Adaptive Network Coding and Scheduling for Maximizing Throughput in Wireless Networks

Prasanna Chaporkar
Dept. of Electrical Engineering
Indian Inst. of Technology, Mumbai
chaporkar@ee.iitb.ac.in

Alexandre Proutiere
Microsoft Research
Cambridge, UK
alexandre.proutiere@microsoft.com

ABSTRACT

Recently, network coding emerged as a promising technology that can provide significant improvements in throughput and energy efficiency of wireless networks, even for unicast communication. Often, network coding schemes are designed as an autonomous layer, independent of the underlying Phy and MAC capabilities and algorithms. Consequently, these schemes are greedy, in the sense that all opportunities of broadcasting combinations of packets are exploited. We demonstrate that this greedy design principle may in fact reduce the network throughput. This begets the need for adaptive network coding schemes. We further show that designing appropriate MAC scheduling algorithms is critical for achieving the throughput gains expected from network coding. In this paper, we propose a general framework to develop optimal and adaptive joint network coding and scheduling schemes. Optimality is shown for various Phy and MAC constraints. We apply this framework to two different network coding architectures: COPE, a scheme recently proposed in [7], and XOR-Sym, a new scheme we present here. XOR-Sym is designed to achieve a lower implementation complexity than that of COPE, and yet to provide similar throughput gains.

Categories and Subject Descriptors

C.2.1 [**Network Architecture and Design**]: Wireless communication

General Terms

Algorithms, Design, Performance

Keywords

Multi-hop wireless networks, network coding, throughput optimality

MobiCom'07, September 9–14, 2007, Montréal, Québec, Canada.

1. INTRODUCTION

Wireless multi-hop networks have been advocated as an efficient and affordable solution for providing access to the Internet. But, unlike wired networks, wireless networks are resource constrained. Hence, researchers and industry alike, strive for designing power efficient and scalable schemes that provide optimized usage of the available bandwidth. Recently, network coding (NC) emerged as a promising technology to this end. Our aim is to investigate possible performance gains through NC and the optimal way of using NC for unicast communication in multi-hop wireless networks.

Though NC was first applied mainly in the context of multicast in wired networks [1, 8], and subsequently in wireless networks [12, 15], it is found to be particularly amicable for enhancing the throughput (the number of packets delivered to the destination per unit time) of wireless networks even for unicast applications [6, 7, 9, 16, 19, 22]. This is mainly due to the broadcast property of wireless channel, meaning that a transmission from a node can potentially be intercepted by all its neighbors.

The throughput gain via NC in case of unicast sessions is typically illustrated using the network shown in Figure 1. Assume that $R_1(t) = R_2(t) = 1$ at all time t. Without NC, packets from a and b arrive at m, and then m transmits these packets to b and a, respectively, one at a time. This process requires 4 transmissions to deliver one packet from each of the sessions. Thus, a throughput λ is achievable if and only if $\lambda \le 1/4$, i.e., if $\lambda \le$ ($>$, resp.) 1/4, then there exists (does not exist, resp.) a scheduling scheme that arbitrates transmissions in various slots such that the throughput λ is provided to each of the sessions. Now, with NC, m XORs two packets, one from each session, and then transmits the XOR-ed packet. Because of the broadcast property, the XOR-ed packet can be received by both a and b simultaneously. Now, nodes a and b recover the desired packet by XOR-ing the received packet from m with their own packet. Thus, only 3 transmissions are required to deliver one packet from each of the sessions. Clearly, λ is achievable iff $\lambda \le 1/3$. The throughput gain of NC is therefore 4/3 in this example.

The promise of potential throughput gain has instigated significant research in designing efficient NC schemes for unicast communication in wireless networks. Following are the two key features of the schemes proposed in the literature: (i) They advocate the use of NC each time an opportunity to combine and broadcast packets is available. Indeed, the schemes are designed to increase the number of NC opportunities through better routing [19] and through opportunistic listening [7]. (ii) Network coding and scheduling schemes are

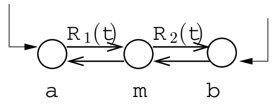

Figure 1: A 3-node network topology handling two sessions, one from a to b and another from b to a. Packets for both the sessions are routed through relay m. The network is symmetric, i.e., the required throughputs for sessions from a to b and from b to a are the same (λ packets/slot), and the rates on the links (a, m) and (m, a) ((b, m) and (m, b), resp.) are equal to $R_1(t)$ ($R_2(t)$, resp.) packets/slot in slot t. Only one of nodes a, b and m can transmit in a slot.

designed separately. We, however, advocate caution in using these features. The main motivation of this paper stems from the following observation regarding the schemes with at least one of these features:

> Systems with NC may have smaller throughputs than those without it.

