always transmits at a constant transmission rate. According, reducing the number of retransmission is improving the overall goodput. However, in practical wireless networks, the AP can adjust its transmission rate based on each station's channel quality. When multiple frames are encoded, the coded frame must be transmitted at the lowest transmission rate in that set. In this multi-rate transmission scenario, sending the coded frame with the maximal decodable set may yield suboptimal results.

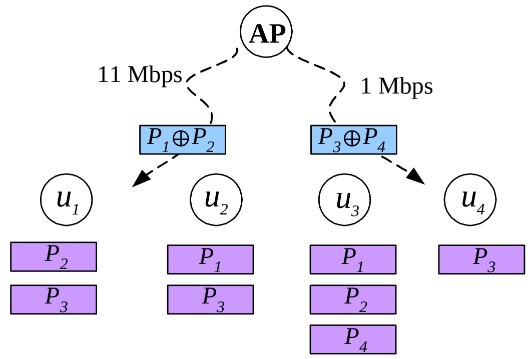
This can be best illustrated by the following example in Figure 3.5, where p_1, p_2, p_3 and p_4 are four frames for the stations u_1, u_2, u_3 and u_4, respectively. The reception status and transmission rate of each station are shown in Figure 3.5(a).



(a) Reception status and transmission rate of each station



(b) Schedule based on the coding metric of maximal decodable set



(c) Alternative schedule which reduces the transmission time by 45%

FIGURE 3.5: Example for multi-rate scenario.

Following the coding metric of maximal decodable set, $p_1 \oplus p_2 \oplus p_3$ is selected first because the coded frame $p_1 \oplus p_2 \oplus p_3$ may be received and decoded by stations u_1, u_2 and u_3 and consequently two transmissions are saved. And then p_4 is retransmitted alone for recovering. Two retransmitted frames, $p_1 \oplus p_2 \oplus p_3$ and p_4, are both transmitted at the rate of 1Mbps as shown in Figure 3.5(b). However, if we retransmit coded frames $p_1 \oplus p_2$ at the rate of 11 Mbps and $p_3 \oplus p_4$ at the rate of 1 Mbps as shown in Figure 3.5(c), this improves the goodput by 45% compared to the selection based on the coding metric of maximal decodable set. In summary, the coding metric considering only coding effect, i.e. the maximal decodable set, is insufficient and even harmful to the system performance.

3.3.3 NC-aware Scheduling

IEEE *802.11* uses a per-node queue with a per-node backoff. More specifically, each node maintains a single first-in-first-out (FIFO) transmission queue and the frames targeted for different destinations are buffered in common FIFO queue.

ER [11], a previous NC-aided retransmission scheme, inherits this per-node FIFO queue from *802.11* and further defers the retransmissions until a pre-defined threshold is reached for more coding opportunities. There are two issues for using per-node FIFO queue with network coding in ER:

1. **Head-of-line (HOL) Blocking**. Per-node FIFO queue in IEEE *802.11* may result in HOL blocking [97]. When the HOL frame is destined to a receiver with bad channel condition, it prevents other frames from being transmitted. Thus, all other stations in the network suffer throughput degradation.

2. **Frame-reordering**. Since the deferred retransmitted frames in ER may target to the same stations, the frames may be delivered out-of-order if the later frames are decoded successfully first. Such frame-reordering has an adverse impact on TCP.

These HOL blocking and frame-reordering problems can be solved by separating queue for different wireless stations (per-flow queue) and scheduling only one frame for each station till the frame is ACKed. However, an opportunistic scheduling with per-flow queue may lead to starvation of the users experiencing a bad channel for prolonged periods of time [58, 98]. Therefore, the scheduling should be balanced with fairness considerations. Fair opportunistic scheduling has been well studied in the literature [52, 60]. Nevertheless, it is non-trivial to design an *NC-aware* fair opportunistic scheduling. It becomes non-intuitive to accounting resource cost (e.g. service time or throughput) for each station in a coding-set, since this cost is essentially shared among these stations when frames are mixed together.

3.4 Chapter Summary

This chapter gives an overview of how the NC-aided ARQ scheme works. Due to the overhearing from stations, the AP can retransmit a single coded frame for several stations and further reduce the number of retransmissions. ER [11] and MU-ARQ [12] have shown the potential coding gain of NC-aided ARQ. However, their coding schemes work independently without considering the characteristics of wireless media (e.g. time-varying) and the algorithms in MAC (e.g. multi-rate).

The problems caused by this independent coding design and the challenges which have to be addressed are further discussed:

1. How to avoid or reduce the overhead of getting reception information?
2. How to define a proper coding metric which considers not only the coding gain but also the effect of the heterogeneity and time-variation of wireless links?
3. NC-aware scheduling.

In next chapter, we will describe in detail how the proposed scheme can address these challenges.

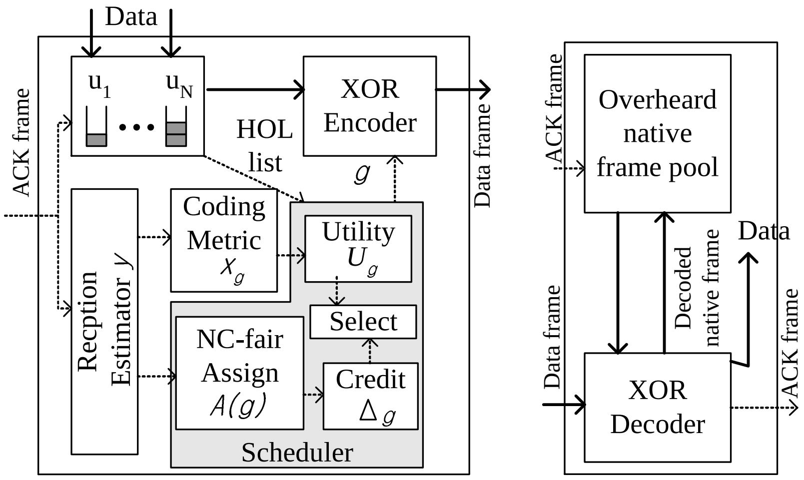
Chapter 4

XOR Rescue: A Practical NC-Aided ARQ Scheme

By tackling the challenges mentioned in Section 3.3, this chapter proposes a practical NC-aided ARQ protocol, namely XOR Rescue (XORR). Unlike previous NC-aided ARQ schemes [12, 11] focusing on only coding aspect, XORR provides a global approach to the design of a retransmission method by taking into account the following aspects:

- Throughput improvement.
- Short-term fairness in link-layer (for achieving QoS requirement in higher layer).
- Adaptation to time-varying channel condition.
- Handling multiple transmission rates.
- Utilization of coding opportunity.

We start this chapter by describing the high level architecture design of XORR in Section 4.1. It assumes no synchronization or prior knowledge of senders or receivers, any of which may vary at any time. The main characteristic of our approach is opportunism: each node relies on local information to detect and exploit the opportunities provided by not only network coding but also multiuser diversity whenever they arise. XORR has four components in its operation which can be succinctly summarized as follows: reception estimation (Section 4.2), coding metric calculation (Section 4.3), NC-aware fair opportunistic scheduling (Section 4.4), and NC-fair assignment (Section 4.5).



(a) AP side (b) Station side

FIGURE 4.1: XORR architecture.

4.1 Architecture

The system architecture of XORR is shown in Fig. 4.1. To avoid HOL blocking, the AP maintains a per-flow queue for each station. Furthermore, for preventing frame reordering, only the HOL (head-of-line) frames of the stations are the candidates for scheduling selection. In the original frame transmission mode, the sender sends the HOL frames, following 802.11's contention mechanism. Let p_i denote the HOL frame of the station u_i. If this HOL frame p_i is an original frame, u_i belongs to $TxGroup$. Otherwise, it belongs to $RetxGroup$.

There are several types of network coding schemes available [72]. XORR adopts a simple and effective scheme based on bitwise XOR, which is particularly suitable for wireless devices that can only afford the fastest operations. More specifically, when the AP wins channel contention, it performs a bitwise XOR of multiple frames destined for different stations. One coding principle in XORR is *never encoding an original frame* with any other frames. Since an original frame has never been transmitted before, if it is encoded with other retransmitted frames, this coded frame can only be decoded by the destination of the original frame. Furthermore, if more than two original frames are encoded together, this coded frame has no chance to be decoded by any stations. Therefore, at each scheduling time, a scheduler selects either an original frame from $TxGroup$ or a set of retransmitted frames [1] from $RetxGroup$ to transmit. If the coding-set with multiple frames is

[1]Note that the set may contain only one native frame.

selected, these frames are encoded using XOR operation into a coded frame. For example, the frames, p_1, p_2, p_3, are selected as a coding-set g, i.e. $g = \{1, 2, 3\}$. Then the coded frame p_g is generated by encoding $p_1 \oplus p_2 \oplus p_3$, where $\oplus$ is bitwise XOR operation.

Each time when a station u_i overhears a native frame p_j, it stores p_j into its native frame pool and may use it to decode other coded frames. If a coded frame p_g (g denotes the set of frames that are encoded.) is received by the station u_i, u_i tries to decode *any* native frame immediately, no matter the decoded native frame is for itself or not. A station can successfully retrieve 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 to itself, i.e. p_i, it should transmit an acknowledgment (ACK) to the AP to confirm the reception. Upon receiving the ACK, the AP removes the frame from the queue. If the decoded native frame is for other stations, it is stored in the local frame pool. After decoding, the coded frame is discarded. Furthermore, each station only needs a queue containing at most $(N - 1)$ native frames in its local frame pool for decoding because XORR only schedules HOL frames of N stations.

Before going into the details of the designs, we briefly sketch the key components in XORR, as shown in Figure 4.1:

- **Reception estimation**. In XORR, the AP does not require its client 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. Hence, a station only acknowledges the reception of its own frames and no extra feedback signaling overhead is introduced by this estimation scheme.

- **Coding metric for adapting coding opportunities and multiuser diversity**. An NC scheme selects a set of frames to encode so that the coded frame can achieve the maximal coding metric. However, selecting the maximal decodable set as coding metric is inefficient because it does not accommodate the heterogeneities in wireless networks, such as transmission rate, link reliability and frame size. A new coding metric, expected goodput, is devised in XORR to replace the metric of the maximal decodable set for measuring the coding benefit. Expected goodput is defined as the achieved system goodput by transmitting a certain (native or coded) frame. To calculate it, the coding effect and the instant link quality are taken into account. More specifically, using expected goodput as a coding metric is opportunism in the aspect of both the coding and the wireless condition, i.e. the AP relies on local information to detect and exploit efficient coding as well as better link opportunities whenever they arise. Therefore, the gain of network coding and multiuser diversity can be utilized at the same time.

• **NC-aware scheduling framework**. A framework of an NC-aware fair opportunistic scheduling is proposed in XORR. The task of the scheduling discipline is to optimize the system performance (utility) under certain fairness constraint. To provide a bounded short-term fairness among all clients, XORR follows a credit based approach as in [60] that assigns a state variable, *credit*, to control the fairness property, but it is extended to support network coding. More specifically, unlike prior work which only selects a single frame to transmit, a set of frames may be selected for transmitting a coded frame. Note that the traditional scheduling disciplines can be regarded as a special case of our NC-aware scheduling, where the selection candidate contains only one frame.

• **NC-fair assignment**. In traditional non-NC fair scheduling, the goodput performance of the station is linear determined by the resource cost (e.g. service time or bandwidth) assigned to it. Accordingly, the fairness guarantee implies certain performance guarantee among the stations. However, with NC, such implication becomes tricky because the resource cost for transmitting a coded frame can be arbitrarily assigned. Therefore, in this dissertation, two terms are defined for clarifying the fairness in NC-aware scheduling:

Definition 4.1 (Coding loss)**.** A node has *coding loss* if it performs worse with an NC-aware scheduler than it would with a non-NC scheduler.

Definition 4.2 (NC-fairness)**.** An NC-aware scheduling achieves *NC-fairness* if it maintains certain fairness constraint (e.g. time or bandwidth) and no node has *coding loss* when applying it.

Thus, the task for designing an algorithm of resource cost assignment for the members whose intended frames are network coded together is to achieve NC-fairness. To the best our knowledge, this is the first work on addressing the fairness issue in the NC-aware scheduling 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 uses Bayesian-learning-based method for estimating 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[2] γ_i^t to each station u_i. This information is already available for most of existing wireless networks. We will discuss more details about learning current link reliability in Section 4.6.

The AP maintains a score-table $\mathcal{Y}$ that has $N \times N$ entries, where N is the number of current backlogged stations. Each entry $y_{i,j}$ records the probability for u_i to have the HOL native frame of station u_j. Initially, the table contains all zeros. 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.

- **Transmitting a native frame**. 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

$$y_{i,j}^{t+1} = 1 - (1 - y_{i,j}^t)(1 - \gamma_i), \; i \neq j. \tag{4.1}$$

- **Transmitting a coded frame**. 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

$$y_{i,j} = 1 - (1 - y_{i,j}) \cdot \left[(1 - \gamma_i) + \gamma_i (1 - \prod_{q \in g \setminus \{j\}} y_{i,q}) \right]. \tag{4.2}$$

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

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 that

$$\mathbf{Pr}\left(\overline{ACK_k^{t+1}}\right) = (1-\gamma_k) + \gamma_k(1 - \prod_{q \in g \setminus \{j\}} y_{k,q}).$$

Thus, $y_{k,j}^{t+1}$ is updated as

$$y_{k,j}^{t+1} = 1 - \frac{1 - y_{k,j}^{t}}{(1-\gamma_k) + \gamma_k(1 - \prod_{q \in g \setminus \{j\}} y_{k,q}^{t})}. \tag{4.3}$$

• **Receiving an ACK**. When a station u_i successfully decodes its own frame p_i, it should transmit 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 HOL frame to u_i. Furthermore, 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.

4.3 Coding Metric

XORR defines *expected goodput* as coding metric, while previous NC schemes choose maximal decodable set. Unlike the latter, the metric of expected goodput naturally accommodates the heterogeneities in wireless networks, e.g. data rate, link reliability as well as frame size. In this subsection, we focus on how to derive the expected goodput of a coded frame.

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 reception table $\mathcal{Y}$ maintained by the reception estimator.

Definition 4.3 (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 $D_i^g = \prod_{j\in g\setminus\{i\}} y_{i,j}$.

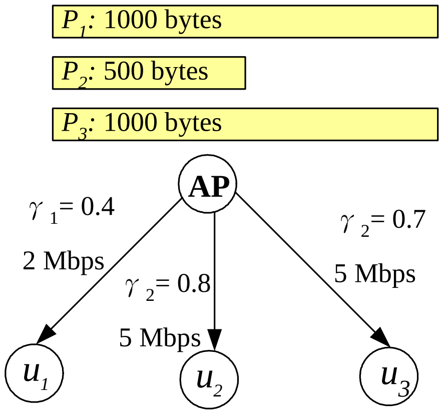
Let L_g be the size of p_g, then $L_g = \max_{j\in g} L_j$. Let r_g be the transmission rate for transmitting the coded frame, p_g, then $r_g = \min_{j\in g} r_j$. Thus, the transmission time of p_g is $T_g = \frac{L_g}{r_g}$. Finally, the expected goodput χ_g of transmitting the coding-set g is the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

$$\chi^g = \sum_{i\in g} \chi_i^g = \sum_{i\in g} \frac{L_i}{T_g} \cdot \gamma_i \cdot D_i^g. \tag{4.4}$$

Equation 4.4 implies that the expected goodputs of some coding-sets are low if many stations in the coding-set are unlikely to decode the coded frame. In XORR, only the coding-sets that improve the system goodput are considered. In other words, only the coding-sets that have higher expected goodput than that of any frame in the coding-sets transmitted alone are considered. A coding-set g satisfying this condition is called a *valid* coding-set. Hereafter, unless otherwise mentioned, a coding-set is always referred to as a *valid* coding-set.

Definition 4.4 (Valid coding set)**.** A coding-set g is a valid coding-set if and only if $\chi^g \geq r_i \cdot \gamma_i$, $i \in g$.

It is also clear 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 network coding in *RetrGroup*.



Reception table

	u_1	u_2	u_3
p_1		0.8	0.7
p_2	0.4		0.7
p_3	0.4	0.8	

FIGURE 4.2: Example of coding metric calculation.

Example: Assume there are three stations, u_1, u_2 and u_3, and the AP transmitted p_1, p_2 and p_3, but they are all lost. Figure 4.2 shows the link reliability and transmission rate of each link, the packet size as well as the estimated reception table. Following the coding metric defined in Equation 4.4, the expected goodputs for transmitting the native frames p_1, p_2 and p_3, and the coded frames $(p_1 \oplus p_2)$, $(p_1 \oplus p_3)$, $(p_2 \oplus p_3)$ and $(p_1 \oplus p_2 \oplus p_3)$ are

$$\begin{aligned}
\chi^1 &= 2 \cdot 0.4 = 0.8 \text{ (Mbps)} \\
\chi^2 &= 5 \cdot 0.8 = 4 \text{ (Mbps)} \\
\chi^3 &= 5 \cdot 0.7 = 3.5 \text{ (Mbps)} \\
\chi^{\{1,2\}} &= \frac{1000}{1000/2} 0.4^2 + \frac{500}{1000/2} 0.8^2 = 1 \text{ (Mbps)} \\
\chi^{\{1,3\}} &= \frac{1000}{1000/2} 0.4^2 + \frac{100}{1000/2} 0.7^2 = 1.3 \text{ (Mbps)} \\
\chi^{\{2,3\}} &= \frac{500}{1000/5} 0.8^2 + \frac{1000}{1000/5} 0.7^2 = 4.05 \text{ (Mbps)} \\
\chi^{\{1,2,3\}} &= \frac{1000}{1000/2} 0.4^3 + \frac{500}{1000/2} 0.8^3 + \frac{1000}{1000/2} 0.7^3 = 1.326 \text{ (Mbps)}
\end{aligned}$$

Accordingly, retransmitting the coded frame $(p_2 \oplus p_3)$ has higher expected goodput than that of the others.

4.4 NC-Aware Scheduling Framework

XORR maximizes the network performance under certain resource fairness constraint (e.g. service time or bandwidth), which allocates equal resource to each station in the network. The credit-based approach in [60] is extended for providing bounded temporal fairness among all stations in NC-aware scheduling. More specifically, the scheduling scheme in [60] incorporates both the transmission utility and fairness into the scheduling decision. Each station is assigned a state variable, *credit*, to control the fairness property. In other words, the credit is used to represent the available resource for the station to use. However, the scheduling scheme in [60] simply selects a native frame from the HOL frames. While XORR shares the same scheduling discipline, the selection candidates in XORR contain not only native frames, but also coded frames, which makes scheduling more complicated.