This observation may seem counter-intuitive as previous work shows that one can only gain by using NC, and the gain can only increase if more opportunities to combine packets are used. We claim that if NC is used each time an opportunity arises or if the scheduling scheme does not account for NC, then the throughput can decrease. We illustrate this fact with two representative examples. In the first example, we fix the scheduling scheme (it provides maximum throughput when NC is not implemented) and demonstrate how applying NC reduces the system throughput. This indicates that NC and scheduling should be jointly considered. In the second example, we compare: (a) the throughput under an optimal scheduling without NC; (b) the throughput of the same system under an optimal scheduling adapted to NC. The scheduling in (b) is optimal subject to using NC at each opportunity. We show again that the throughput decreases when NC is used. This conclusion is more striking than that of the first example as here the scheduling scheme is aware of the NC capabilities.

Example 1: Fading. Consider the network of Figure 1. Let the links experience random fading. Consequently, their rates oscillate randomly and independently between 1 and N: $R_1(t)$ and $R_2(t)$ are independent and identically distributed (i.i.d.), and equal to 1 with probability (w.p.) 1/2, and to N w.p. 1/2. With NC, for correct reception at both a and b, m has to broadcast at rate $\min\{R_1(t), R_2(t)\}$.

First, consider the system without NC and with the following optimal opportunistic scheduling: If $R_1(t) = N = R_2(t)$), schedule link (a, m) w.p. 1/2 and (b, m) w.p. 1/2; if $R_1(t) = 1 = R_2(t)$, schedule each link w.p. 1/4; if $(R_1(t), R_2(t)) = (N, 1)$, schedule (a, m) w.p. 1/4 and (m, a) w.p. 3/4; if $(R_1(t), R_2(t)) = (1, N)$, schedule (b, m) w.p. 1/4 and (m, b) w.p. 3/4. With this scheme, a throughput λ is achievable iff $\lambda \le (1+3N)/16$. When NC is implemented, node m broadcasts XOR-ed packets whenever either (m, a) or (m, b) is scheduled in the above scheme. For the above scheduling scheme with NC, λ is achievable iff $\lambda \le 1/2$ as m always transmits at rate 1 when scheduled. Note that applying NC strictly reduces the throughput if $N > 7/3$.

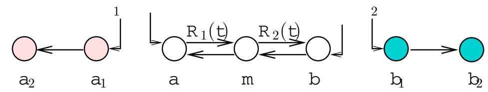

Figure 2: An extension of the network shown in Figure 1. Here, two sessions from a_1 to a_2 and b_1 to b_2 are added, and these require throughputs of λ_1 and λ_2, respectively. The maximum transmission rate on (a_1, a_2) and (b_1, b_2) is 1 packet/slot in each slot. We assume that a_1 (b_1, resp.) can not transmit when a (b, resp.) is either transmitting or receiving. This interference model arises if IEEE 802.11 MAC with RTS and CTS is used.

We make the following two observations on Example 1. (1) Assume that the rates $R_1(t) = r_1$ and $R_2(t) = r_2$ are not time varying, and without loss of generality, let $r_1 \le r_2$. Then, irrespective of the scheduling used, NC provides a higher throughput than that without it. This is because with NC, packets from m to b (faster link) are transmitted along with packets from m to a (slower link). Since the transmissions from m to a have to happen in any case, NC saves transmissions from m to b. (2) With NC, there exists a scheduling scheme that can provide a throughput λ iff $\lambda \le (1 + 3N)/12$ (which is higher than the achievable throughput without NC). The optimal scheme is as follows: If $(R_1(t), R_2(t)) = (N, 1)$, then schedule (a, m); if $(R_1(t), R_2(t)) = (1, N)$, then schedule (b, m);
if $(R_1(t), R_2(t)) = (N, N)$, then broadcast XOR-ed packets from m; if $(R_1(t), R_2(t)) = (1, 1)$, then schedule transmissions from nodes uniformly at random.

From the first observation, it may seem that if the link rates are constant, then NC improves the throughput performance for any topology. And from the second observation, it may seem that if an optimal scheduling with NC is used, then again the throughput increases. But, in the following example, we show that both statements do not hold.