We now define the XORR scheduling algorithm by describing how credits are initially set, how they are used to make a scheduling decision, how they are increased when a flow is not scheduled, and how a credit is decreased when a flow is scheduled. Let K_i denote the credit of station u_i. The *deficit* credit of a station in the coding-set g is defined as its assigned resource 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:

$$\hat{g}^t = \arg\max_{g} \left(U_g^t - \Delta_g^t\right), \tag{4.5}$$

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 (possibly with only one frame) that maximizes the utility while having minimal resource *deficit* to ensure the fairness. A station accumulates its credit if it is not selected in a coding-set.

Following the scheduling decisions, all backlogged stations update their credits as described in Fig. 4.3.

Once a set g is selected and the coded frame is transmitted, all stations in g decrease their credits by the fraction of resource assigned to them (Line 2-4). If

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

FIGURE 4.3: Pseudo-code for updating credits.

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.

While throughput fairness and temporal fairness are equivalent in single-rate wireless network, the distinction between them is critical in multi-rate wireless networks [67, 68, 70]. Under temporal fairness, each user may allocate equal service time (air-time) instead of bandwidth. By using temporal fairness as a performance objective, two pathological situations can be eliminated: (i) performance anomaly in which the rate of a slower host limits the throughput of a fast host and (ii) starvation of slow hosts that may occur if an AP does not allow switching to a lower bit rate (the last approach is commonly used to deal with performance anomaly in current products). Therefore, in this dissertation, we mainly focus on temporal fairness in the scheduling. However, our NC-aware fair scheduling is a general framework, which can be used for temporal or bandwidth-based fairness.

In order to maintain temporal fairness, the channel time for transmitting the frames (native or coded) is allocated equally to all stations. If the coding-set contains only one frame, the allocated time is simply the time needed for transmitting that native frame. However, when transmitting a coded frame, the transmission time T_g is shared among the stations in that coding-set. We denote $\mathcal{A}(g, i)$ is the portion of service time assigned to station u_i, $i \in g$. Obviously, it has $\sum_{i \in g} \mathcal{A}(g, i) = T_g$, where T_g is the overall transmission time for transmitting the coded frame p_g. We will elaborate how service time is shared later in Section 4.5.

In the following, we describe the details of three key components in XORR scheduler: 1) fairness bound, 2) utility function and 3) coding-set selection.

4.4.1 Fairness Bound

The target of XORR fair scheduling is to fairly allocate the shared channel time. To characterize the fairness property of XORR, we derive a bound on the difference in allocated time duration for any stations u_i and u_j that are continuously backlogged in $[t_1, t_2)$, i.e.

$$|\alpha_i(t_1, t_2) - \alpha_j(t_1, t_2)|$$

where $\alpha_i(t_1, t_2)$ is the service time allocated to the station u_i during the time interval $[t_1, t_2)$. In order to prove the bound, we first present two results in Lemma 4.5 and 4.5. We then bound the service time discrepancy of any two stations in Theorem 4.7.

Lemma 4.5. *For any flow i and for any schedule time t, the credit counter value is always bounded by*

$$0 \leq K_i \leq \max_t \frac{L_i^t}{r_i^t} + U_{max}, \tag{4.6}$$

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

Proof. For the left part of the inequality, according to the credit update in Figure 4.3, the credit value is non-negative. Thus, $K_i \geq 0$.

For the right part of the inequality, the proof is separated into two cases according to whether or not user i is in the coding group g^t. In each case, there are two sub-cases regarding how the credit is updated.

Case I: $i \notin g^t$. If $\Delta_g^t > 0$, then from Equation (4.5) 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

$$\begin{aligned} K_i^{t+1} &= K_i^t + \Delta_g^t \\ &\leq T_i^t + U_g^t - U_i^t \\ &\leq \max_t \frac{L_i^t}{r_i^t} + U_{max}. \end{aligned}$$

On the other hand, if $\Delta_g^t \leq 0$, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t \\ &\leq \max_t \frac{L_i^t}{r_i^t} + U_{max}. \end{aligned}$$

Case II: $i \in g^t$. If $\Delta_g^t > 0$, according to the credit update, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t - \mathcal{A}(g,j) + \Delta_g^t \\ &\leq T_i^t + U_g^t - U_i^t - \mathcal{A}(g,j) \\ &\leq \max_t \frac{L_i^t}{r_i^t} + U_{max}. \end{aligned}$$

On the other hand, if $\Delta_g^t \leq 0$, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t - \mathcal{A}(g,j) \\ &\leq K_i^t \leq \max_t \frac{L_i^t}{r_i^t} + U_{max}. \end{aligned}$$

Considering both cases, Lemma 4.5 holds. □

Lemma 4.6. *Assume coding-sets $(g^{t_1}, \cdots, g^{t_2-1})$ are transmitted during time period $[t_1, t_2)$. For any flow i continuously backlogged during $[t_1, t_2)$, its received service time $\alpha(t_1, t_2)$ during $[t_1, t_2)$ can be expressed as:*

$$\alpha(t_1, t_2) = K_i^{t_1} - K_i^{t_2} + \sum_{t=t_1}^{t_2-1} \max\left(0, \Delta_g^t\right) \tag{4.7}$$

Proof. If no code set including flow i is transmitted during (t_1, t_2), there is no service time for flow i, i.e. $\alpha_i(t_1, t_2) = 0$. According to the credit update, the equality holds. If coding-set g^t including flow i is transmitted in (t_1, t_2), there are two cases:

1. If $\Delta_g^t > 0$, we have $K_i^{t+1} = K_i^t - \Delta_i^t + \Delta_g^t$. Thus

$$\begin{aligned} K_i^t + \max\left(0, \Delta_g^t\right) &= (K_i^{t+1} + \mathcal{A}(g,j) - \Delta_g^t) + \Delta_g^t \\ &= \mathcal{A}(g,j) + K_i^{t+1}. \end{aligned}$$

2. If $\Delta_g^t \leq 0$, we have $K_i^{t+1} = K_i^t - \mathcal{A}(g, j)$. Thus,

$$K_i^t + \max\left(0, \Delta_g^t\right) = \mathcal{A}(g, j) + K_i^{t+1}.$$

Considering both cases, Lemma 4.6 holds. □

The following Theorem 4.7 demonstrates that XORR scheduling discipline achieves bounded temporal fairness:

Theorem 4.7 (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*

$$|\alpha_i(t_1, t_2) - \alpha_j(t_1, t_2)| \leq \max_t \frac{L_i^t}{r_i^t} + \max_t \frac{L_j^t}{r_j^t} + 2U_{max}, \tag{4.8}$$

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.

Proof. Based on Lemma 4.5 and Lemma 4.6, the service discrepancy can be derived as following

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

Thus Theorem 4.7 is proven. □

Although our NC-aware scheduling is proved to ensure fairness guarantee, it is still insufficient to claim this scheduling achieves NC-fairness. As defined in Definition 4.2, an NC-aware scheduling achieves NC-fairness if two conditions are satisfied: 1) the scheduling provides fairness guarantee; 2) no coding loss happens in any stations in the networks. However, the performance of individual stations depends on how to assign the resource usage (time or bandwidth) to the stations whose frames are encoded in the coded frame. Thus, an NC-fair assignment algorithm is needed to properly assign the resource (time or bandwidth) to the members whose intended frames are encoded together to achieve NC-fairness. To the best our knowledge, this is the first work on addressing the fairness issue in the NC-aware scheduling in the literature.

4.4.2 Utility Function

• **Utility Function Mapping**. Having derived the fairness bound in terms of the utility function, one of the key problems we face next is how to select a proper transmission utility function. In other words, given the desired fairness requirements and the coding metric, what transmission utility function should be defined? Different coding schemes employ different coding metrics, which result in different utility functions to translate the information from the coding benefit. Given the coding metric, we now need to specify the mapping to the transmission utility in order to achieve the desired fairness property. Note that a similar methodology can be performed for different coding metrics.

In XORR, the expected goodput is used as the coding metric and the temporal fairness is required. The utility function of a coding-set g is defined as an increasing function of the expected goodput χ_g and is bounded as shown in Equation (4.9).

$$U_g = \beta \cdot T_{\max} \cdot (1 - e^{-\frac{\chi_g}{r_{\max}}}), \tag{4.9}$$

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 possible transmission rate. 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 [60]. More specifically, by tuning utility scaling factor, β, the system can obtain stronger or weaker fairness guarantees. Intuitively, with larger possible values given to the utility, the coding benefit weighs more heavily into XORR's packet selection decision. Consequently, a greater total coding gain will be achieved at the expense of a looser fairness constraint.

Furthermore, XORR can be viewed as a generalized version of NC-aware schedulers, which stems from its flexible utility function design mapped from different coding metrics. For example, the coding metric of the maximal decodable set used in [8, 11] can be applied to our NC-aware scheduler. In the future work, different coding metrics may be designed for different network coding schemes (e.g. linear coding). By adopting appropriate utility function mappings, the new coding schemes can be applied into our NC-aware scheduler.

• **Coding Opportunity**. XORR introduces a new coding opportunity to reduce the retransmissions. If the sender retransmits a frame whenever *RetxGroup* is non-empty, it achieves lowest retransmission delay. However, there is only one frame in *RetxGroup*, and the frame has to be sent by the sender itself and results in zero coding gain. Therefore, it is reasonable to defer the recovery of a lost frame for a while to explore the coding potential with the expense of longer retransmission delay. To strike a good balance between low delay and high coding gain, we artificially bias in choosing an original frame to transmit by multiplying a factor on the expected goodput. More specifically, the *biased* expected goodput χ_g^* is calculated as the following equation:

$$\chi_g^* = \begin{cases} \theta\chi_i, & \text{if } p_i \text{ is an original frame} \\ \chi_g, & \text{otherwise} \end{cases} \tag{4.10}$$

where χ_g is the expected goodput of coded frame g. $\theta \geq 1$ is called *deferring retransmission factor*, which gives some bias for scheduling original frames. By biasing for choosing an original frame, the AP defers the recovery of a lost frame for a moderate period for exploiting potential coding opportunities. Then the utility of the selected set is calculated using biased expected goodput, χ_g^*, in Equation 4.9.

4.4.3 Coding-set Selection

Each time before transmission, the scheduler finds a coding-set that maximizes $\arg\max_g U_g - \Delta_g$. The pseudo-code of the scheduler in XORR is shown in Fig. 4.4. 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. 4.3).

However, searching a best coding set in *RetxGroup* is complicated since the selection candidates are $2^{|\Psi|} - 1$, where Ψ is the set of stations in *RetxGroup*. The following theorem shows that it is NP-hard to search a best coding-set in *RetxGroup*.

Theorem 4.8. *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 $\epsilon > 0$.*

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 show 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 (Zero-error Probabilistic Polynomial time) [99], coding-set selection problem is also not approximable within $O(|\Psi|^{1-\epsilon})$. □

An exhaustive search algorithm for finding the optimal coding-set is computationally very expensive and is not feasible for practical use. In this thesis, the exhaustive search algorithm just serves as an interesting baseline comparison to quantify the effectiveness of the heuristic selection algorithm that will be proposed later.

4.4.3.1 Heuristic Coding-set Selection

We now describe a practical heuristic algorithm to solve the coding-set selection problem. Given the NP-hard nature of the problem, the heuristic is not guaranteed for optimal results. However as we will show in Section 6.3, it works well in practice. Thus, a heuristic algorithm is described in Fig. 4.5. The algorithm starts with finding the station in *RetxGroup* which maximizes utility minus time deficit. This station is added into the coding-set. Then, the algorithm tries to search again in the remaining stations in *RetxGroup* and find another station to form a better coding-set. This process continues until no more such stations can be found or all stations in *RetxGroup* are selected. Therefore, the complexity of the heuristic algorithm is $O(|\Psi|^2)$, where Ψ is the set of stations in *RetxGroup*.

4.5 NC-Fair Assignment

As aforementioned, when transmitting a coded frame, the resource (time or bandwidth) is shared among the stations in the coding-set. How to distribute the resource among them is critical. Since we mainly focus on temporal fairness in this

function scheduling
loop
    $g_{tx} \leftarrow \arg\max_{j \in TxGroup} U_j - \Delta_j$
    $g_{rx} \leftarrow$ SelectCodingSet $(RetxGroup)$
    if $U_{g_{tx}} - \Delta_{g_{tx}} < U_{g_{rx}} - \Delta_{g_{rx}}$ **then**
        $g_{tx} \leftarrow g_{rx}$
    end if
    EncodeAndTransmit(g_{tx})
    UpdateCredit (g_{tx})
end loop
end function

FIGURE 4.4: 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 4.5: Pseudo-code for heuristic coding-set selection

dissertation, the consumed resource is the service time used for transmitting the frames. Hence, in this section, the time assignment is derived for XORR maintaining temporal fairness. However, the resource assignment based on other fairness property can be easily extended from our service time assignment algorithm.

Our simulations show that even though the service time is evenly allocated by the NC-aware 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. We call that such stations have *coding loss* as defined in Definition 4.1. Therefore, for achieving NC-fairness, the service time assignment algorithm in XORR has to be designed carefully. Two terms are defined for explaining our assignment algorithm.

Definition 4.9 (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 4.10 (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 which ensures that the effective goodput of the station in the coding-set is no less than that if its native frame is transmitted alone.

Theorem 4.11. $\forall u_i \in g$. *If the service time is assigned proportionally to the* relative coding edge *of each station in the coding-set, i.e.*

$$\mathcal{A}(g,i) = T_g \cdot \frac{\psi_i^g}{\sum_{j \in g} \psi_j^g}, \tag{4.11}$$

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

Proof.

$$\begin{aligned}
\lambda_i^g &= \frac{\chi_i^g T_g}{\mathcal{A}(g,i)} \\
&= \chi_i^g \times \frac{\sum_{j \in g} \psi_j^g}{\psi_i^g} \\
&= \chi_i^g \times \frac{\sum_{j \in g} \frac{\chi_j^g}{\gamma_j r_j}}{\frac{\chi_i^g}{\gamma_i r_i}} \\
&= \gamma_i r_i \times \sum_{j \in g} \frac{\chi_j^g}{\gamma_j r_j}
\end{aligned}$$

Let $\hat{j} = \arg\max_{j \in g} \gamma_j r_j$. Then we have

$$\begin{aligned}
\sum_{j \in g} \frac{\chi_j^g}{\gamma_j r_j} &\geq \frac{1}{\gamma_{\hat{j}} r_{\hat{j}}} \sum_{j \in g} \chi_j^g \\
&= \frac{\chi_g}{\gamma_{\hat{j}} r_{\hat{j}}}
\end{aligned}$$

Because the coding-set g is a valid coding-set, *i.e.* $\chi_g \geq \gamma_{\hat{j}} r_{\hat{j}}$, we have

$$\sum_{j \in g} \frac{\chi_j^g}{\gamma_j r_j} \geq 1$$

Therefore,

$$\lambda_i^g \geq \gamma_i r_i$$

is proven. □

More specifically, Theorem 4.11 implies that in each scheduling, transmitting the coded frame does improve the goodput for every station, compared to transmitting its native frame alone.

4.5.1 NC-Fairness

In order to prove XORR is an NC-fairness scheduler, we first present the supporting Lemma 4.12 and Lemma 4.13. Then NC-fairness property in XORR is proven in Theorem 4.14. We make some assumptions for our system. Assume our system is stationary process, i.e. the distributions of γ_i^t, r_i^t and L_i^t do not change as time elapses, **s.t.**

$$\begin{aligned} E[\gamma_i] &= E[\gamma_i^t], \\ E[r_i] &= E[r_i^t], \\ E[L_i] &= E[L_i^t], \forall t. \end{aligned}$$

We further assume that γ_i^t, r_i^t and L_i^t are independent random variables.

Lemma 4.12. *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 there 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. Similarly, for the fair scheduler $\mathcal{N}$, we have

$$\frac{t_2 - t_1}{|U|} - \theta^{\mathcal{N}} \leq \alpha_i^{\mathcal{N}}(t_1, t_2) \leq \frac{t_2 - t_1}{|U|} + \theta^{\mathcal{N}}.$$

Combining above two bounds, the lemma is proven. □

Lemma 4.13. *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*

$$\lambda_i \geq \gamma_i r_i,$$

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 the station u_i in (t_1, t_2) in the scheduler with and without coding, respectively. Then, the average goodput in (t_1, t_2) is the summation of the received bytes divided by the time duration $(t_2 - t_1)$:

$$\lambda_i^{\mathcal{L}}(t_1, t_2) = \frac{\sum_{t \in Q_i^{\mathcal{L}}} \chi_i^g(t) \cdot T_g^t}{t_2 - t_1}.$$

Hence the expected goodput for the NC scheduler is

$$
\begin{aligned}
E[\lambda_i^{\mathcal{L}}(t_1,t_2)] &= \frac{\sum_{t\in Q_i^{\mathcal{L}}} E[\chi_i^g(t)\cdot T_g^t]}{t_2-t_1} \\
&= \frac{\sum_{t\in Q_i^{\mathcal{L}}} E[L_i^t\cdot \gamma_i^t \cdot D_i^g]}{t_2-t_1} \\
&= \frac{E[L_i]E[\gamma_i]\sum_{t\in Q_i^{\mathcal{L}}} E[D_i^g(t)]}{t_2-t_1}
\end{aligned}
$$

Similarly, the expected goodput for the non-NC scheduler is

$$
\begin{aligned}
E[\lambda_i^{\mathcal{N}}(t_1,t_2)] &= \frac{\sum_{t\in Q_i^{\mathcal{N}}} E[\gamma_i^t L_i^t]}{t_2-t_1} \\
&= \frac{\sum_{t\in Q_i^{\mathcal{N}}} E[\gamma_i]E[L_i]}{t_2-t_1}.
\end{aligned}
$$

On the other hand, because $\lambda_i^t \geq \gamma_i^t r_i^t$ and $\lambda_i^t = \frac{\chi_i^g(t)T_g^t}{\mathcal{A}^t(g,i)}$, we have

$$
\mathcal{A}^t(g,i) \leq \frac{\chi_i^g(t)T_g}{\gamma_i^t r_i^t} = \frac{L_i^t D_i^g(t)}{r_i^t}.
$$

Therefore, the overall service time for the station u_i during time interval (t_1,t_2) is

$$
\begin{aligned}
\alpha_i^{\mathcal{L}}(t_1,t_2) &= \sum_{t\in Q_i^{\mathcal{L}}} \mathcal{A}^t(g,i) \\
&= E\left[\sum_{t\in Q_i^{\mathcal{L}}} \mathcal{A}^t(g,i)\right] \\
&\leq \sum_{t\in Q_i^{\mathcal{L}}} E\left[\frac{L_i^t D_i^g(t)}{r_i^t}\right] \\
&= \frac{E[L_i]}{E[r_i]} \times \sum_{t\in Q_i^{\mathcal{L}}} E[D_i^g(t)]
\end{aligned}
$$

By applying Lemma 4.12 to the above equation, we get

$$\begin{aligned}\frac{E[L_i]}{E[r_i]}\sum_{t\in Q_i^{\mathcal{L}}} E[D_i^g(t)] &\geq \alpha_i^{\mathcal{L}}(t_1,t_2)\\ &\geq \alpha_i^{\mathcal{N}}(t_1,t_2)-(\theta^{\mathcal{L}}+\theta^{\mathcal{N}})\\ &= \sum_{t\in Q_i^{\mathcal{N}}} E\left[\frac{L_i^t}{r_i^t}\right]-(\theta^{\mathcal{L}}+\theta^{\mathcal{N}})\end{aligned}$$

Therefore, we have

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

Hence, the lemma is proven. □

Theorem 4.14. *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. (4.11) is applied to XORR, XORR achieves NC-fairness, i.e.*

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

Proof. For achieving NC-fairness, two conditions have to be satisfied: 1) fairness guarantee; 2) no station has coding loss. It has been proven in Theorem 4.7 that XORR achieves temporal fairness guarantee. Thus, the first condition is satisfied.