Example 2: Interference. We now provide an example illustrating why taking all opportunities to combine packets may result in throughput reduction, even when an optimal scheduling is used. Consider, in Figure 2, a simple extension of the network shown in Figure 1. Let $R_1(t) = 2$ and $R_2(t) = 1$ for all t, i.e., the rates are fixed but different. Such scenarios are common in wireless networks, e.g., in IEEE 802.11-based mesh networks. One of the possible reasons for having different $R_1(t)$ and $R_2(t)$ is simply because the distances m to a and m to b are different, resulting in different attenuations of the transmitted signals along these links. Now, for correct receptions at both a and b, m has to broadcast combined packets at rate 1. Let the throughput requirements be $\lambda = \lambda_1 = 2/3$ and $\lambda_2 = 1/3$. We claim that the desired throughputs can be provided if NC is not used, while they can not be guaranteed if NC is used. Without NC, to provide the desired throughputs, we can use a scheduling scheme that activates the links (a_1, a_2) and (m, b) simultaneously and (a_1, a_2) and (b, m) simultaneously in 1/3 fraction of slots each, and activates (b_1, b_2) and (m, a) simultaneously and (b_1, b_2) and (a, m) simultaneously in 1/6 fraction of slots each. Now we prove that these throughputs can not be achieved using NC. Indeed, since $\lambda_1 = 2/3$, (a_1, a_2) has to be active in at least 2/3 fraction of slots. As a consequence, (a, m) and (m, a) can be active in at most 1/3 fraction of slots. Thus, to provide a throughput of 1/3 to each of the two sessions that use (a, m) or (m, a), these links must transmit at a rate no less than 2 when active. This is impossible if NC is used as then m broadcasts XOR-ed packets at rate 1 only.

Note that a key feature used in the construction of the above example is that when an XOR-ed packet is transmitted to multiple receivers, all the other nodes in the neighborhood of the receivers have to remain silent: the use of NC reduces the spatial reuse in the network. Hence, for deciding whether to use NC, one has to evaluate the trade-off between the reduction in capacity due to the reduction in the spatial reuse and the capacity improvement due to the broadcast of XOR-ed packets.

Now, we summarize the insights from the above examples. Example 1 shows that NC and scheduling should be jointly designed. Using NC with arbitrary scheduling may result in performance losses. Example 2 shows that the decision to use NC has to be a function of many parameters including the network topology, the link rates and the throughput requirements of the various sessions. This calls for the design of joint NC and scheduling schemes that adapt to the network topology and link rates and provide the required throughput to each session, if doing so is at all possible.

Contributions. In this paper, we present the following contributions:

- We propose a general framework that allows us to characterize the throughput region (the set of achievable throughputs of the various sessions) of networks with NC, and to design optimal and adaptive joint NC and scheduling schemes. The schemes are optimal as they provide the required throughputs, whenever possible. The schemes are adaptive as they take the scheduling and NC decisions based on the current system state only, and do not require the knowledge of channel and arrival statistics a priory.
- We show how our framework can be applied to COPE, a NC scheme recently proposed for unicast sessions in wireless networks [7].
- We also propose a novel NC scheme, XOR-Sym, which exhibits a lower computational complexity than that in COPE. Under XOR-Sym, packets have to be decoded at their destinations only, not at intermediate nodes. In spite of this additional constraint, we show that XOR-Sym and COPE may provide similar throughput gains.

The paper is organized as follows. In Section 2, we present the system model. In Section 3, we characterize the throughput region of networks with NC and provide optimal scheduling policies in a general setting. These results are then applied to specific NC schemes: in Section 4, we design a joint adaptive NC and scheduling scheme for COPE. In Section 5, we propose XOR-Sym, a computationally simple NC scheme, and an associated optimal scheduling scheme. In Section 6, we evaluate the performance of XOR-Sym using simulations. Section 7 discusses possible generalizations. Section 8 provides concluding remarks.

2. SYSTEM MODEL

2.1 Network Topology and Sessions

Consider a multi-hop wireless network, represented as a directed graph $G = (V, E)$, where V and E denote the set of nodes and links, respectively. The network is used by sessions to transport data packets. A session A is characterized by a doublet $(s(A), d(A)) \in V \times V$, where $s(A)$ and $d(A)$ denote the source and the destination, respectively, of A. Note that the sessions are defined at the macroscopic level. Thus, a session can comprise of many applications that communicate between the same source and destination. Let $\mathcal{S}$ denote the set of all sessions. Time is slotted.

We assume that the exogenous packets corresponding to the session A arrive at $s(A)$ as per a stochastic process $\{\lambda_A(t)\}_{t\geq 1}$, where $\lambda_A(t)$ denote the number of packets arriving in slot t. We assume that all the packets have the same length. Exogenous arrivals across the slots are assumed to be i.i.d. Moreover, assume that $\lambda_A(1) \leq c < \infty$ for every A and define $\lambda_A = \mathbb{E}[\lambda_A(1)]$. Packets are stored in infinite buffers until served.