As shown in Theorem 4.11 that $\lambda_i \geq r_i\gamma_i$ if the service time in XORR is assigned according to Equation (4.11). Consequently, the results of Lemma 4.13 can be applied to XORR coding scheme. Therefore,

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

which satisfies the second condition. Consequently, XORR can achieve NC-fairness. □

4.6 Discussion

4.6.1 Link Reliability Estimation

XORR relies on reception estimation to select coding-sets. To estimate the reception of native frame for each user, the AP needs to estimate the link reliability. Many existing wireless systems already maintain such statistics (*e.g.* WLAN [100]) for rate adaptation purposes [101, 102, 103, 104, 105, 69, 106, 107, 108, 109, 110, 100]. Some of these proposals have even been used in real products [101, 103, 104]. The link quality estimation is typically achieved by using a few metrics collected at the sender and the associated design rules. The widely used metrics include probe packets [101, 102, 104], consecutive successes/losses [101, 102, 108], short-term loss ratio [100], and long-term statistics [103]. Most practical algorithms [101, 102, 103, 104, 108, 106, 100] estimate the link quality without introducing extra signaling overhead. XORR can leverages the statistics from these schemes for link estimation.

4.6.2 Weighted Fairness

Because of different QoS requirements for different end hosts, scheduling should also provide *weighted fairness*, wherein flows with larger weights receive correspondingly better service in accordance with a system-wide fairness model. To this end, the goal of this section is to formally investigate the weighted fairness in XORR. Consider every station u_i in the network is assigned a weight ω_i, $\omega_i > 0$. We say the scheduling can achieve weighted fairness if the normalized service time during (t_1, t_2) for each station, $\frac{\alpha_i(t_1,t_2)}{\omega_i}$, is fair.

For supporting weighted fairness, the scheduling discipline and the credit update are changed accordingly. We first define the weight ω_g for the coding-set g:

$$\omega_g = \omega_I,$$

where $I = \arg\max_{i\in g} \Delta_i$. Then, XORR scheduler is defined as follows:

$$\hat{g}^t = \arg\max_g \left(\frac{U_g^t - \Delta_g^t}{\omega_g} \right), \tag{4.12}$$

where U_g^t is the utility of transmitting the coding-set g.

Following the scheduling decisions, all backlogged stations update their credits as described in Fig. 4.6.

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

FIGURE 4.6: Pseudo-code for updating credits with assigned weight.

Once a set g is selected and the coded frame is transmitted, all the stations in g decrease their credits by the fraction of resource 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.

Now we formally discuss the fairness bound of XORR fair scheduling for supporting weighted fairness. In order to provide the bound, we first present two results in Lemma 4.15 and 4.16. We then provide the weighted service time discrepancy of any two stations in Theorem 4.17

Lemma 4.15. *For any flow i and for any schedule time t, the credit counter value is always bounded by*

$$0 \leq K_i \leq \max_t \frac{L_i^t}{r_i^t} + \frac{\omega_i}{\omega_{min}} U_{max}, \tag{4.13}$$

where U_{max} is the maximal value of utility function and ω_{min} is the minimal value of weight.

Proof. For the left part of the inequality, according to the credit update in Figure 4.3, the credit value is non-negative. Thus, $K_i \geq 0$.

For the right part of the inequality, the proof is separated into two cases according to whether or not user i is in the coding group g^t. In each case, there are two sub-cases regarding how the credit is updated.

Case I: $i \notin g^t$. If $\Delta_g^t > 0$, then from Equation (4.12) we have

$$\frac{U_i^t - (T_i^t - K_i^t)}{\omega_i} \leq \frac{U_g^t - \Delta_g^t}{\omega_g}.$$

According to the credit update, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t + \frac{\omega_i}{\omega_g}\Delta_g^t \\ &\leq T_i^t + \frac{\omega_i}{\omega_g}U_g^t - U_i^t \\ &\leq \max_t \frac{L_i^t}{r_i^t} + \frac{\omega_i}{\omega_{min}}U_{max}. \end{aligned}$$

On the other hand, if $\Delta_g^t \leq 0$, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t \\ &\leq \max_t \frac{L_i^t}{r_i^t} + \frac{\omega_i}{\omega_{min}}U_{max}. \end{aligned}$$

Case II: $i \in g^t$. If $\Delta_g^t > 0$, according to the credit update, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t - \mathcal{A}(g,j) + \frac{\omega_i}{\omega_g}\Delta_g^t \\ &\leq T_i^t + \frac{\omega_i}{\omega_g}U_g^t - U_i^t - \mathcal{A}(g,j) \\ &\leq \max_t \frac{L_i^t}{r_i^t} + \frac{\omega_i}{\omega_{min}}U_{max}. \end{aligned}$$

On the other hand, if $\Delta_g^t \leq 0$, we have

$$\begin{aligned} K_i^{t+1} &= K_i^t - \mathcal{A}(g,j) \\ &\leq K_i^t \\ &\leq \max_t \frac{L_i^t}{r_i^t} + \frac{\omega_i}{\omega_{min}}U_{max}. \end{aligned}$$

Considering both cases, Lemma 4.15 holds. □

Lemma 4.16. *Assume coding-sets $(g^{t_1}, \cdots, g^{t_2-1})$ are transmitted during time period $[t_1, t_2)$. For any flow i continuously backlogged during $[t_1, t_2)$, its received* *service time $\alpha(t_1, t_2)$ during $[t_1, t_2)$ can be expressed as:*

$$\alpha(t_1, t_2) = K_i^{t_1} - K_i^{t_2} + \sum_{t=t_1}^{t_2-1} \max\left(0, \Delta_g^t\right) \times \frac{\omega_i}{\omega_g} \tag{4.14}$$

Proof. If no code set including flow i is transmitted during (t_1, t_2), there is no service time for flow i, i.e. $\alpha_i(t_1, t_2) = 0$. According to the credit update, the equality holds. If coding-set g^t including flow i is transmitted in (t_1, t_2), there are two cases:

1. If $\Delta_g^t > 0$, we have $K_i^{t+1} = K_i^t - \Delta_i^t + \frac{\omega_i}{\omega_g}\Delta_g^t$. Thus

$$\begin{aligned} K_i^t + \max\left(0, \Delta_g^t\right) &= (K_i^{t+1} + \mathcal{A}(g, j) - \frac{\omega_i}{\omega_g}\Delta_g^t) + \frac{\omega_i}{\omega_g}\Delta_g^t \\ &= \mathcal{A}(g, j) + K_i^{t+1}. \end{aligned}$$

2. If $\Delta_g^t \leq 0$, we have $K_i^{t+1} = K_i^t - \mathcal{A}(g, j)$. Thus,

$$K_i^t + \max\left(0, \Delta_g^t\right) \times \frac{\omega_i}{\omega_g} = \mathcal{A}(g, j) + K_i^{t+1}.$$

Considering both cases, Lemma 4.16 holds. □

The following Theorem 4.17 demonstrates that XORR scheduling discipline achieves bounded weighted temporal fairness:

Theorem 4.17 (Weighted 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*

$$\left|\frac{\alpha_i(t_1, t_2)}{\omega_i} - \frac{\alpha_j(t_1, t_2)}{\omega_j}\right| \leq \frac{1}{\omega_i} \max_t \frac{L_i^t}{r_i^t} + \frac{1}{\omega_j} \max_t \frac{L_j^t}{r_j^t} + 2\frac{U_{max}}{\omega_{min}}, \tag{4.15}$$

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.

Proof. Based on Lemma 4.15 and Lemma 4.16, the service discrepancy can be derived as following

$$\begin{aligned}
\left|\frac{\alpha_j(t_1,t_2)}{\omega_j}-\frac{\alpha_j(t_1,t_2)}{\omega_j}\right| &= \left|\frac{K_i^{t_1}-K_i^{t_2}}{\omega_i}-\frac{K_j^{t_1}-K_j^{t_2}}{\omega_j}\right| \\
&\leq \left|\frac{K_i^{t_1}-K_i^{t_2}}{\omega_i}\right|+\left|\frac{K_j^{t_1}-K_j^{t_2}}{\omega_j}\right| \\
&\leq \left|\frac{\max\left(K_i^{t_1},K_i^{t_2}\right)}{\omega_i}\right|+\left|\frac{\max\left(K_j^{t_1},K_j^{t_2}\right)}{\omega_j}\right| \\
&= \frac{\max_t \frac{L_i^t}{r_i^t}+\frac{\omega_i}{\omega_{min}}U_{max}}{\omega_i}+\frac{\max_t \frac{L_j^t}{r_j^t}+\frac{\omega_j}{\omega_{min}}U_{max}}{\omega_j} \\
&= \frac{1}{\omega_i}\max_t \frac{L_i^t}{r_i^t}+\frac{1}{\omega_j}\max_t \frac{L_j^t}{r_j^t}+2\frac{U_{max}}{\omega_{min}}
\end{aligned}$$

Thus Theorem 4.17 is proven. □

4.6.3 Two-Way Traffic

XORR is also applicable with two-way traffic to further reduce the retransmissions for both up- and down-link traffic, assuming nodes in the network can overhear each other. For example, as shown in Figure 4.7, u_i transmits uplink frame p_i^u to the AP and the AP transmits downlink frame p_j to u_j. Both frames are lost. Assume the uplink frame p_i^u is overheard by u_j and the downlink frame p_j is overheard by u_i. When u_i retransmits an up-link frame p_i^u , it can apply XORR to recover another lost down-link frame p_j by transmitting $p_i^u \oplus p_j$. If u_j and the AP receive the coded frame, they can decode their intended frames from the coded frame. Note that the AP can always decode the uplink frame as it already has all downlink frames. Scheduling two-way traffic follows the same scheduling policy as presented.

4.6.4 Cooperation with TCP Congestion Control

Most applications run on top of TCP. Hence, it is essential that the coding scheme does not impose any adverse impact on TCP performance. Two issues are particularly relevant: 1) loss recovery, and 2) packet reordering. First, TCP interprets

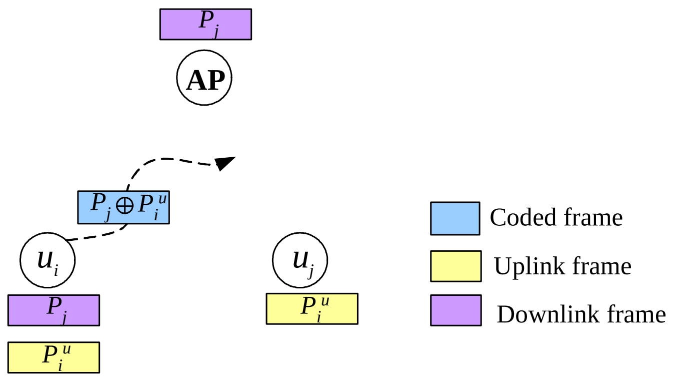


FIGURE 4.7: Example of using XORR for up- and down-link traffic.

a packet loss as a signal of congestion to which it reacts by halving the transmission rate. Since wireless links usually have higher error rates than what TCP can handle, an efficient link-layer retransmission scheme is demanded for improving TCP performance in wireless network. Our XORR exploits network coding for link-layer retransmissions and thus is more resilient to the errors compared to the traditional 802.11 MAC retransmissions. Therefore, owing to providing better wireless link quality, XORR improves TCP performance.

Second, since TCP relies on the packet sequence numbers to detect losses, it may confuse packet reordering as a sign of congestion. The reordered packets may trigger TCP congestion backoff, resulting in low throughput. Hence, a proper coding scheme should prevent packet reordering problem. The AP with XORR only schedules HOL frames. Therefore, there is no frame-reordering caused by XORR.

4.6.5 Overhead of Network Coding

The benefit that one can achieve with network coding comes at the expense of more battery and CPU time consumption for encoding and decoding the information. Traditional network coding uses operations over large finite fields. Decoding operations have quadratic complexity, which becomes too slow for high throughput applications. Further encoding operations are also complicated since they involve multiplications in large finite fields. However, XORR uses simple bitwise XOR for the encoding and decoding. Hence, the coding overhead of XORR is moderate but more coding benefits are provided by XORR.

4.7 Chapter Summary

In this chapter, we propose a practical NC-aided ARQ scheme, namely XOR Rescue (XORR). By considering various aspects, XORR provides a global approach for designing an efficient retransmission scheme:

1. A Bayesian-learning-based reception estimation is designed, which accurately estimate the reception status without causing extra signaling overheads.
2. A new coding metric is devised for accommodating the heterogeneities in wireless environments.
3. A framework of a NC-aware scheduler is designed to handle the coding-set selection with the objective of maximizing the system goodput as well as maintaining resource fairness.
4. A novel NC-fair assignment algorithm is devised theoretically proven for ensuring the NC-fairness, i.e. guaranteeing temporal fairness and no coding loss in all stations.

XORR can be further extended for supporting weighted fairness and using in the two way traffic (uplink and downlink).

Chapter 5

Theoretical Analysis

XORR attempts to exploit the broadcast capacity of wireless networks with network coding for providing efficient retransmissions. After presenting the detailed design of XORR, in this chapter, we analyze the theoretical bounds on the XORR's performance. The assumption and the basic concept of the proof are described in Section 5.1. The theoretical bound for coding-set size and the coding gain are discussed in Section 5.2 and Section 5.3, respectively. Finally, the numerical results are provided in Section 5.4.

5.1 Assumption and Basic Concept

The expected performance gain of XORR is characterized in this section. Our main focus is to derive the bounds for XORR's performance improvement. A simple model is considered, where all stations in the network have the same transmission rate r and the same link reliability γ. Assume that all frames have the same size and all N stations are always backlogged. Assume that the AP has the perfect knowledge on reception status. Hence the AP only selects a coding-set, which is decodable at every targeted station when the station receives it.

Let Q_t and Q_r denote the number of the frames in *TxGroup* and *RetxGroup*, respectively. The AP starts to recover lost frames when the number of frames in *RxGroup* reaches a threshold δ,*i.e.* $Q_r = \delta$; otherwise, the AP transmits original frames. Since the AP only transmits HOL frames for N backlogged stations, we have

$$\begin{aligned} Q_t + Q_r &= N, \\ \delta &\leq N. \end{aligned}$$

Figure 5.1 shows a retransmission period between two retransmissions. When $Q_r = \delta$, the AP starts to retransmit (either a coded or a native frame is selected from *RetxGroup*) and later receives α_r ACKs. Upon receiving α_r ACKs, the AP removes the ACKed frames from *RetxGroup* and thus $Q_r = \delta - \alpha_r$. Since $Q_r < \delta$ now, the AP starts to transmit original frames till $Q_r = \delta$ again. Assume X original frames are transmitted and α_t ACKs are consequently received. Thus, the total number of transmissions is $1 + X$ and the total number of ACKs is $\alpha_r + \alpha_t$. Then the expected goodput can be easily calculated as

$$\lambda \simeq \frac{1+X}{\alpha_r + \alpha_t} \cdot r$$

α_r, α_t and X are related the size of the retransmitted coding-set(*i.e.* the number of native frames XORed in a retransmitted frame). Once the average coding-set size is calculated, the expected goodput of XORR can be derived. Therefore, the upper and lower bound of average coding-set size will be conducted in the next section.

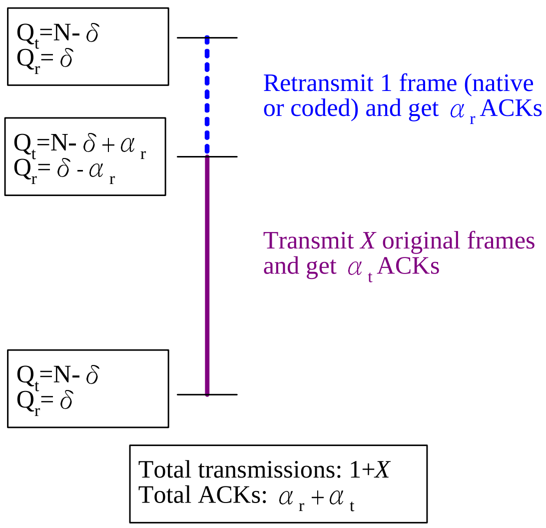


FIGURE 5.1: One retransmission period

5.2 Coding-set Size

Let $\mathbf{K}$ denote the expected coding size. An upper and a lower bound on the expected *coding-set* size, $\mathbf{K}$, is presented in Theorem 5.1.

Theorem 5.1. *The expected coding-set size* $\mathbf{K}$ *satisfies,*

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

Proof. Assume $\hat{g}$ is the selected decodable set. Then the average size of the set is

$$\begin{aligned}\mathbf{K} &= \sum_{\kappa=1}^{\delta} \kappa \cdot Pr(|\hat{g}|=\kappa) \\ &= \sum_{\kappa=1}^{\delta} Pr(|\hat{g}| \geq \kappa).\end{aligned}$$

Assume g_i^κ is a set with κ frames, where $|g_i^\kappa| = \kappa$. Let $\mathcal{D}$ denote the *decodable* event for the set if the set can be decoded by all members coded in the coding-set. Thus, the probability for the g_i^κ to be a decodable set is

$$Pr(g_i^\kappa = \mathcal{D}) = \gamma^{(\kappa-1)\kappa}.$$

Let $\overline{\mathcal{D}}$ denote the *undecodable* event. Accordingly,

$$\begin{aligned}Pr(g_i^\kappa = \overline{\mathcal{D}}) &= 1 - Pr(g_i^\kappa = \mathcal{D}) \\ &= 1 - \gamma^{(\kappa-1)\kappa}.\end{aligned}$$

Note that g_i^κ may be part of larger set with more than κ frames. Thus, we have

$$\begin{aligned}Pr(|\hat{g}| \geq \kappa) &= Pr\left(\bigcup_{i=1}^{m} (g_i^\kappa = \mathcal{D})\right) \\ &= 1 - Pr\left(\bigcap_{i=1}^{m} (g_i^\kappa = \overline{\mathcal{D}})\right),\end{aligned}$$

where m is total number of sets with κ frames, $m = \binom{\delta}{\kappa}$. Since there are overlaps among undecodable sets, the joint probability can be bounded as

$$\begin{aligned} Pr\left(\bigcap_{i=1}^{m}(g_i^\kappa = \overline{\mathcal{D}})\right) &\geq \prod_{i=1}^{m} Pr(g_i^\kappa = \overline{\mathcal{D}}) \\ &= (1-\gamma^{(\kappa-1)\kappa})^m. \end{aligned}$$

Thus the upper bound is proven.