Packets of session $A \in \mathcal{S}$ are routed from $s(A)$ to $d(A)$ in, possibly, multiple hops. We consider fixed routing, and denote by $\mathcal{R}_A$ the route for session A. This route is an ordered subset of V, $\mathcal{R}_A = \{a_0, a_1, \dots, a_{N_A}\}$, such that $a_0 = s(A)$ and $a_{N_A} = d(A)$. For notational convenience, we denote by $e_k^A = (a_k, a_{k+1})$ the $(k+1)$-th link used by packets of session A for every $k \in \{0, \dots, N_A - 1\}$. Furthermore, for every $i \in \mathcal{R}_A$ and $i \neq s(A)$, let $s_i(A)$ denote the node preceding node i on the route of session A, i.e., packets of session A use link $(s_i(A), i)$. Similarly, for every $i \neq d(A)$, $d_i(A)$ denotes the node after node i on route of session A.

For each session $A \in \mathcal{S}$, each node $i \in V$ maintains a queue $q_{i,A}$ to store packets corresponding to this session. All the queues are served in First In First Out (FIFO) order. At the beginning of slot t, the queue length of $q_{i,A}$ is denoted by $Q_{i,A}(t)$, and its Head of Line (HoL) packet by $P_{i,A}(t)$. Finally, let A' be the symmetric session of A, i.e., $s(A) = d(A')$ and $d(A) = s(A')$. Note that the sets of links traversed by the packets of symmetric sessions may not be the same. The notations are illustrated in Figure 3.

2.2 MAC Layer and Scheduling Policies

In networks without NC, a scheduling policy at MAC layer decides, in each slot, which links should be activated and which sessions should be served on these links. In networks with NC, a scheduling policy has to additionally decide whether and how NC should be used. In other words, the policy imposes which nodes should use NC, and which packets should be encoded at these nodes. In this paper, we restrict our attention to NC schemes that allow bit-wise XOR of packets only. Thus, the NC scheme defines the set of possible XORs at each node; but, it is the scheduling scheme that decides whether and when to perform these XORs. For illustration, consider Figure 2. Here, NC scheme facilitates XOR-ing of packets from nodes a and b at node m, but the scheduling policy will arbitrate whether and when to avail this facility. For example, if $\lambda = \lambda_1 = 2/3$ and $\lambda_2 = 1/3$, then a scheduling policy that provides the required throughputs to all the sessions will not XOR packets at m (refer to Example 2); but if $\lambda = 4/3$ and $\lambda_1 = \lambda_2 = 1/3$, then a scheduling policy that provides the required throughputs to all the sessions will XOR packets at m.

Let $\mathcal{L}$ denote the set of L feasible scheduling decisions, or schedules. Each element of $\mathcal{L}$ defines (1) the links that are activated, (2) the sessions that are served on these links, and (3) the sessions whose packets are XOR-ed together. We make the following natural assumption on $\mathcal{L}$.

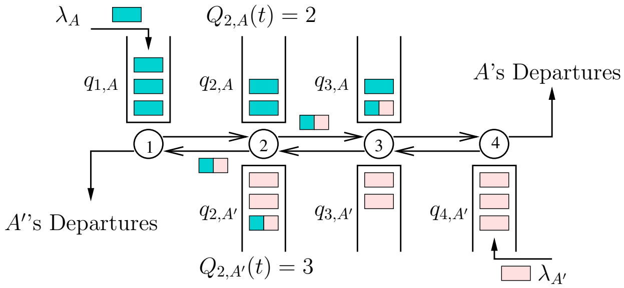

Figure 3: A 4-node linear network with symmetric sessions A and A'. $s(A) = d(A') = 1$ and $s(A') = d(A) = 4$. At $s(A)$ and $s(A')$ new packets arrive at rate λ_A and $\lambda_{A'}$, respectively. Here, $\mathcal{R}_A = \{1, 2, 3, 4\}$, while $\mathcal{R}_{A'} = \{4, 3, 2, 1\}$. Regarding node 2 for example: $s_2(A) = 1$ and $d_2(A) = 3$; 2 FIFO queues, $q_{2,A}$ and $q_{2,A'}$, corresponding to both sessions are maintained; packets from these queues can be XOR-ed and broadcasted to nodes 1 and 3.

Assumption 1. *If $\ell \in \mathcal{L}$, then every ℓ_1 such that the set of active links under ℓ_1 is a subset of that under ℓ also belongs to $\mathcal{L}$.*

The exact nature of $\mathcal{L}$ depends on the MAC and Phy layer constraints, and also on the NC scheme used. We provide the description of $\mathcal{L}$ after presenting the NC schemes considered in Sections 4 and 5. But, for illustration, let us assume that the NC scheme XORs packets corresponding to symmetric sessions only. Then, each schedule $\ell \in \mathcal{L}$ is a subset of $E \times \overline{\mathcal{S}}$, where $\overline{\mathcal{S}} = \mathcal{S} \cup \{A \oplus A', A \in \mathcal{S}\}$. Notation $(e, A) \in \ell$ means that the link $e = (i, j)$ is active and serves queue $q_{i,A}$; $(e, A \oplus A') \in \ell$ means that $e = (i, j)$ is active and serves XOR-ed packets from queues $q_{i,A}$ and $q_{i,A'}$. The MAC and Phy layer constraints further restrict the choice of valid schedules. For example, if the RTS/CTS mechanism is used in IEEE 802.11-based networks and link $e = (i, j)$ is scheduled, then no node in the *neighborhood* of i and j can be scheduled. Thus, $\mathcal{L}$ can not contain a schedule that allows nodes in the neighborhood of i and j to transmit, while simultaneously activating link (i, j). Finally, the set of feasible schedules in a given slot has to reflect the fact that the transmissions from empty queues can not be scheduled.

Definition 1 (Scheduling Policy). *A scheduling policy Δ is an algorithm that chooses a schedule $\ell \in \mathcal{L}$ in each slot t.*

To describe the system states under policy Δ, we use the superscript Δ: for example, $\ell^\Delta(t)$ will denote the schedule chosen by Δ in slot t; $Q^\Delta_{i,A}(t)$ will denote the length of $q_{i,A}$ in slot t under Δ. Let $\mathcal{C}_\mathcal{L}$ denote the class of scheduling policies Δ such that $\ell^\Delta(t) \in \mathcal{L}$ for all t. The class $\mathcal{C}_\mathcal{L}$ also includes the *off-line* policies that arbitrate scheduling by taking into account past, present and even future network states.

2.3 The Phy Layer

Now we present generic models to capture the features of various Phy layer technologies. We categorize the wireless systems into two classes, namely, systems with fixed link rates and systems with adaptive link rates.

2.3.1 Fixed Rate Systems

In such systems, the transmitter and receiver of each link negotiate the link rate during network set-up, and then always use this rate to communicate. Examples of such systems are networks based on the IEEE802.11 standards, where the rate control is performed rarely (at much longer time scale than that of packet transmissions). Let R_e denote the rate negotiated on link e. The variations in channel quality induced by fading and interference can be captured through packet error probabilities (PEP). Specifically, the PEP is the probability that the SINR is above certain level. We denote by $p_{e_k^A}(\ell)$ the PEP on link e_k^A for session A under schedule ℓ. The PEP also depends on t if the model accounts for fading. We assume that $p_{e_k^A}(\ell) = 1$, if session A is not scheduled on e_k^A under ℓ. Now, we give an example to show how the PEP is related to the interference model.

Example 3: The Protocol Model. This model is a generalization of that considered in [5]. A transmission on link $e = (i, j)$ at the negotiated rate is successful if none of the nodes in the set K_e is transmitting. Typically $k \in K_e$, if the distance from k to j is sufficiently small. As a consequence, $p_e(\ell) = 0$, if all nodes in K_e are inactive under ℓ; and $p_e(\ell) = 1$ otherwise. In the network of Figure 3, assume that all links have the same negotiated rate, say 1, and that for link (i, j), the interfering nodes are 1-hop neighbors of j, e.g., $K_{(1,2)} = \{3\}$. Then, for example, when $(1, 2)$ and $(3, 4)$ are simultaneously active, only transmission on $(3, 4)$ is successful. □

2.3.2 Adaptive Rate Systems

In systems with a more elaborated Phy layer, link rates are adapted to the channel conditions and interference (e.g., by using advanced coding capabilities such as Hybrid ARQ). We denote by $R_{e_k^A}(\ell)$ the rate of link e_k^A for session A under schedule ℓ. The link rate also depends on t if the model has to account for fading. We assume that $R_{e_k^A}(\ell) = 0$ if session A is not scheduled on e_k^A under ℓ. Here is an example to show how the link rates relate to the interference model.