Assume there is an inefficient XORR scheme that the AP groups δ frames into several coding-sets so that there is no overlap among the grouped sets, i.e. there are $\lfloor \frac{\delta}{\kappa} \rfloor$ sets, and κ frames in each set, where $\kappa = 1, 2, \cdots, \delta$. The AP later selects a decodable set among those un-overlapped sets as a coding set.

Assume $\mathbf{K_{in}}$ is the expected size of the coding-set in the inefficient XORR scheme. Due to less coding opportunities caused by restricted set grouping, it can be inferred that the expected coding-set size in XORR is larger than that in inefficient scheme, *i.e.* $\mathbf{K} \geq \mathbf{K_{in}}$.

Since there is no overlap among sets in the inefficient scheme, its joint probability is

$$Pr\left(\bigcap_{i=1}^{m'}(g_i^\kappa = \overline{\mathcal{D}})\right) = (1-\gamma^{(\kappa-1)\kappa})^{m'},$$

where $m' = \lfloor \frac{\delta}{\kappa} \rfloor$. Therefore,

$$\begin{aligned} \mathbf{K_{in}} &= \sum_{\kappa=1}^{\delta} 1 - Pr\left(\bigcap_{i=1}^{m'}(g_i^\kappa = \overline{\mathcal{D}})\right) \\ &= \sum_{\kappa=1}^{\delta} 1 - (1-\gamma^{(\kappa-1)\kappa})^{m'}. \end{aligned}$$

Thus, the lower bound is proven. $\square$

As shown in Lemma 5.1, 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 during loss-recovery. 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 networks, the threshold can be adjusted to higher value for the sake of more coding opportunities.

5.3 Coding Gain

The *coding gain* is defined as the ratio of the goodput achieved by XORR to that by the non-NC approaches. Theorem 5.2 characterizes the coding gain of XORR.

Theorem 5.2. *The coding gain of XORR is* $B = \frac{\mathbf{K}}{1-\gamma+\gamma\mathbf{K}}$.

Proof. After transmitting 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 *RetxGroup*, there are α_r vacancies in *RetxGroup*. Consequently, the AP transmits original frames until the number of *RetxGroup*, Q_r, reaches δ again. Thus, the average number of transmitted original frames for α_r frames is

$$\begin{aligned} X &= \alpha_r \times \sum_{i=1}^{\infty} \gamma^{i-1}(1-\gamma) \cdot i \\ &= \frac{\gamma \mathbf{K}}{1-\gamma}. \end{aligned}$$

Since α_r frames are not ACKed after sending X original frames, it can be inferred that the number of consequent ACKs is

$$\alpha_t = X - \alpha_r = \frac{\gamma^2 \mathbf{K}}{1-\gamma}.$$

Therefore the expected goodput of XORR is

$$\begin{aligned} \lambda^{\text{XORR}} &\simeq \frac{1+X}{\alpha_r + \alpha_t} \cdot r \\ &= \frac{\gamma \mathbf{K}}{1-\gamma+\gamma\mathbf{K}} \cdot r. \end{aligned}$$

Furthermore, the expected goodput for non-NC scheme can be calculated if $\mathbf{K} = 1$ is set in the previous equation. Hence, the expected goodput for non-NC scheme is

$$\lambda^{\text{non-NC}} = \gamma \cdot r.$$

Accordingly, the coding gain of XORR is

$$\begin{aligned} B &= \frac{\lambda^{\text{XORR}}}{\lambda^{\text{non-NC}}} \\ &= \frac{\mathbf{K}}{1-\gamma+\gamma\mathbf{K}}. \end{aligned}$$

□

5.4 Numerical Results

According to Theorem 5.1 and Theorem 5.2, we calculate the coding-set size and XORR coding gain compared to non-NC approach by changing the retransmission threshold and link reliability. Furthermore, we show the coding gain under different number of stations in the network.

- **Varying the retransmission threshold δ**

Assume there are an AP and 100 stations in the network. Figure 5.2 and Figure 5.3 show the upper and lower bound of average coding-set size and coding gain with varying the retransmission threshold δ under different settings of link reliabilities γ, respectively. The maximal value of the threshold is 100 (i.e. the number of stations). We make the following observations.

First, the coding-set size increases with the retransmission threshold δ, which suggests that the larger retransmission threshold results in more opportunities to encode more frames in one coded frame. This is because a larger number of queued retransmitted frames makes it easier to find frames that are overheard by different stations and thus creates more coding opportunities. Accordingly, the coding gain increases with the retransmission threshold δ, as shown in Figure 5.3.

Second, coding based retransmissions results in more coding gain in a lower link reliability network (e.g. $\gamma = 0.3$ in Figure 5.3) than in a higher link reliability network, especially with larger retransmission threshold. This demonstrates that the lossier environments can benefit more coding gain from network coding because more lost frames can be recovered with the help of coding.

- **Varying link reliability γ**

Next we evaluate the performance by varying the link reliability. Assume there are an AP and 100 stations in the network. Figure 5.4 and Figure 5.5 show the upper

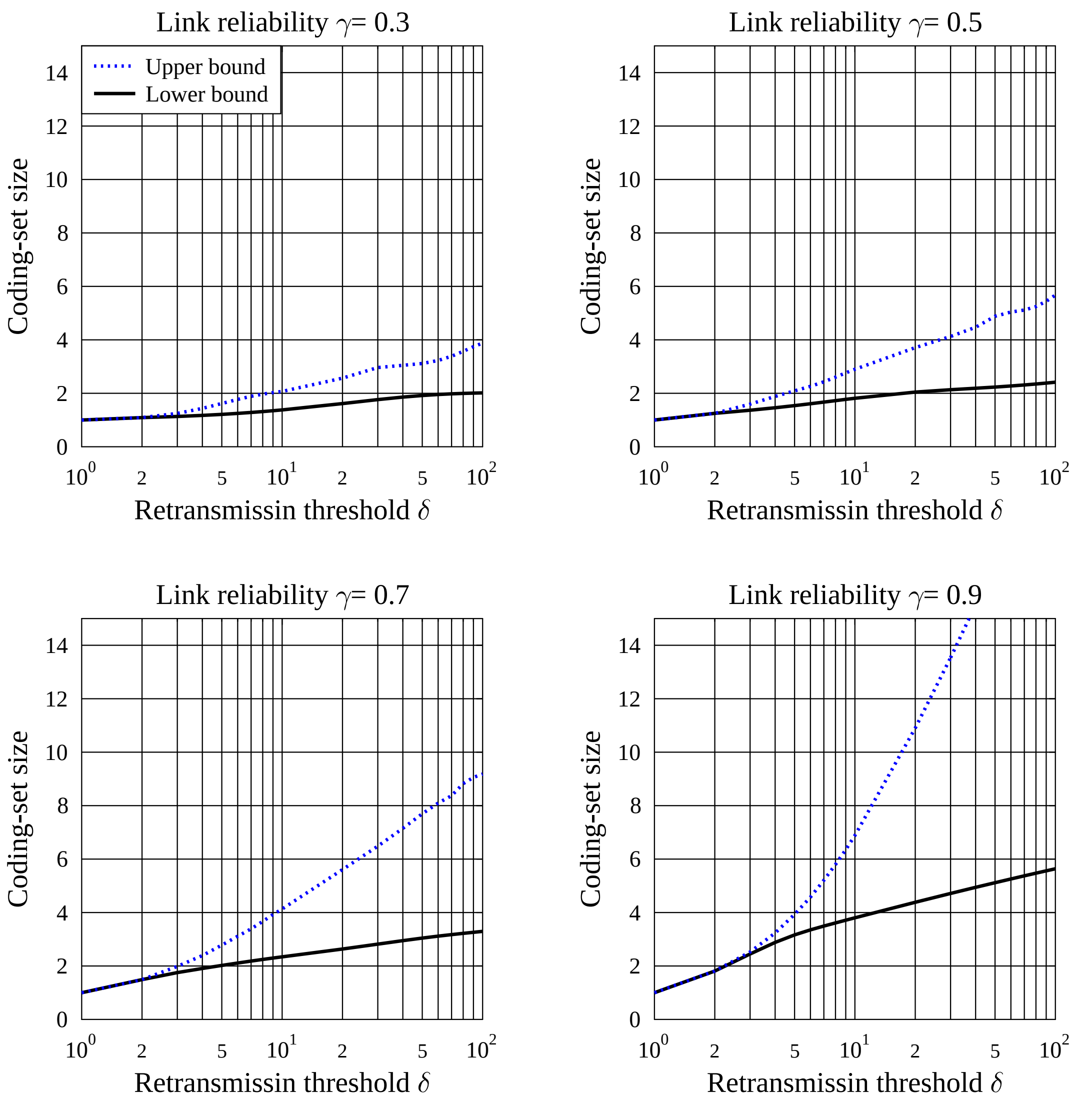


FIGURE 5.2: Theoretical upper and lower bound of average coding-set size with varying retransmission threshold δ. Assume there are 100 stations in the network.

and lower bound of average coding-set size and coding gain with varying the link reliability γ under different settings of retransmission threshold δ, respectively. As depicted in Figure 5.4, the coding-set size increases with the link reliability. This is because with the higher link reliability, a lost frames is more likely to be overheard by more stations at the same time and thus more lost frames could be encoded as a decodable set. Consequently, the coding-set size is larger in higher link reliability. However, interestingly, the coding gain decreases with the link reliability, as demonstrated in Figure 5.5. This is because with the decrease of the reliability, more frames are lost and need to be retransmitted. XORR thus

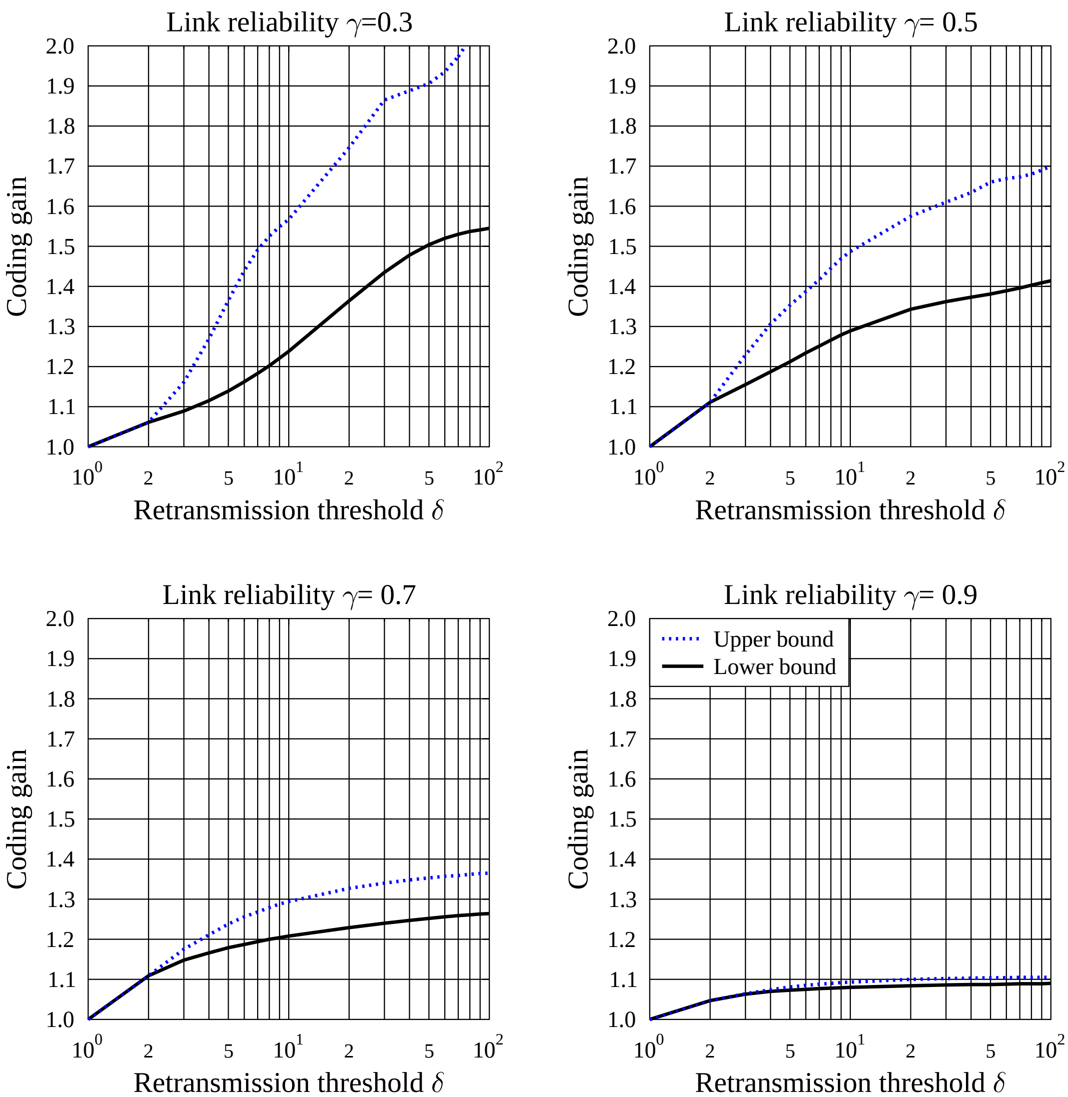


FIGURE 5.3: Theoretical upper and lower bound of coding gain with varying retransmission threshold δ. Assume there are 100 stations in the network.

increasingly improves the network throughput by reducing more retransmissions.

- **The impact of the number of stations**

As shown in previous results, a better scheduler can defer multiple frame retransmissions and accumulate them (e.g. setting larger retransmission threshold), so that when doing loss-recovery, it can potentially encode more frames into one retransmission. In order to investigate the impact of the number of stations, we set that the retransmission threshold is always equal to the number of stations in the network. In other words, we can always get the maximal coding gain under certain number of stations. Table 5.1 gives some numerical results of the lower and upper

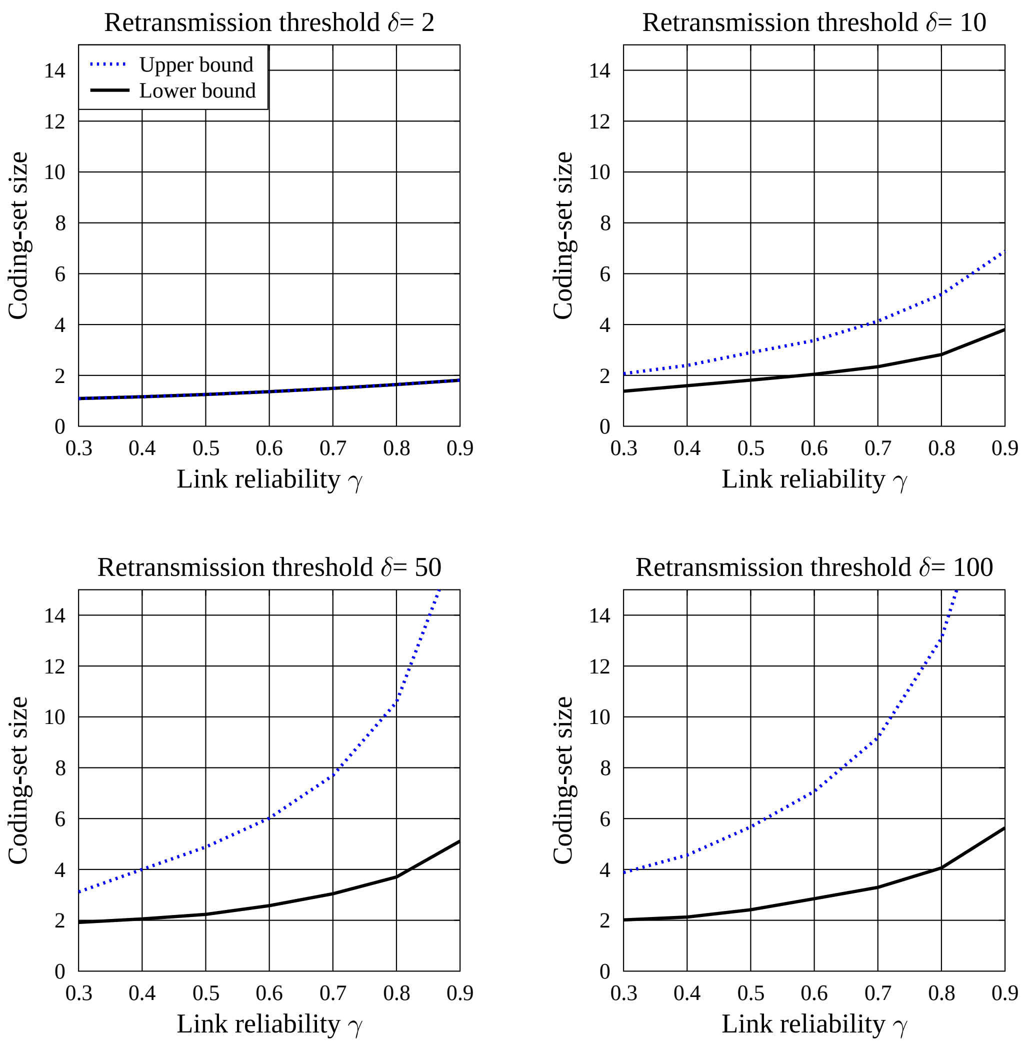


FIGURE 5.4: Theoretical upper and lower bound of average coding-set size with varying link reliability γ. Assume there are 100 stations in the network.

bounds of XORR's coding gain with respect to different numbers of stations N in the network. We can see with a moderate number of N, XORR can effectively reduce retransmissions and thus improve the system performance.