Example 4: The SINR-rate Model. Usually the link rate is related to the SINR at the receiver, and it is often well approximated by Shannon formula (up to a multiplicative constant). For example, consider the network of Figure 3, and assume that all nodes transmit at full power, say 1, when scheduled. If links $e_1^A = (1, 2)$ and $e_1^{A'} = (3, 4)$ are active under ℓ in slot t, then, the rate on link e_1^A is:

$$R_{e_1^A}(\ell, t) = W \log\left(1 + \frac{G_{12}(t)}{N_0 + G_{41}(t)}\right),$$

where $G_{ij}(t)$ is the channel gain from i to j in slot t, N_0 is the noise power, and W is the bandwidth. Now, if node 2 broadcasts XOR-ed packet to nodes 1 and 3 under ℓ in slot t, then the rates on these links are:

$$R_{e_2^A}(\ell, t) = R_{e_3^{A'}}(\ell, t) = W \log\left(1 + \min\left\{\frac{G_{21}(t)}{N_0}, \frac{G_{23}(t)}{N_0}\right\}\right).$$

□

Note that when a packet is broadcasted on several links, the rate on each of the links is the minimum of that over all these links. We make the following assumption.

Assumption 2. *Let ℓ_1 be such that the set of active links in ℓ_1 is a subset of that in ℓ. Then, the rate (PEP, resp.) on every active link in ℓ_1 is greater (smaller, resp.) than or equal to that on the same link in ℓ.*

Assumption 2 is typically valid in wireless networks as activating fewer links reduces interference.

2.4 Design Objectives

Our aim is to propose *optimal* joint adaptive NC and scheduling schemes. Next, we introduce various definitions and then state this optimization problem.

Recall that the set of valid schedules $\mathcal{L}$ accounts for the possible NC opportunities, i.e., for the NC scheme. Most of the proposed NC schemes, e.g. COPE, are designed under the constraint that XOR-ed packets must to be decoded at the next hop. Here, we relax this constraint. Thus, encoded packets can be further XOR-ed with other, possibly encoded, packets. Hence, we have to carefully study the decodability of packets. For scalability, we impose that packets are decoded *on the fly*: if the NC scheme decides that an XOR-ed packet P has to be decoded at node i, then i should be able to decode P immediately after it receives P. Thus, the NC schemes considered have to be correct in the following sense.

Definition 2 (Correctness). *Lat a packet P corresponding to session A arrives in $q_{s(A),A}$ at time t and the packets of A are to be decoded at node $i \in \mathcal{R}_A$. Also, let $L = \{\ell(u)\}_{u \geq t}$ denote a sequence of valid schedules after time t such that the first packet containing P (say P') arrives at i in slot t_L. Then, we say that the NC scheme is correct, if i can decode P' to recover P immediately upon arrival of P' for every valid scheduling sequence L.*

Intuitively, the notion of correctness decouples NC scheme and scheduling strategy. Note that NC scheme only affects the set of valid schedules $\mathcal{L}$. But, once $\mathcal{L}$ is defined, NC oblivious scheduling policy can be designed (see Definition 1). Moreover, if the NC scheme is correct, then each packet of every session can be recovered at its respective destination irrespective of the scheduling decisions as the packets of each session A must be decoded at $d(A)$.

Next, we define the performance measures of interest.

Definition 3 (Stability). *The system is stable under Δ, if $\sup_{t\geq 1}\{\mathbb{E}[Q^{\Delta}_{i,A}(t)]\} < \infty$ for every $i \in V$ and $A \in \mathcal{S}$. An arrival rate vector $\boldsymbol{\lambda} = [\lambda_A : A \in \mathcal{S}]$ is said to be stabilizable by Δ, if the system is stable under Δ for $\boldsymbol{\lambda}$.*

Note that stability ensures finite expected delay for every packet. Moreover, in practice, the buffer capacity is finite, though large. In such systems, stability guarantees limited losses due to buffer overflow.

Definition 4 (Throughput Region). *The throughput region of Δ is the set Λ^{Δ} of all the stabilizable rate vectors by Δ. The throughput region of the class of scheduling policies $\mathcal{C}_{\mathcal{L}}$ is $\Lambda_{\mathcal{L}} = \cup_{\Delta\in\mathcal{C}_{\mathcal{L}}}\Lambda^{\Delta}$.*

Definition 5 (Throughput Optimality). *A policy Δ is said to be throughput optimal in class $\mathcal{C}_{\mathcal{L}}$, if $\Lambda^{\Delta} = \Lambda_{\mathcal{L}}$.*

In the next section, we aim at designing a throughput optimal scheme within the class $\mathcal{C}_{\mathcal{L}}$, for a given set $\mathcal{L}$ of valid schedules. Since $\mathcal{L}$ accounts for the NC opportunities offered by the underlying NC scheme, the throughput optimal policy is an optimal *joint NC and scheduling policy.*

3. OPTIMAL SCHEDULING THEOREM

Now, we propose a throughput optimal policy within the class $\mathcal{C}_{\mathcal{L}}$, for any given set of schedules $\mathcal{L}$. In fact, we obtain a more general result: we provide a throughput optimal policy that minimizes certain cost. The cost may, for example, reflect the power consumption in the system, or as explained in Section 6, may also be used to control the packet header size. We use the results derived here to obtain the throughput optimality of the NC and scheduling schemes considered in Sections 4 and 5.

Let $f(\ell)$ denote the cost if schedule ℓ is chosen. We assume that this cost function satisfies:

Assumption 3. *The function $f(\cdot)$ is bounded, and for every ℓ_1 such that the set of activated links under ℓ_1 is a subset of that under ℓ, $f(\ell_1) \leq f(\ell)$.*

Clearly, Assumption 3 holds if $f(\ell)$ is the total power required when schedule ℓ is chosen. Now let the arrival rate vector be $\boldsymbol{\lambda}$. Then, the cost incurred under scheduling policy Δ is: $F^{\Delta}(\boldsymbol{\lambda}) = \limsup_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} f(\ell^{\Delta}(t))$.

Moreover, let $\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})$ denote the set of all policies that stabilizes $\boldsymbol{\lambda}$ using schedules in $\mathcal{L}$. Then, define

$$F^{\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})}_{\min} = \inf_{\Delta\in\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})}\{F^{\Delta}(\boldsymbol{\lambda})\}.$$

Definition 6 (ϵ-Optimality). *A policy Δ is said to be ϵ-optimal for a given $\boldsymbol{\lambda}$, if $\Delta \in \mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})$, and $F^{\Delta}(\boldsymbol{\lambda}) \leq F^{\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})}_{\min} + \epsilon$.*

We propose a policy that is both throughput optimal and ϵ-optimal. Due to space limitations, we obtain the results only for adaptive rate systems. Similar results can be obtained for fixed rate systems by replacing $R_{e^A_k}(\ell)$ with $R_{e^A_k}(1 - p_{e^A_k}(\ell))$ in the following. We analyze systems without random fading. We generalize the analysis to account for fading in Section 7.

3.1 Throughput Region

We first characterize the throughput region of $\mathcal{C}_{\mathcal{L}}$. Let $\mathcal{X}_{\mathcal{L}}$ denote the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exists a vector $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_L]$ such that for all ℓ, $\alpha_\ell \geq 0$, $\sum_{\ell\in\mathcal{L}} \alpha_\ell = 1$, and,

$$\sum_{\ell\in\mathcal{L}} \alpha_\ell R_{e^A_k}(\ell) \geq \lambda_A, \ \forall\, k < N_A \text{ and } \forall A \in \mathcal{S}. \qquad (1)$$

Let $\mathcal{X}^{\circ}_{\mathcal{L}}$ be the set of $\boldsymbol{\nu}$ such that there exists $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}$ with $\boldsymbol{\nu} < \boldsymbol{\lambda}$ coordinate-wise. The following theorem, proved in appendix, characterizes the throughput region of $\mathcal{C}_{\mathcal{L}}$.

Theorem 1. *The throughput region $\Lambda_{\mathcal{L}}$ satisfies $\mathcal{X}^{\circ}_{\mathcal{L}} \subseteq \Lambda_{\mathcal{L}} \subseteq \mathcal{X}_{\mathcal{L}}$. In words, if $\boldsymbol{\lambda} \in \mathcal{X}^{\circ}_{\mathcal{L}}$, then there exists $\Delta \in \mathcal{C}_{\mathcal{L}}$ such that $\boldsymbol{\lambda} \in \Lambda^{\Delta}$, but if $\boldsymbol{\lambda} \notin \mathcal{X}_{\mathcal{L}}$, then $\boldsymbol{\lambda} \notin \Lambda^{\Delta}$ for every $\Delta \in \mathcal{C}_{\mathcal{L}}$.*

3.2 Optimal Policy

Now, we define a parameterized back-pressure based policy denoted by $\Delta^*(\kappa)$, and prove its throughput optimality and ϵ-optimality. Let $\partial Q_{k,A}(t)$ denote the back-pressure along e^A_k, i.e, $\partial Q_{k,A}(t) = Q_{a_k,A}(t) - Q_{a_{k+1},A}(t)$. Depending on the various queue lengths at time t, $\Delta^*(\kappa)$ chooses the schedule defined by:

$$\ell^{\Delta^*(\kappa)}(t) = \arg\max_{\ell \in \mathcal{L}} \left\{ \sum_{A,k} R_{e_k^A}(\ell) \partial Q_{k,A}(t) - \kappa f(\ell) \right\}. \quad (2)$$