5.5 Chapter Summary

We have theoretically characterized the potential coding gain of XORR in this Chapter. The numerical results shows that the larger retransmission threshold results in the chance of encoding more frames in coded frames. Hence, the coding

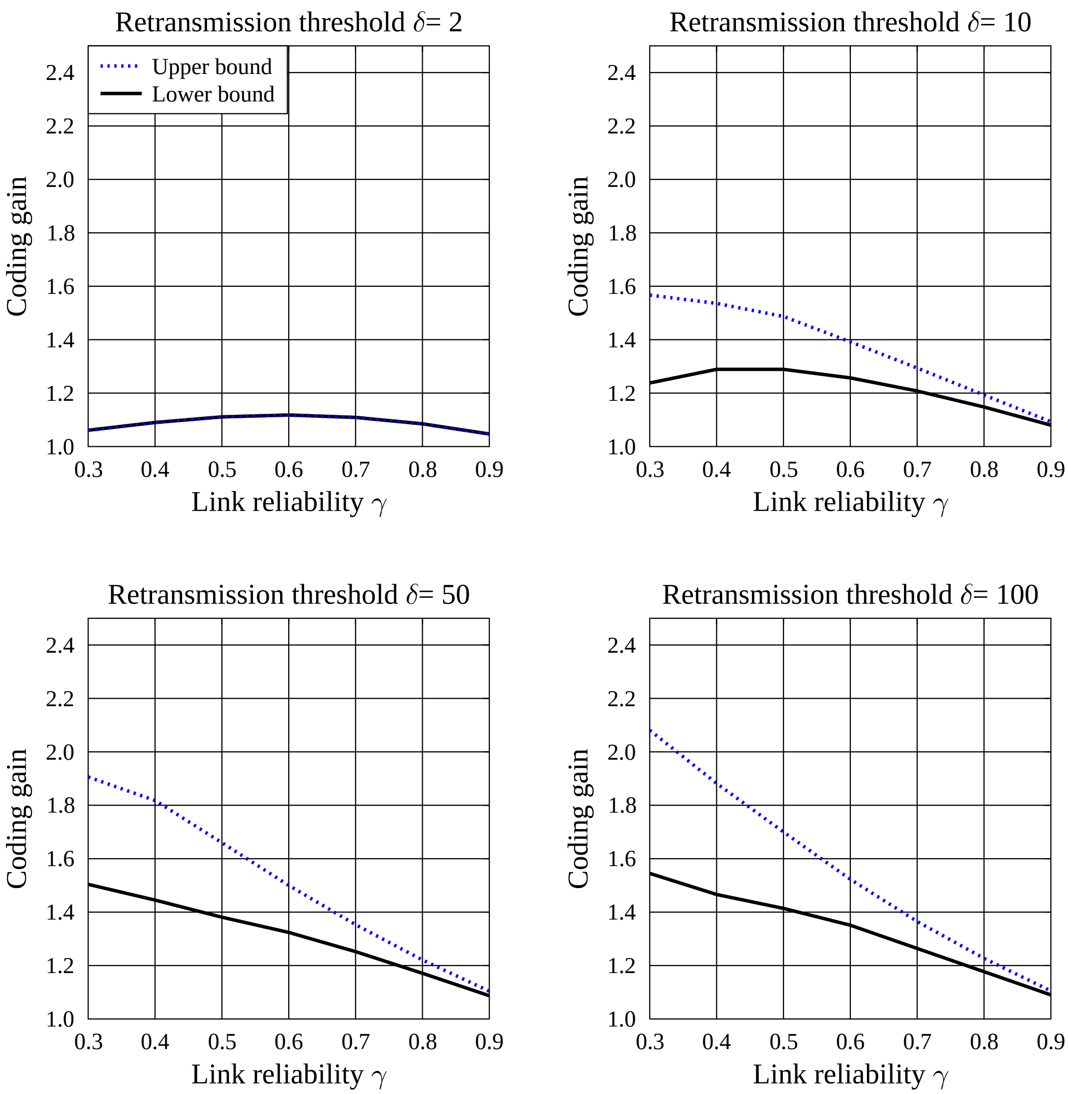


FIGURE 5.5: Theoretical upper and lower bound of coding gain with varying link reliability γ. Assume there are 100 stations in the network.

gain increases with the retransmission threshold. This suggests that the XORR should defer the retransmissions moderately for creating more coding gains. Furthermore, the results also show that with a moderate number of N (e.g. $N = 100$), XORR is capable of effectively enhancing system performance.

γ	$N = 10$		$N = 100$		$N \rightarrow \infty$	
	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 5.1: Numerical results of coding gain when the number of stations N is 10, 100 and infinity, respectively. The retransmission threshold δ is always set as the number of stations, i.e. $\delta = N$.

Chapter 6

Performance Evaluation

In this chapter, XORR is evaluated by simulations and experiments. The simulation setup, benchmarks, and performance metrics are defined in Section 6.1. The effects of two tunable parameters, utility scaling factor β and deferring retransmission factor θ, in XORR are demonstrated in Section 6.2. The effectiveness of our heuristic algorithm is verified in Section 6.3. In Section 6.4, the throughput performance of XORR is evaluated under different wireless link models. The effect of the NC-fair assignment algorithm and the weighted fairness are manifested in Section 6.5. The impact of reliability estimation error and the performance of delay are shown in Section 6.6 and Section 6.7, respectively.Finally, we preliminarily evaluate XORR's performance on real wireless test-bed in Section 6.8.

6.1 Simulation Setup

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. 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. Furthermore, the utility scaling factor β is set as $\beta = 50$ for balancing between fairness and goodput. The deferring retransmission factor θ is set as $\theta = 2$ for achieving sufficient coding opportunities. The selection of the parameters will be discussed in the Section 6.2.

We evaluate the performance of the proposed XORR under different wireless channel models. In order to explore the effect of link reliability and the number of stations, we use the simple static wireless channel, where the channel quality does not change over time. Two types of link conditions are used in the static wireless channel, namely homogeneous and heterogeneous wireless links. In homogeneous wireless links, every wireless link is assumed to be the same. In the heterogeneous wireless links, the link quality of every wireless link is different. We model homogeneous links $\gamma_i \in [\gamma_{min}, \gamma_{max}]$ by choosing the link reliability uniformly between γ_{min} and γ_{max}. Then we use more realistic time-varying channels to explore the coding benefits.

6.1.1 Benchmarks

In our simulations, XORR is compared with the following three 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 [11]. 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. As shown in Figure 3.4 in Section 3.3, when the number of the station is 10, ER with 50 ms performs best in the most of the link reliabilities. So unless otherwise mentioned we use 50 ms as the report period for ER when the number of the stations is around 10.

4. *XORR Ideal.* This is an upper bound of XORR by assuming the AP has an oracle to learn every native frame received by all stations and thus there is

no overhead of feedback frames. Furthermore, an exhaustive search is used to find the best coding-set.

6.1.2 Performance Metrics

We evaluate XORR by two parts: system performance and fairness. For qualifying system performance, we use *goodput gain* and *reduced retransmission ratio.* The baseline scheme for evaluating system performance is *802.11.* So we have

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

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

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

For examining the fairness of XORR, two metrics are introduced: *fairness index* and *coding improvement ratio.* Fairness index was introduced by Jain et al [111] and its value ranges between 0 and 1.

$$\text{Fairness Index} = \frac{\left(\sum_{i=1}^{N} \alpha_i\right)^2}{N \cdot \sum_{i=1}^{N} \alpha_i^2},$$

where α_i is the allocated service time of station u_i. 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.

6.2 Impact of Parameters

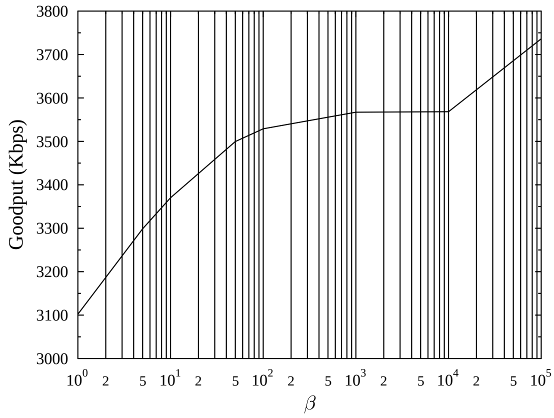
XORR has two tunable parameters: 1) utility scaling factor β and 2) deferring retransmission factor θ. β is used to balance the fairness bound and the system performance gain [60]; while θ decides how much priority should be given to original frames than retransmissions for creating more coding opportunities. The effects of varying the utility scaling factor β and the deferring retransmission factor θ in XORR will be extensively investigated in this section.

6.2.1 Utility Scaling Factor, β

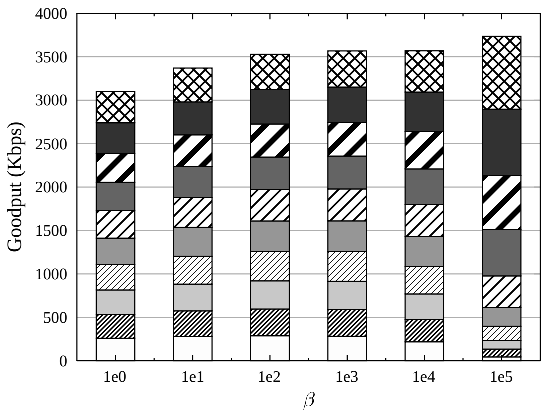
First, we study the system goodput performance for various β, the parameter modulating the utility function. The static heterogeneous link model is used. All users are transmitted using 5.5Mbps and the reliability is randomly chosen from 0.4 to 0.6. The *theta* is set to 2. Figure 6.1 shows the performance of XORR with respect to different β. Figure 6.1(c) and Figure 6.1(b) show the fairness and individual goodput under different β. The boxes in the Figure 6.1(b) are sorted by the link condition. The upper box represents the best link condition in the network.

We make three observations about Figure 6.1. First, for the case of $\beta = 1$, the flows achieve identical temporal shares of the channel (i.e. Fairness index = 1), but do not achieve identical goodput due to their different average channel conditions. The station with better channel condition achieves better goodput. Second, as expected, the network goodput is increased with β as shown in Figure 6.1(a). In other words, with larger β, the scheduler is more opportunistic by trading more fairness for performance by assigning more service time to better users. Third, while increasing β enables higher system goodput, excessive weighting of the channel condition in the scheduling criteria has starved the flows with bad conditions. Regardless, between these extremes, a wide range of β yields an effective tradeoff between goodput gain and fairness. Such tradeoff has been sufficiently discussed in literature [60].

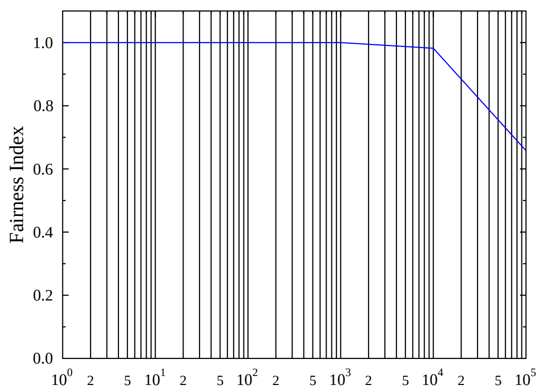
According to the simulation results, in the following evaluation, we choose $\beta = 50$ for better balancing between fairness and goodput performance.



(a) Goodput



(b) Individual Goodput



(c) Fairness index

FIGURE 6.1: Coding Performance when applying different values of β and setting $\theta = 2$.

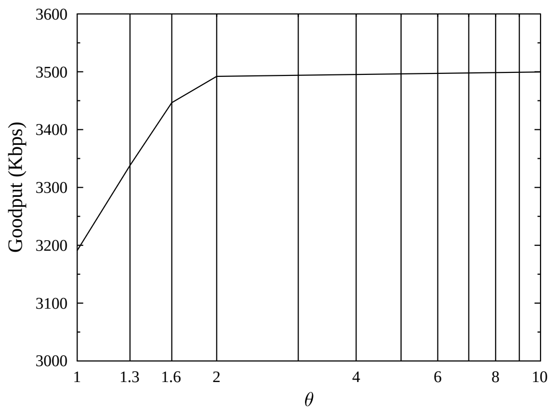
6.2.2 Deferring Retransmission Factor, θ

The static heterogeneous link model is used. All users are transmitted using 5.5Mbps and the reliability is randomly chosen from 0.4 to 0.6. The β is set to 50. Figure 6.1 shows the performance of XORR with respect to different θ. Figures 6.2(a), 6.2(b) and 6.2(c) show the goodput performance, the statistic for the size of coding-set and the fairness under different θ, respectively.

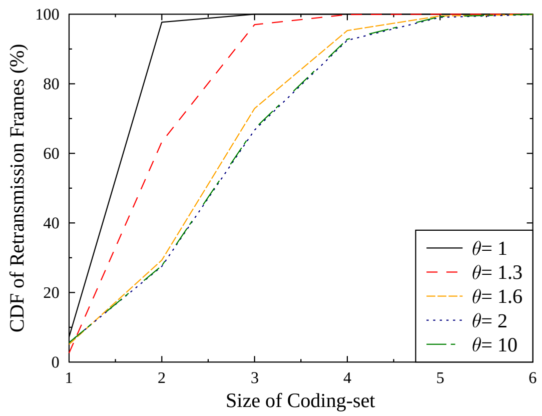
We make three observations about Figure 6.2. First, in the case of $\theta = 1$, the AP retransmits a frame whenever its utility is largest no matter it is an original frame or a coded frame. Therefore, it achieves the lowest retransmission delay. However, as shown in Figure 6.2(b), the average coding size is two when $\theta = 1$, which demonstrates that such retransmission behavior would reduce the coding opportunities. Second, Figure 6.2(a) depicts the goodput of XORR with the increase of θ. This could be explained by Figure 6.2(b). The average size of coding-set increases with θ because a larger θ gives the AP more favor to transmit original frames and defer the retransmission for potential coding opportunity. Third, as shown in Figure 6.2(a), a small θ would be enough for coding opportunity. Accordingly to the simulation results, in the following evaluation, we set $\theta = 2$ for achieving sufficient coding opportunities.

6.3 Impact of Heuristic Selection

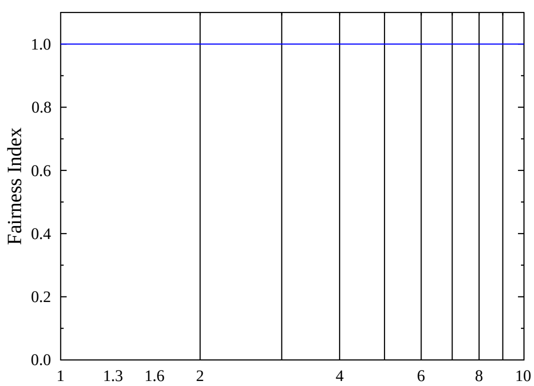
We have proven that finding the optimal coding-set is NP-hard in Theorem 4.8. Therefore, a practical heuristic coding-set selection algorithm is proposed in Section 4.4.3. In order to verify the effectiveness of our heuristic algorithm, we compare it with an exhaustive search algorithm, which is guaranteed to give an optimal solution but computationally very expensive. As shown in Figure 6.3, our heuristic algorithm is efficient because it only slightly degrades the performance but reduces drastically the complexity of the exhaustive search.



(a) Goodput



(b) The size of Coding-set



(c) Fairness index

FIGURE 6.2: Coding Performance when applying different values of θ and setting $\beta = 50$.

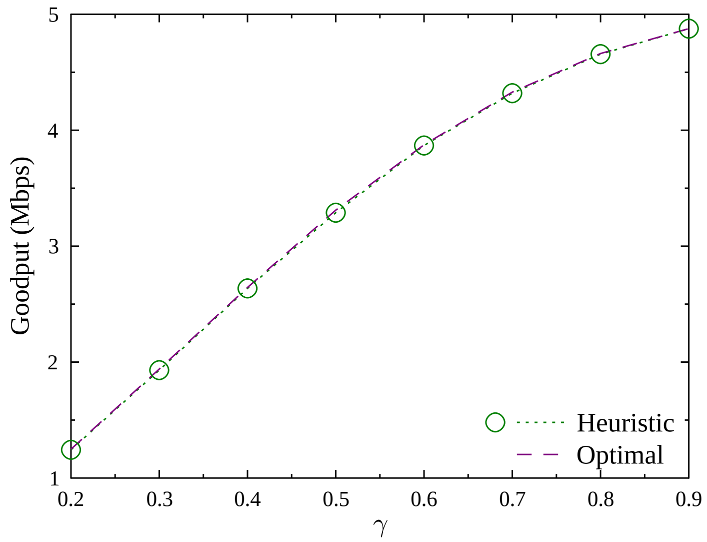


Figure 6.3: Comparing goodput performance with various reliability between heuristic and exhaustive coding-set selection.

6.4 Throughput

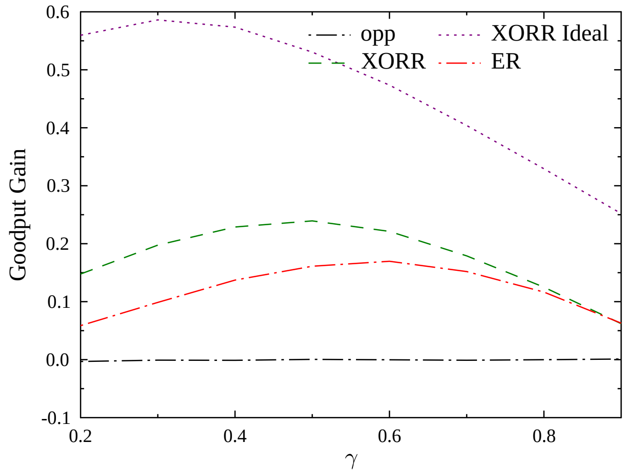
In the following simulations, we evaluate the throughput performance in different wireless link models. We first discuss the results in the static channels. A time-varying wireless link model is used for demonstrating more realistic results.

6.4.1 Static Channels

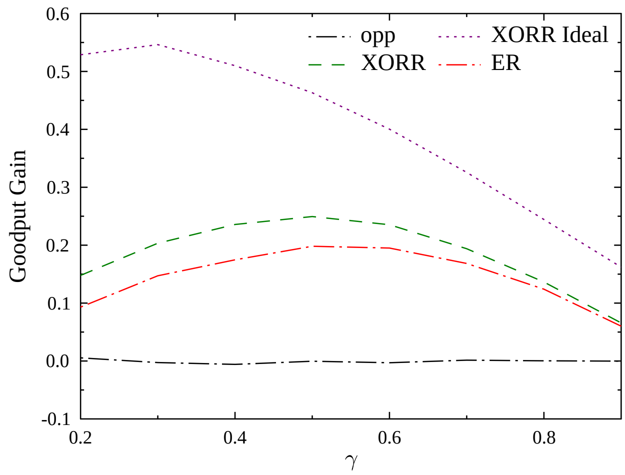
In the static wireless channels, three effects are explored, namely homogeneous link reliability, the number of stations, and heterogeneous link reliability.

6.4.1.1 Impact of Link Reliability

Figure 6.4 and Figure 6.5 show the impact of different link reliabilities on the goodput gain and the reduced retransmission ratio for XORR, XORR Ideal, *Opp*, and ER having a report period of 50 ms. All links have the same reliability, which varies from 0.2 (least reliable) to 0.9 (most reliable).



(a) $R = 11$ Mbps



(b) $R = 5.5$ Mbps

FIGURE 6.4: Goodput gain with various link reliability in static homogeneous channels.

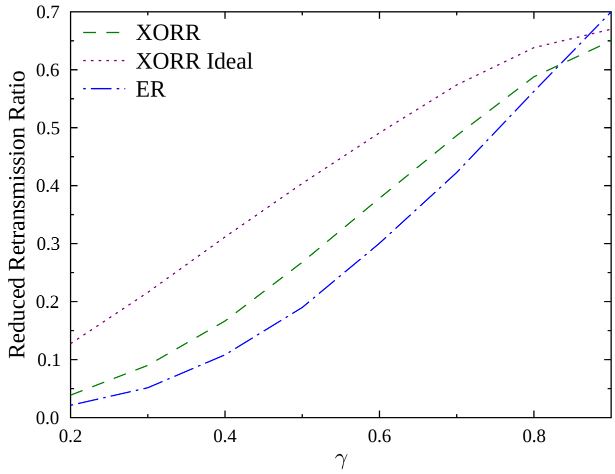
As shown in Figure 6.4, *Opp* has the same goodput as that of *802.11* in the context of homogeneous channels, because all stations have exactly the same transmission rate and reliability, and thus no multi-user diversity gain can be utilized.

Both XORR and ER employ network coding for reducing retransmissions. As shown in Figure 6.5, when the link reliability is higher than 0.8, over 60% of their retransmissions are saved. On the other hand, when the link reliability is low, their reduced retransmission ratios are also small. This is because stations receive less native frames, which results in less coding opportunities. Note that this does not conflict with the results shown in Figure 6.4 where the goodput gain does not increase with the link reliability. Due to the low goodput in low link reliability cases, a small portion of reduction on retransmission would increase the goodput significantly in percentage.

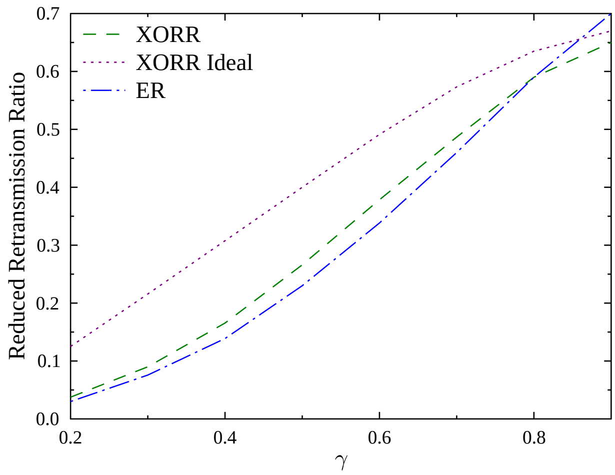
The coding efficiency of ER heavily depends on feedback information carried by reception reports as shown in Figure 3.3. By contrast, XORR estimates the reception status, which mitigates signaling overheads, thus outperforms ER. As the link reliability decreases, more native frames are lost and need to be retransmitted. XORR improves the goodput by reducing retransmissions as depicted in Figure 6.4. 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. This is 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. On the other hand, XORR Ideal using an oracle can always find the optimal coding-set, so that the goodput gain continues increasing as more frames are lost and need retransmission.

6.4.1.2 Impact of the Number of Stations

Figure 6.6 illustrates the impact of different number of stations on the goodput gain for XORR, XORR Ideal and ER having a report period of 50 ms, when the link reliability is 0.2, 0.5 or 0.8. We also plot the theoretical bounds we derived in Chapter 5. As the number of stations increases, the goodput gain also increases for both XORR and ER. This is expected since there are more coding opportunities. However, the curves of goodput gain of XORR become flat with only a moderate number of stations (e.g. 10). This implies that network coding opportunities are already significant when there are only a moderate number of stations. However,

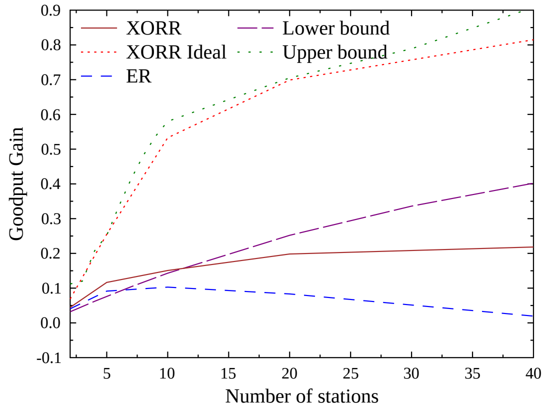


(a) $R = 11$ Mbps

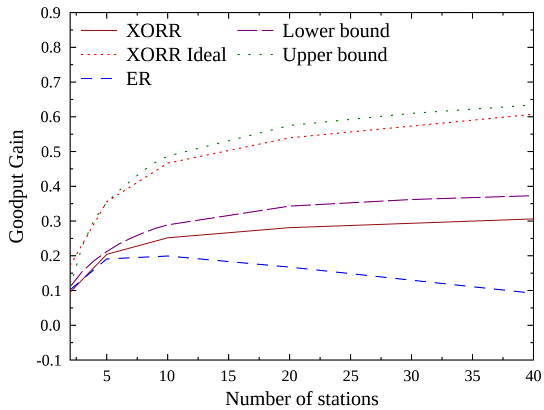


(b) $R = 5.5$ Mbps

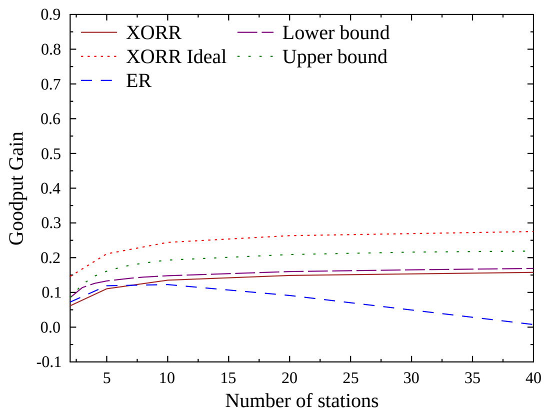
FIGURE 6.5: Reduced retransmission ratio with various link reliability in static channels.



(a) $\gamma = 20\%$



(b) $\gamma = 50\%$



(c) $\gamma = 80\%$

FIGURE 6.6: Goodput gain with different number of stations in static channels.

the goodput gain of ER bends down when the number of stations is greater than 10. This is due to overheads of reception reports. More stations introduce more feedback frames, which overwhelms the coding gain in ER.

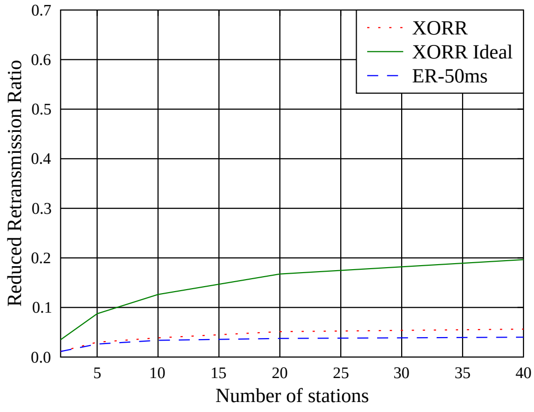
6.4.1.3 Heterogeneous Wireless Links

In Figure 6.8(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.

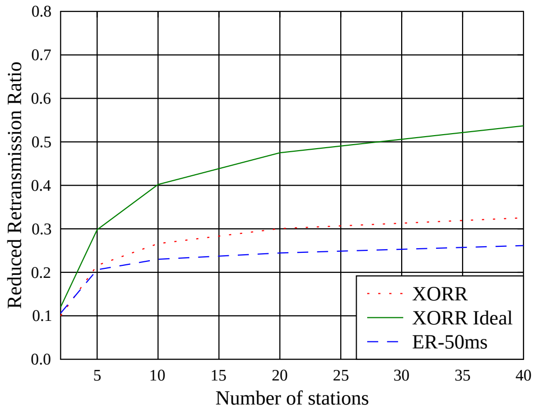
When $\Delta\gamma$ is large, the goodput of *802.11* drops dramatically, while *Opp*'s goodput remains unchanged. This is because that *802.11* allocates more channel time to stations with worse channel conditions [70], while *Opp* always allocates equal service time to all stations. Therefore, *Opp* performs better than *802.11*.

Due to 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 reliabilities. Hence, there are less effective coding opportunities. Furthermore, Figure 6.8(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 under all simulations, the calculated fairness index is close to 1. Hence, we omit it in the following results.

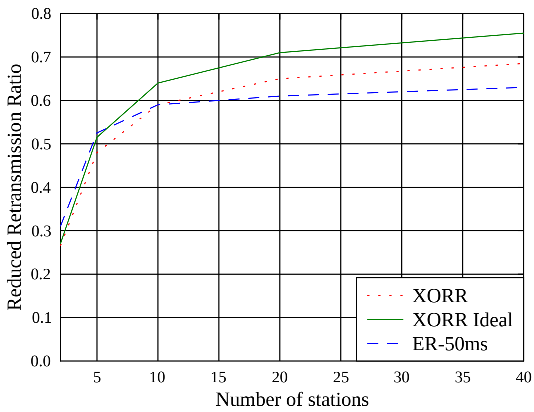
Figure 6.8(c) shows the *coding improvement ratio* for every station using XORR and the station indexes are sorted by their link reliabilities. XORR improves the individual stations' goodput by 10% to 25% compared to *Opp*. This demonstrates that XORR is an *NC-fairness* scheduler because it improves *all* stations' performance while maintaining temporal fairness (*i.e.* the calculated fairness index is 1).



(a) $\gamma = 20\%$

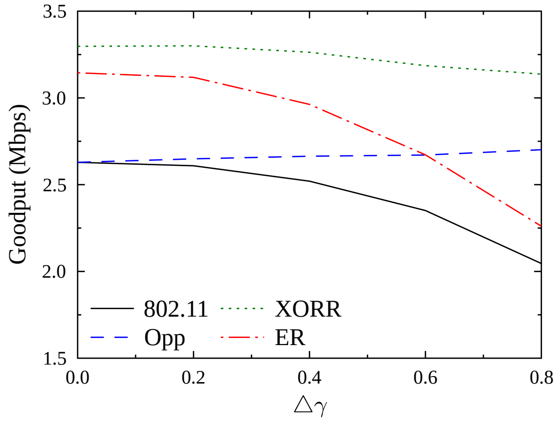


(b) $\gamma = 50\%$

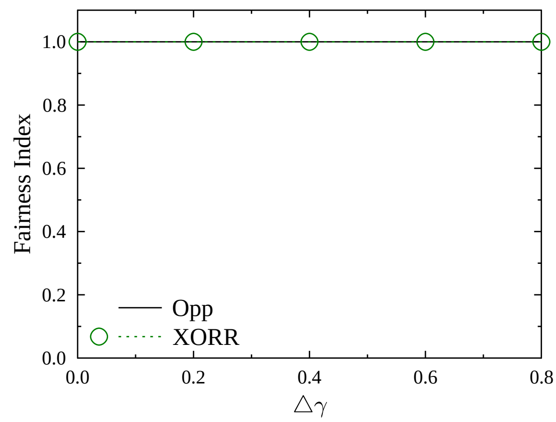


(c) $\gamma = 80\%$

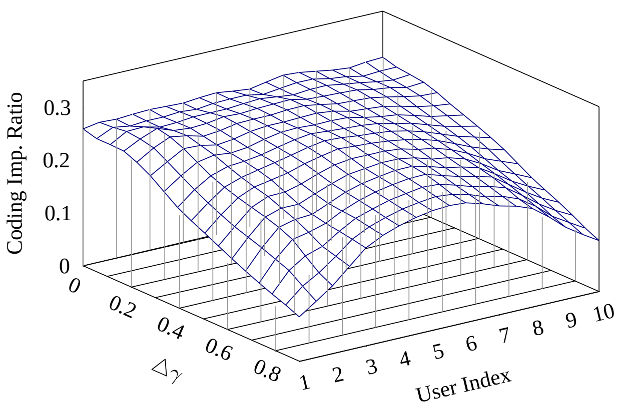
FIGURE 6.7: Reduced retransmission ratio with different number of stations in static channels.



(a) Goodput



(b) Fairness index



(c) Coding improvement ratio

FIGURE 6.8: Heterogeneous and static channels, where $E(\gamma) = 0.5$.

Mobile Speed (m/s)	1	5	10	20
Coherence Time (ms)	122.88	24.57	12.28	6.14

TABLE 6.1: Mapping between mobile speed and coherence time.

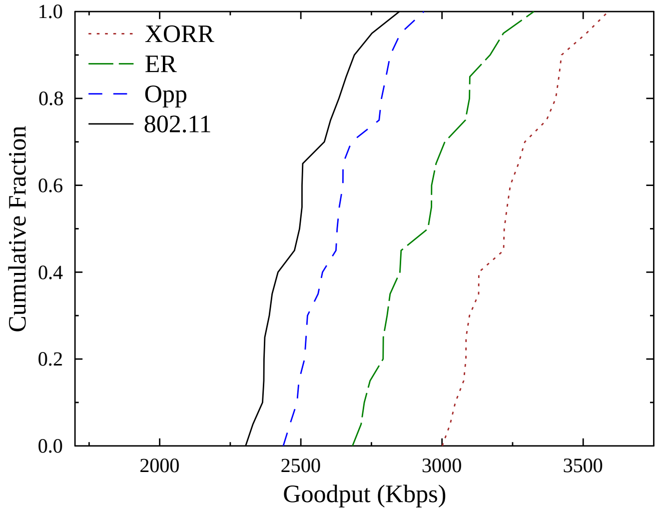
6.4.2 Time-Varying Channels

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" [112]. The coherence time of a wireless link depends on the station's mobility and its surrounding environment. Table 6.1 shows the mapping between mobile speeds and the coherence time [69]. Assume that the channel is stationary. We characterize the wireless channel by using Gaussian distribution for link reliability and transmission rate. Therefore, the link reliability and the transmission rate are modelled by the mean and the variance, i.e. ($\bar{\gamma}_i$, $\sigma_{\gamma,i}$) and ($\bar{r}_i$, $\sigma_{r,i}$)[1]

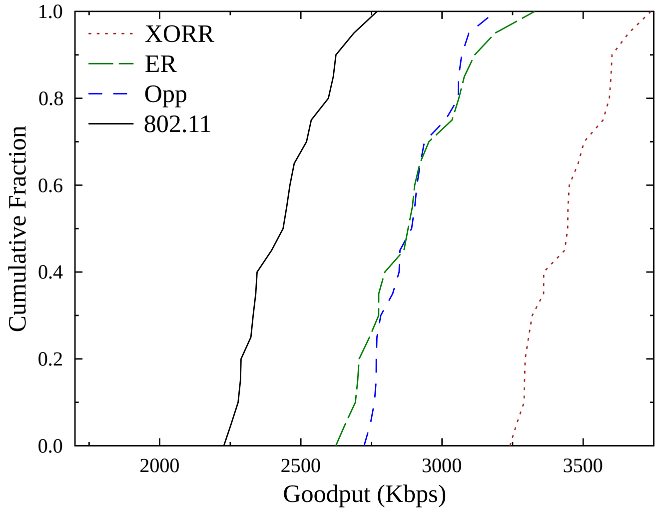
In the simulations, the mean of each link's reliability $\bar{\gamma}_i$ is randomly chosen from [0.3,0.7]. Figure 6.9 shows the Cumulative Distribution Function (CDF) of the goodput when the transmission rate is fixed to 5.5Mbps while the variance of the link reliabilities is $\sigma_\gamma = 0.1$ and $\sigma_\gamma = 0.01$, respectively. In Figure 6.9(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 Figure 6.9(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.

Figure 6.10 shows the goodput performance with time-varying transmission rates. Each station's transmission rate varies among 2, 5.5 and 11Mbps. As shown in Figure 6.10, 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 [7]. 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

[1] 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) $\sigma_\gamma = 0.01$



(b) $\sigma_\gamma = 0.1$

FIGURE 6.9: Goodput in time-varying channel: static and homogeneous transmission rate $r = 5.5$Mbps. $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$. Coherence time is 24.57 ms. The number of stations is 12.

multi-user diversity. XORR, on the other hand, performs best by exploiting both multi-user diversity and the network-coding.

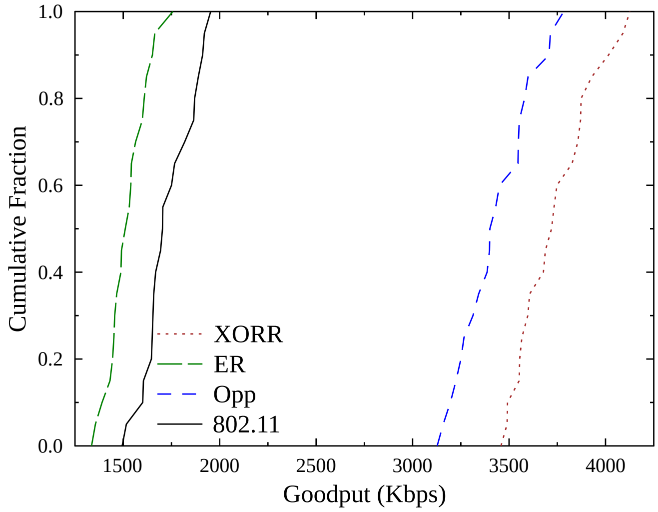


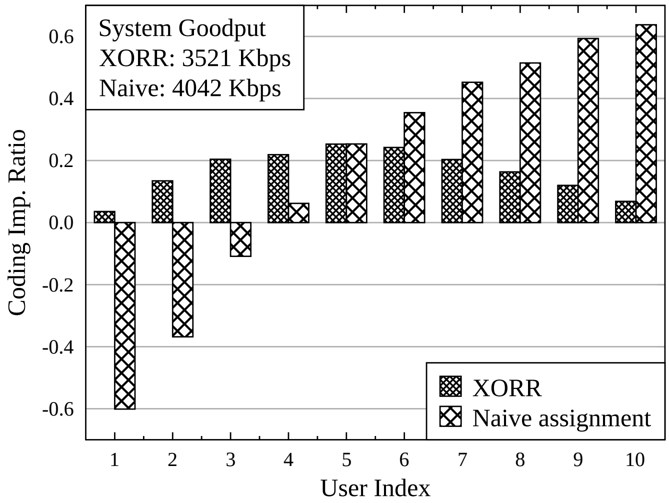
FIGURE 6.10: Goodput in time-varying channel: time-varying transmission rates r. $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$. Coherence time is 24.57 ms.

6.5 Fairness

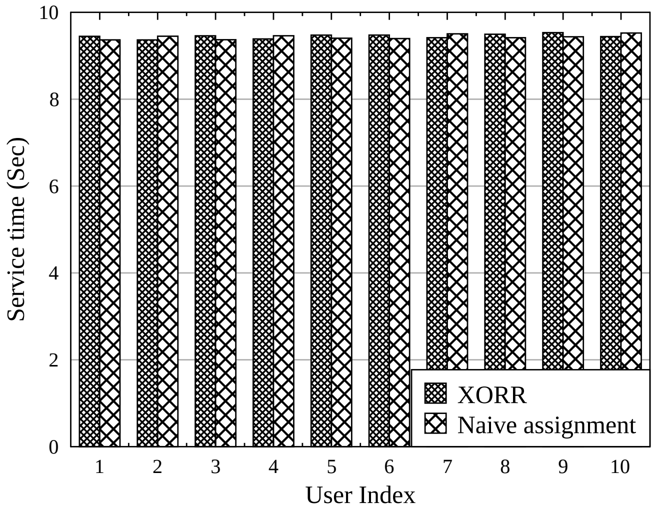
6.5.1 NC-fair Assignment

As aforementioned, NC-fair assignment is critical for designing a fair NC scheduling. For maintaining temporal fairness, NC-fair assignment concerns about service time assignment for transmitting network coded frames. For understanding the impact of service time assignment on XORR, we compare our algorithm in Equation (4.11) with a naive service time assignment, which evenly distributes the transmission time among all stations in the coding-set. The static wireless link model is used in the simulations. The link reliability for each wireless link γ_i is randomly chosen from [0.2,0.9]. Figure 6.11 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 figures.

As shown in Figure 4.11, the service time is evenly allocated to stations in two algorithms. However, when the naive service time assignment is used in XORR, some less reliable stations have *coding loss*. That is because the service time is not assigned proportional to their coding condition but distributed evenly. Hence, the stations with bad wireless conditions would eventually lose the “real” service



(a) Coding improvement ratio



(b) Allocated service time

FIGURE 6.11: Starvation of XORR with naive service time assignment

time than they would have in non coding scheme. By contrast, XORR improves the goodput of all stations while maintaining fairness. In other words, XORR can achieve NC-fairness, where temporal fairness is maintained and there is no *coding loss* for all stations. Note that in this case, the system goodput of XORR with naive assignment is 4Mbps, which is higher than XORR (3.5Mbps). But this is reasonable since the naive assignment algorithm assigns more service time to the stations with good wireless conditions and thus the overall system goodput is better.

6.5.2 Weighted Fairness

In the previous simulations, we showed only the results of equal service time share, i.e. the Jain's index of each user's service time is one. However, due to different QoS requirements, the end hosts may be assigned with different weights. We have derive scheduling mechanism for supporting the weighted fairness in XORR. In this section, XORR schemes with different weight assignments are evaluated. In order to explore the coding gain without the effect of temporal fairness, we compare XORR with *Opp* under different weight settings.

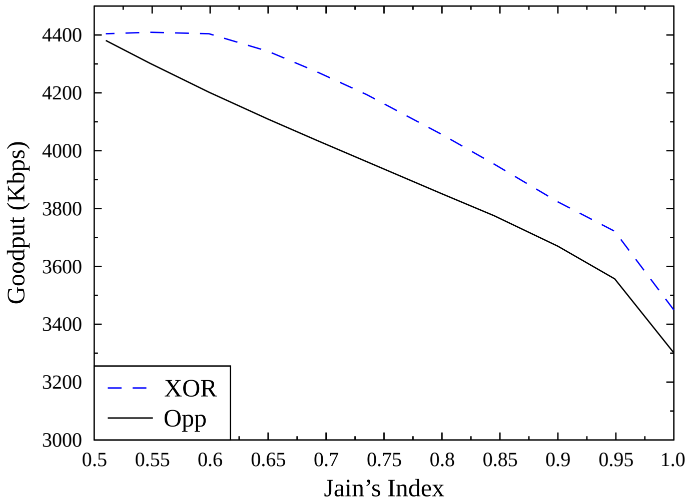
For generating different weights for stations, We define that time weight for user i is

$$\omega_i = 1 + \triangle\omega \times (i-1). \tag{6.1}$$

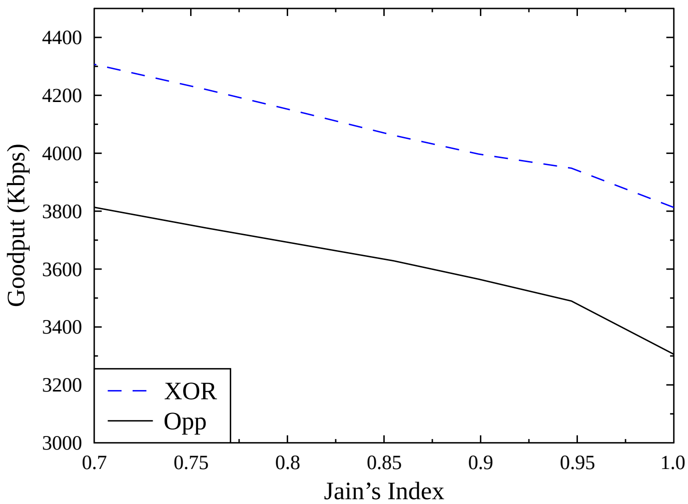
We vary $\triangle\omega$ in order to create different weight requirements. Jain's Index [111] is used for indicating the difference among weights.

$$\text{Jain's Index} = \frac{\left(\sum_{i=1}^{N} \omega_i\right)^2}{N \cdot \sum_{i=1}^{N} \omega_i^2},$$

where ω_i is the time weight for the station u_i. The smaller value of Jain's index, the larger difference among weights. The static heterogeneous channel model is used, where each station's link reliability γ_i is randomly selected between 0.4 and 0.8. The simulation time is 600 seconds. Figure 6.12 shows the results under 2 and 5 stations. It is shown that XORR can outperform traditional opportunistic scheduling scheme under different time weight requirements. Furthermore, the more stations in the network, the more coding gain XORR can achieve under different weight requirements.



(a) $N = 2$



(b) $N = 5$

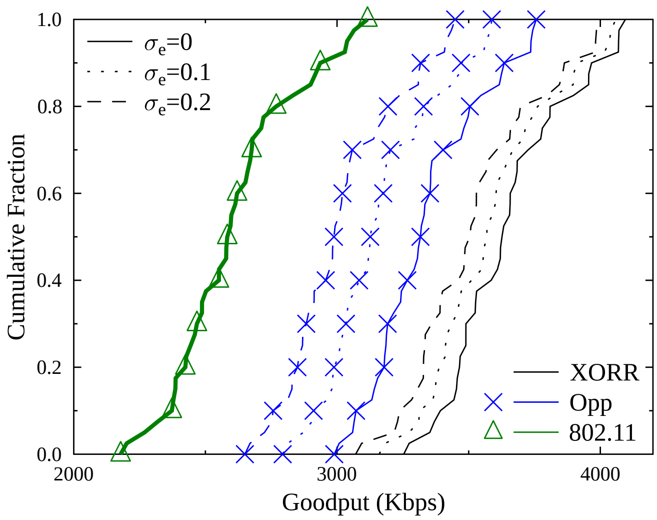
FIGURE 6.12: Goodput in heterogeneous channel model with different weight assignment for service time. In heterogeneous channel model, γ for each user is randomly chosen between 0.4 and 0.8. The number of stations is 2 and 5, transmission rate is 5.5 Mbps, $\theta = 2$ and $\beta = 50$.

6.6 Impact of Estimation Accuracy

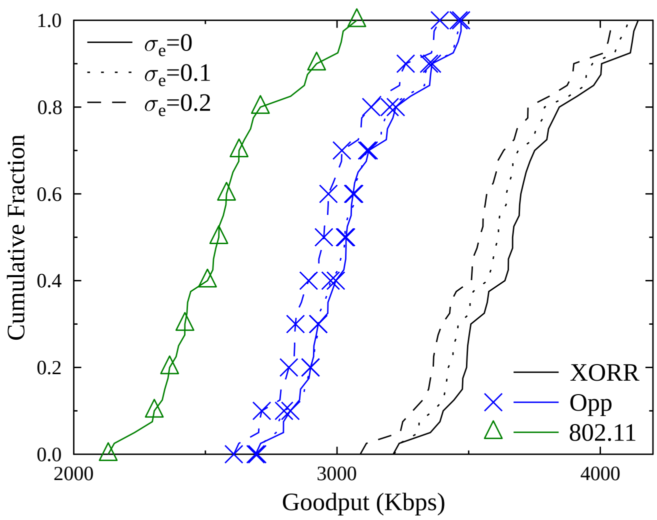
XORR relies on reception estimation to select coding-sets. To estimate the reception of native frame for each user, the AP further needs to estimate the link reliability. Many existing wireless system already maintains such statistics (*e.g.* WLAN [100]). We now evaluate XORR if reliability estimation 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 $\mathcal{N}(0, \sigma_e)$. Figure 6.13 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 Figure 6.13(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.

6.7 Impact of Transmission 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. Figure 6.14 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, *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.



(a) Coherence time 6.14 ms.



(b) Coherence time 24.57 ms.

FIGURE 6.13: Goodput with estimation error in time-varying channel model where, $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.3, 0.7]$.

6.8 Experiment Results

We have prototyped XORR and preliminarily evaluated its performance on real wireless test-bed[2]. Our implementation is based on Atheros AR5212 wireless NIC

[2]The implementation and the test-bed setup are done by the colleagues in Micorsoft Research Asia. Please refer to our report [113] for more details.

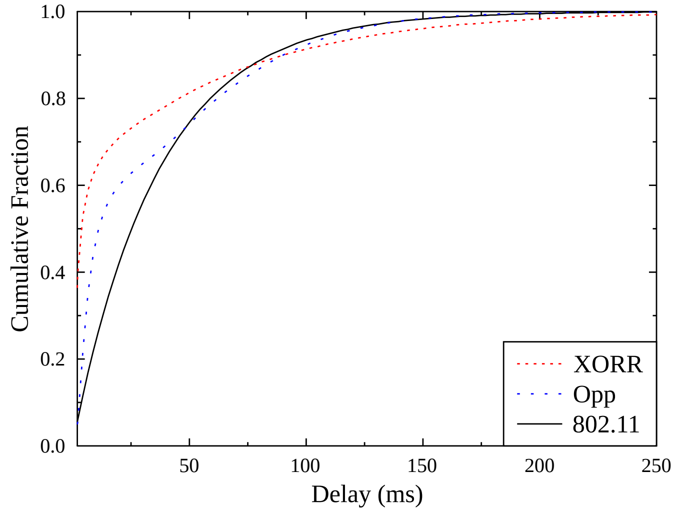


FIGURE 6.14: CDF of frame transmission delay in time-varying channel model where, $\sigma_\gamma = 0.1$, $\bar{\gamma}_i$ is randomly chosen from $[0.4, 0.8]$. Coherence time is 24.57 ms.

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

TABLE 6.2: Goodput improvement in test-bed experiments. XX/YY means the goodput improvement of XX over YY.

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 6.2 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 *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.

6.9 Chapter Summary

In this chapter, we have evaluated XORR with simulations and real wireless testbeds. Here are the conclusions of our results:

1. The proposed heuristic algorithm only slightly degrades the performance but reduces drastically the complexity of the exhaustive search.

2. XORR outperforms ER by $10-20\%$, *Opp* by $20-25\%$ and *802.11* by $30-40\%$.

3. Confirmed by the simulations, XORR does achieve NC fairness and an inappropriate time assignment may cause significant unfairness.

4. The schedulers in XORR and *Opp* rely on the estimation of the link quality to make scheduling decisions. XORR is less sensitive to estimation error compared to *Opp* because the network coding actually could average this error out.

5. Although XORR may defer the retransmission for potential coding opportunity, a large portion of frames in XORR has an even shorter transmission delay (70% of frames has less than 5ms delay).

6. In our testbed results, XORR improves goodput by 10.7% for UDP traffic and 15.7% for TCP traffic, compared to the IEEE 802.11 network.

Chapter 7

Conclusions and Future Work

In this chapter, we conclude the dissertation by summarizing our contributions, enumerating several remaining challenges and proposing directions for future work.

7.1 Contributions

Below, we highlight the contributions of this dissertation.

7.1.1 A Global Design for Wireless Network Coding

The key difference of XORR from past work is that, rather than designing the coding scheme independently from other wireless properties, XORR adopts a global design by considering the utilization of coding opportunity, fairness issue, adaptation to time-varying channel condition and handling multiple transmission rates. Each node relies on local information to detect and exploit the opportunities provided by not only network coding but also multiuser diversity whenever they arise.

7.1.2 Integrating Bayesian Learning with Wireless Network Coding Protocols

For utilizing coding opportunities, each node in the network has to know the reception status of the nearby nodes. Obtaining reception status is critical for the design of wireless network coding, since the efficiency of coding utilization relies heavily on the sufficiency of reception information. Previous work [8, 11] adopts the reception report scheme for obtaining the reception information. However, through our simulations in Section 3.3, such a scheme causes the following problems:

- The signaling overhead for reception report offsets the coding gains.

- Using larger period of report can avoid the burden of signaling overhead, but results in less coding efficiency

- It is difficult to adjust the period for sending reports because the optimal period depends on the transmission rate, link quality as well as the number of nodes in the network, which are normally time-varying in wireless networks.

On the other hand, rather than explicitly receiving reception report from other nodes, our Bayesian-learning based estimation for reception status of nearby nodes can provide substantially coding opportunity and completely eliminate signaling overhead. Note that although we focus on WLAN scenarios, our estimation scheme can be easily extended for wireless network coding schemes used in multi-hop wireless networks.

7.1.3 New Concept for Wireless Network Coding

In this dissertation, we introduce new concepts and techniques that may be used by other wireless network coding protocols outside the XORR framework.

- **Coding Metric**.

An NC scheme selects a set of frames to encode so that the coded frame can achieve the maximal coding metric. Existing NC schemes (e.g. [8, 11]) search for the coding-set with the maximal size of the decodable sets[1]. However, in practical wireless networks, the transmission rate is adjusted based on its channel quality. Therefore, such a coding metric may yield suboptimal results, since the coded frame must be transmitted at the lowest transmission rate in that set.

A new coding metric, expected goodput, is devised in XORR to replace the metric of the maximal decodable set for measuring the coding benefit. This coding

[1]Please see its definition in Seciton 3.1

metric, expected goodput, naturally accommodates the heterogeneities in wireless networks, such as transmission rate, link reliability and frame size. More specifically, the coding metric of expected goodput is opportunism in the aspect of both the coding and the wireless condition, i.e. each node relies on local information to detect and exploit coding as well as better link opportunities whenever they arise. Therefore, the gain of network coding and multiuser diversity can be utilized at the same time.

- **NC-Fairness**

In traditional non-NC fair scheduling, the individual performance is linear determined by the resource (e.g. service time or bandwidth) assigned to it. Accordingly, the fairness guarantee in non-NC fair scheduling implies certain performance guarantee for individual wireless stations. However, with NC, such implication becomes tricky because the resource cost for transmitting a coded frame is essentially shared among the stations whose frames are encoded in the coded frame, and it can be arbitrarily assigned. With a extreme case that a station always shares all the resource cost when its frames are encoded in the coded frames, it can be inferred that the performance of this station may be worse than it is with the non-NC scheme.

Therefore, we propose the concept of *NC-fairness*, where not only the fair resource share is guaranteed but also the performance for *every* wireless station is improved. An NC-aware fair opportunistic scheduling is designed in XORR, which is theoretically proven to achieve NC-fairness. This is also confirmed by our simulation results.

7.1.4 Framework of NC-aware Scheduling

A framework of an NC-aware fair opportunistic scheduling is proposed in XORR. The task of the scheduling discipline is to optimize the system performance (utility) under certain fairness constraint. To provide a bounded short-term fairness among all clients, XORR follows a credit based approach as in [60] that assigns a state variable, *credit*, to control the fairness property. It is extended to support network coding by selecting a set of frames, rather than a single frame in the previous scheduling mechanisms.

An exhaustive search algorithm for finding the optimal coding-set is computationally expensive. We prove such an optimal selection of coding-sets is NP-hard. Hence, a heuristic algorithm for selection is proposed in XORR.

Furthermore, for satisfying different QoS requirements for different end hosts, our proposed scheduling algorithm also supports weighted fairness, wherein flows with larger weights receive correspondingly better service.

7.1.5 XORR Outperforms Other Protocols

We theoretically characterize the potential coding gain of XORR. We also evaluate XORR with simulations and real wireless testbeds. The numerical results suggest that the XORR should defer the retransmissions moderately for creating more coding opportunities. Furthermore, the results also show that with a moderate number of wireless station N (e.g. $N = 100$), XORR is able to effectively enhance system performance. Our simulation results demonstrate that XORR is less sensitive to estimation error of the link quality. And the proposed heuristic algorithm only slightly degrades the performance but reduces drastically the complexity of the exhaustive search. Moreover, a large portion of frames in XORR has an even shorter transmission delay (70% of frames has less than 5ms delay).

In summary, the results show that XORR outperforms ER by $10 - 20\%$, *Opp* by $20-25\%$ and *802.11* by $30-40\%$. In our testbed results, XORR improves goodput by 10.7% for UDP traffic and 15.7% for TCP traffic, compared to the IEEE 802.11 network.

7.2 Remaining Challenges and Future Work

The system in this dissertation addressed the major challenges involved in integrating network coding with wireless networks. Below, we enumerate a few challenging issues that our dissertation odes not solve. There are not fundamental limitations, rather they are outstanding problems that, we believe, are solvable within XORR framework.

7.2.1 Joint Uplink/Downlink NC-aware Scheduling

The scheduling in XORR mainly focuses on the downlink traffic since current Internet traffic are mostly downloading. However, due to the emerging new applications such as collaborative download, and peer-to-peer file sharing, WLAN traffic trends to evolve to be more symmetric. Without proper scheduler, the downlink throughput gain diminishes proportionally to the increasing number of stations transmitting on the uplink. Hence, the synergistic integration of both the uplink and the downlink NC-aware scheduling remains an open problem.

7.2.2 NC-aware Bit-Rate Adaptation

Our XORR scheduler is independent from the rate adaptation for links. The rate adaptation algorithm first decides the transmission rates for the flows according to its link quality. Then the XORR scheduler selects the coding-set and transmits the coded frame with the lowest rate among flows. Such independent rate adaptation algorithm behaves sub-optimally under network scheme because the estimation of channel does not consider the effect of network coding. An NC-aware rate adaptation algorithm certainly enhance the performance. The interaction between rate adaptation and network coding remains an open problem.

7.3 Final Remarks

In this dissertation, we presented XORR, an efficient retransmission scheduling scheme based on network coding. XORR exploits the *broadcast* advantage of wireless medium to reduce the retransmissions. We conducted extensive simulations and real testbed experiments to study the performance of XORR. Our results showed that, by exploiting both multiuser diversity and network coding, XORR has a consistent improvement over non-coding schemes (802.11 and traditional opportunistic scheduling); while the prior NC-aided ARQ scheme, ER, 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 in a better goodput for *each* wireless stations in the system, compared with traditional opportunistic schedulers. We believe that, because of the substantial gains possible as well as the viable and practical design, XORR could be the foundation for building up a new link-layer retransmission scheme.

Bibliography

[1] "IEEE 802.11a Medium Access Control (MAC) and Physical Layer (PHY) Specification: High-speed Physical Layer in the 5 GHz Band," December 1999.

[2] "IEEE 802.11g Medium Access Control (MAC) and Physical Layer (PHY) Specification Amendment 4: Further High Data Rate Extension in the 2.4 GHz Band," June 2003.

[3] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High Throughput Path Metric for MultiHop Wireless Routing," in *ACM MOBICOM*, 2003.

[4] 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.

[5] M. Rodrig, C. Reis, R. Mahajan, D. Wetherall, and J. Zahorjan, "Measurement-based characterization of 802.11 in a hotspot setting," in *ACM SIGCOMM workshop on Experimental approaches to wireless network design and analysis (E-WIND)*, pp. 5–10, 2005.

[6] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network Information Flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

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

[8] 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.

[9] S. Sengupta, S. Rayanchu, and S. Banerjee, "An Analysis of Wireless Network Coding for Unicast Sessions: The Case for Coding-Aware Routing," in *IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1028–1036, May 2007.

[10] 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/.

[11] E. Rozner, A. P. Iyer, Y. Mehta, and L. Qiu, "ER: Efficient Retransmission Scheme for Wireless LANs," in *ACM CoNext*, December 2007.

[12] P. Larsson and N. Johansson, "Multi-User ARQ," in *IEEE Vehicular Technology Conference 2006-Spring*, pp. 2052 – 2057, 2006.

[13] R. Knopp and P. Humblet, "Information capacity and power control in single-cell multiusercommunications," in *IEEE International Conference on Communications*, vol. 1, pp. 331 – 335, June 1995.

[14] L. Hanzo, M. Mnster, B. Choi, and T. Keller, *OFDM and MC-CDMA for Broadband Multi-User Communications, WLANs and Broadcasting*. John Wiley & Sons Ltd., 2003.

[15] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error-Correcting Coding and Decoding: Turbo Codes," in *IEEE International Conference on Communications*, pp. 1064–1070, 1993.

[16] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, pp. 1710–1722, March 1999.

[17] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society of Industrial and Applied Mathematics SIAM*, vol. 8, pp. 300–304, June 1960.

[18] "IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications," August 1999.

[19] J.-S. Ahn, S.-W. Hong, and J. Heidemann, "An Adaptive FEC Code Control Algorithm for Mobile Wireless Sensor Networks," *Journal of Communications and Networks*, vol. 7, no. 4, pp. 489–499, 2005.

[20] P. S. Sindhu, "Retransmission Error Control with Memory," *IEEE Transactions on Communications*, vol. Com-25, pp. 473–479, May 1977.

[21] S. S. Chakraborty, M. Liinaharja, and K. Ruttik, "Diversity and packet combining in Rayleigh fading channels," in *IEE Proceedings in Communications*, November 2004.

[22] A.-G. A. Daraiseh and C. W. Baum, "Methods for packet combining in HARQ systems over bursty channels," in *ACM Mobile Networks and Applications*, vol. 2, pp. 213–224, October 1997.

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

[24] D. Chase, "Code combiningXA maximum likelihood decoding approach for combining an arbitrary number of noisy packets," *IEEE Transaction on Communications*, vol. 33, pp. 385–393, May 1985.

[25] D. Mandelbaum, "An Adaptive-Feedback Coding Scheme Using Incremental Redundancy," *IEEE Transactions on Infomation Theory*, vol. 20, pp. 388–389, May 1974.

[26] J. Metzner, "Improvements in Block-Retransmission Schemes," *IEEE Transactions on Communications*, vol. 27, pp. 524–532, February 1979.

[27] S. Lin and P. Yu, "A hybrid ARQ scheme with parity retransmissions for error control of satellite channels," *IEEE Transactions on Communications*, vol. 30, pp. 1701–1719, 1982.

[28] R. K. Ganti, P. Jayachandran, H. Luo, and T. F. Abdelzaher, "Datalink streaming in wireless sensor networks," in *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, (New York, NY, USA), pp. 209–222, ACM, 2006.

[29] K. Jamieson and H. Balakrishnan, "PPR: Partial Packet Recovery for Wireless Networks," in *ACM SIGCOMM*, (Kyoto, Japan), August 2007.

[30] A. Miu, H. Balakrishnan, C. Emre, and Koksal, "Improving loss resilience with multi-radio diversity in wireless networks," in *ACM MobiCom*, pp. 16–30, 2005.

[31] G. Woo, P. Kheradpour, D. Shen, and D. Katabi, "Beyond the bits: cooperative packet recovery using physical layer information," in *ACM MobiCom*, pp. 147–158, 2007.

[32] H. Dubois-Ferrière, D. Estrin, and M. Vetterli, "Packet combining in sensor networks," in *Proceedings of ACM SenSys*, pp. 102–115, 2005.

[33] Y.-C. Cheng, J. Bellardo, P. Benk, A. C. Snoeren, G. M. Voelker, and S. Savage, "Jigsaw: solving the puzzle of enterprise 802.11 analysis," in *ACM SIGCOMM*, 2006.

[34] G. Judd and P. Steenkiste, "Using Emulation to Understand and Improve Wireless Networks and Applications," in *2nd Symposium on Networked Systems Design and Implementation (NSDI'05)*, 2005.

[35] S. Khurana, A. Kahol, and A. P. Jayasumana, "Effect of Hidden Terminals on the Performance of IEEE 802.11 MAC Protocol," in *23rd Annual Conference on Local Computer Networks*, 1998.

[36] P. C. Ng, S. C. Liew, K. C. Sha, and W. T. To, "Experimental Study of Hidden node Problem in IEEE 802.11 Wireless Networks," in *ACM Sigcomm Poster*, 2005.

[37] J. C. C. Ware, J. Judge and E. Dutkiewicz, "Unfairness and capture behaviour in 802.11 ad hoc networks," in *IEEE International Conference Communications (ICC)*, vol. 1, p. 159V163, 2000.

[38] P. Karn, "MACA - a new channel access method for packet radio," in *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990.

[39] K. Xu, M. Gerla, and S. Bae, "Effectiveness of RTS/CTS Handshake in IEEE 802.11 Based Ad Hoc Networks," *Ad Hoc Network Journal*, 2003.

[40] "Broadcom Wireless LAN Adapter User Guide."

[41] "Reference Manual for the NETGEAR ProSafe 802.11g Wireless AP WG102."

[42] D. Halperin, J. Ammer, T. Anderson, and D. Wetherall, "Interference Cancellation: Better Receivers for a New Wireless MAC," in *ACM Hotnets*, 2007.

[43] D. Halperin, T. Anderson, and D. Wetherall, "Practical interference cancellation for wireless LANs," in *ACM MOBICOM*, 2008.

[44] C. Reis, R. Mahajan, M. Rodrig, D. Wetherall, and J. Zahorjan, "Measurement-based models of delivery and interference," in *ACM SIGCOMM*, 2006.

[45] B. G. R. Gummadi, D. Wetherall and S. Seshan, "Understanding and Mitigating the Impact of RF Interference on 802.11 Networks," in *ACM SIGCOMM*, 2007.

[46] S. S. V. Bharghavan, A. J. Demers and L. Zhang, "MACAW: A Media Access Protocol for Wireless LAN's," in *ACM SIGCOMM*, 1994.

[47] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Solutions to Hidden Terminal Problems in Wireless Networks," in *ACM SIGCOMM*, p. 39V49, 1997.

[48] A. Muqattash and M. Krunz, "CDMA-Based MAC Protocol for Wireless Ad Hoc Networks," in *ACM MOBIHOC*, 2003.

[49] J. Zhu, X. Guo, S. Roy, and K. Papagiannaki, "CSMA Self-Adaptation based on Interference Differentiation," in *IEEE Globecom*, 2007.

[50] S. Gollakota and D. Katabi, "Zigzag decoding: combating hidden terminals in wireless networks," in *ACM SIGCOMM*, no. 4, pp. 159–170, October 2008.

[51] P. Viswanath, D. N. C. Tse, and V. Anantharam, "Asymptotically optimal waterfilling in vector multiple access channels," *IEEE Transactions on Information Theory*, January 2001.

[52] N. Joshi, S. R. Kadaba, S. Patel, and G. S. Sundaram, "Downlink scheduling in CDMA data networks," in *ACM MOBICOM*, 2000.

[53] P. Viswanath, D. N. C. Tse, and R. Laroia, "Opportunistic beamforming using dumb antennas," *IEEE Transactions of Information Theory*, 2002.

[54] P. Bender, P. Black, M. Grob, R. Padovani, N. Sindhushayana, and A. Viterbi, "CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Communications Magazine*, 2000.

[55] H. Holma and A. Toskala, *WCDMA for UMTS, Radio access for third generation mobile communications*. Wiley, 2nd ed., 2002.

[56] X. Qin and A. Berry, "Exploiting multiuser diversity for medium access control in wireless networks," in *IEEE INFOCOM*, pp. 1084–1094, 2003.

[57] Z. Ji, Y. Yang, J. Zhou, M. Takai, and R. Bagrodia, "Exploiting medium access diversity in rate adaptive wireless LANs," in *Proceedings of the 10th annual international conference on Mobile computing and networking (MobiCom'04)*, Septemer - October 2004.

[58] S. Lu, V. Bharghavan, and R. Srikant, "Fair scheduling in wireless packet networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 7, no. 4, pp. 473 – 489, 1999.

[59] T. Ng, I. Stoica, and H. Zhang, "Packet fair queuing algorithms for wireless networks with location dependent errors," in *IEEE INFOCOM*, 1998.

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

[61] K. T. Chan, B. Bensaou, and D. H. K. Tsang, "Credit-based fair queueing (cbfq," *IEE Electronics Letters*, vol. 33, pp. 584–585, 1997.

[62] X. Liu, E. K. P. Chong, and N. B. Shroff, "Opportunistic transmission scheduling with resource-sharing constraints in wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 2053–2064, October 2001.

[63] H. Luo, S. Lu, and V. Bharghavan, "A new model for packet scheduling in multihop wireless networks," in *ACM MOBICOM*, August 2000.

[64] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 3, pp. 86–98, January-February 2004.

[65] N. Vaidya and P. Bahl, "Fair scheduling in broadcast environments," tech. rep., Microsoft Research, 1999. MSR-TR-99-61.

[66] N. Vaidya, P. Bahl, and S. Gupta, "Distributed fair scheduling in a wireless LAN," in *ACM MOBICOM*, 2000.

[67] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," in *IEEE INFOCOM*, 2003.

[68] B. Radunovic and J. Y. L. Boudec, "Rate Performance Objectives of Multi-hop Wireless Networks," in *IEEE INFOCOM*, 2004.

[69] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knighlty., "Opportunistic Media Access for Multirate Ad Hoc Networks," in *ACM MOBICOM*, 2002.

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

[71] C. Fragouli, J.-Y. L. Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Computer Communication Review*, vol. 36, pp. 63–68, January 2006.

[72] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear Network Coding," *IEEE Transactions on Informatoin Theory*, vol. 49, pp. 371–381, February 2003.

[73] T. Ho, R. Koetter, M. Médard, D. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE International Symposium on Information Theory (ISIT)*, pp. 442 – 442, June 2003.

[74] T. Ho, M. Médard, M. Effros, and D. Karger, "On Randomized Network Coding," in *41st Allerton Annual Conference on Communication, Control and Computing*, October 2003. http://web.mit.edu/medard/www/pubs.html.

[75] Y. Wu, P. Chou, and K. Jain, "A Comparison of Network Coding and Tree Packing," in *Proc. of IEEE International Symposium on Information Theory (ISIT)*, 2004. http://research.microsoft.com/~pachou/pubs/WuCJ04.pdf.

[76] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial Time Algorithms for Network Information Flow," in *15th ACM Symposium on Parallel Algorithms and Architectures*, pp. 286–294, 2003. http://www.mpi-sb.mpg.de/~sanders/papers/index.html.

[77] C. Fragouli and E. Soljanin, "Decentralized network coding," in *Information Theory Workshop*, October 2004.

[78] P. Chou, Y. Wu, and K. Jain, "Practical Network Coding," in *41st Allerton Conf. Communication, Control and Computing*, October 2003.

[79] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "A Network Coding Approach to Energy Efficient Broadcasting: From Theory to Practice," in *IEEE INFOCOM*, pp. 1–11, April 2006.

[80] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, pp. 782 – 795, October 2003.

[81] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, K. Jain, and L. Tolhuizen, "Polynomial Time Algorithms for Multicast Network Code Construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.

[82] D. S. Lun, M. Médard, and R. Koetter, "Efficient operation of wireless packet networks using network coding," in *International Workshop on Convergent Technologies (IWCT)*, 2005.

[83] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, 2007.

[84] C. Fragouli, J. Widmer, and J.-Y. L. Boudec, "Efficient broadcasting using network coding," *IEEE/ACM Transactions on Networking*, 2007.

[85] N. Cai and R. Yeung, "Secure network coding," in *IEEE International Symposium on Information Theory (ISIT)*, p. 323, 2002.

[86] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of byzantine adversaries," *IEEE Transactions on Information Theory*, 2006.

[87] C. Gkantsidis and P. Rodriguez, "Network Coding for Large Scale Content Distribution," in *IEEE INFOCOM*, 2005.

[88] J. Widmer and J.-Y. L. Boudec, "Network Coding for Efficient Communication in Extreme Networks," in *ACM SIGCOMM WDTN*, 2005.

[89] A. Jiang, "Network Coding for Joint Storage and Transmission with Minimum Cost," in *IEEE International Symposium on Information Theory (ISIT)*, 2006.

[90] T. Ho, B. Leong, M. Médard, R. Koetter, Y. Chang, and M. Effros, "On the utility of network coding in dynamic environments," in *International Workshop on Wireless Ad-hoc Networks (IWWAN)*, (Finland), June 2004. http://web.mit.edu/trace/www/.

[91] D. S. Lun, M. Médard, R. Koetter, and M. Effros, "Further results on coding for reliable communication over packet networks," in *IEEE International Symposium on Information Theory*, 2005.

[92] S. Deb, M. Effros, T. Ho, D. R. Karger, R. Koetter, D. S. Lun, M. Medard, and N. Ratnakar, "Network coding for wireless applications: A brief tutorial," in *IWWAN*, 2005.

[93] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *ACM SIGCOMM*, 2007.

[94] S. Katti, S. Gollakota, and D. Katabi, "Embracing wireless interference: Analog network coding," in *ACM SIGCOMM*, 2007.

[95] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level Network Coding for Wireless Mesh Networks," in *ACM SIGCOMM*, 2008.

[96] D. S. Lun, M. Médard, and R. Koetter, "Network coding for efficient efficient wireless unicast," in *International Zurich Seminar on Communications (IZS)*, 2006.

[97] P.Bhagwat, P. Bhattacharya, A. Krishma, and S. Tripathi, "Enhancing throughput over wireless LANs using channel state dependent packet scheduling," in *IEEE INFOCOM*, April 1996.

[98] T. E. Ng, I. Stoica, and H. Zhang, "Packet fair queueing algorithms for wireless networks withlocation-dependent errors," in *IEEE INFOCOM*, vol. 3, pp. 1103 – 1111, March-April 1998.

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

[100] S. H. Y. Wong, H. Yang, S. Lu, and V. Bharghavan, "Robust rate adaptation for 802.11 wireless networks," in *ACM MobiCom*, pp. 146–157, 2006.

[101] A. Kamerman and L. Monteban, "WaveLAN II: A high-performance wireless LAN for the unlicensed band," in *Bell Labs Technical Journal*, 1997.

[102] M. Lacage, M. H. Manshaei, and T. Turletti, "IEEE 802.11 Rate Adaptation: A Practical Approach," in *ACM MSWiM*, 2004.

[103] "Onoe Rate Control." http://madwifi.org/browser/madwifi/trunk/ath_rate/onoe.

[104] J. Bicket, "Bit-rate Selection in Wireless Networks," Master's thesis, MIT, 2005.

[105] G. Holland, N. Vaidya, and V. Bahl, "A Rate-Adaptive MAC Protocol for Multihop Wireless Networks," in *ACM MOBICOM*, 2001.

[106] J. Kim, S. Kim, S. Choi, and D. Qiao, "CARA: Collision-aware Rate Adaptation for IEEE 802.11 WLANs," in *IEEE INFOCOM*, 2006.

[107] D. Qiao, S. Choi, and K. Shin, "Goodput Analysis and Link Adaptation for IEEE 802.11a Wireless LANs," *IEEE Transactions on Mobile Computing*, vol. 1, no. 4, 2002.

[108] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Hybrid Rate Control for IEEE 802.11," in *ACM MobiWac*, 2004.

[109] I. Haratcherev, K. Langendoen, R. Lagendijk, and H. Sips, "Fast 802.11 Link Adaptation for Real-time Video Streaming by Cross-Layer Signaling," in *IEEE International Symposium on Circuits and Systems*, 2005.

[110] D. Qiao and S. Choi, "Fast-responsive Link Adaptation for IEEE 802.11 WLANs," in *IEEE ICC*, 2005.

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

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

[113] F.-C. Kuo, K. Tan, X.-Y. Li, J. Zhang, and X. Fu, "Network Coding-Aware Fair Opportunistic Scheduling in Lossy Wireless Networks," Technical Report IFI-TB-2008-003, Institute of Computer Science, University of Göttingen, 2008.

Lebenslauf

Name	Kuo
Vorname	Fang-Chun
Geburtsdatum	02.04.1978
Geburtsort	Pingtung, Taiwan
Staatsangehörigkeit	Taiwan

Schulische Ausbildung

1984 - 1990	Chung-Hsiao Elementary School, Pingtung, Taiwan
1990 - 1993	Chung Cheng Junior High School, Pingtung, Taiwan
1993 - 1996	Nationall Pingtung Girls' Senior High School, Pingtung, Taiwan

Wissenschaftlicher Bildungsweg

1996 - 2000	National Chung-Hsing University, Taiwan Bachelor of Science in Electrical Engineering
2000 - 2002	National Chiao-Tung University, Taiwan Master of Science in Communication Engineering
2004 - 2005	University of Southampton, UK Master of Philosophy in Electronics
2006 - 2009	University of Göttingen, Germany Doctor of Philosophy (Doctor rerum naturalium) in Computer Science