THEOREM 2. *For all $\kappa < \infty$, $\Delta^*(\kappa)$ is throughput optimal in $\mathcal{C}_{\mathcal{L}}$. Moreover, for all $\epsilon > 0$, there exists $\widehat{\kappa} > 0$ such that for all $\kappa > \widehat{\kappa}$, $\Delta^*(\kappa)$ is ϵ-optimal.*

The above theorem is proved in appendix. The problem of minimizing cost subject to stability has been studied previously in [3, 14, 20]. However, our result is not a consequence of the results derived there. In [3, 14], the authors analyze one hop sessions only. So, the queueing process is driven primarily by the exogenous arrivals that are independent of the scheduling decisions. Here, however, the queueing process is affected by the chosen schedule as the arrivals in $q_{a_k,A}$ are the departures from $q_{a_{k-1},A}$. In [20], Stolyar has studied multi-hop networks, but under the following assumption: if the set of active links under ℓ_1 is a subset of that under ℓ, then for all links e_k^A activated under both ℓ_1 and ℓ, $R_{e_k^A}(\ell_1) = R_{e_k^A}(\ell)$. This assumption does not hold in typical wireless networks as the link rates depend on the interference caused by the transmissions on other active links. Thus, typically, $R_{e_k^A}(\ell_1) > R_{e_k^A}(\ell)$. In view of these differences, though the nature of our optimal policy $\Delta^*(\kappa)$ is similar to those proposed earlier, the proofs from [3, 14, 20] do not hold here.

Like many other back-pressure based policies proposed in literature [3, 14, 21, 20], $\Delta^*(\kappa)$ is centralized and has high computational complexity. Fortunately, back-pressure based policies are extensively studied, and many schemes for reducing their complexity [3] and for distributed implementations [2, 4, 13, 18] have been proposed. We believe that similar approaches can be developed for the joint NC and scheduling proposed here. We, however, omit the detailed discussion on the design of computationally simple distributed implementations as our aim in this paper is to characterize throughput region of the system that has NC capabilities.

4. OPTIMAL SCHEDULING FOR COPE

Here, we apply the general framework developed in Section 3 to provide an optimal scheduling adapted to COPE, a NC scheme recently introduced in [7].

4.1 Overview of COPE

COPE is a practical NC scheme designed for improving the throughput of unicast sessions in networks with arbitrary topology. In COPE, nodes send XOR-ed combinations of packets that can be decoded at the next hop: a node i sends an XOR-ed packet $P_1 \oplus \ldots \oplus P_m$ only to nodes that already have $m-1$ of m packets $P_1, \ldots, P_m$. When a node j receives an encoded packet, it immediately decodes it. A node j possesses the $m-1$ required packets in two possible scenarios: (i) these packets have been transmitted by j or (ii) j has intercepted these packets by listening to the transmissions (not meant for j) from its neighboring nodes; this is referred to as *opportunistic listening* (OL). Scenario (ii) is possible because of the broadcast nature of the wireless channel. We discuss the implications of (i) and (ii).

Advantages and limitations of OL. When OL is used, nodes may store packets that do not necessarily correspond to sessions routed through them. This creates more opportunities to XOR packets, and thereby potentially increases the sessions' throughputs. Note however that as shown in Example 2, the throughput improvement is not guaranteed. With OL, the increase in NC opportunities comes at the cost of a higher energy consumption. For example, in IEEE 802.11-based networks, to use OL, nodes have to operate in promiscuous mode all the time, and can not enter in sleep mode. Energy consumption in promiscuous mode is significantly higher than that in sleep mode. Moreover, a larger buffer is required as the nodes have to store additional packets. OL also increases the load of signaling messages as each node has to advertise its buffer content. Finally, OL increases the computational complexity as nodes have to process the additional signaling messages from their neighbors to decide how to XOR packets. In view of these limitations, we do not consider OL in this paper. Note here that the performance gain achieved by COPE is higher with OL than that without it, but OL is not an essential feature of COPE [7]. The OL capability merely provides a way to trade performance gain with operational complexity and energy consumption. Thus, the performance of COPE can be studied without considering OL.

Locally Symmetric Sessions. Since, we do not allow OL, a node can have the packets required to decode an encoded packet only if (i) is satisfied. Let packets of sessions A and B be routed through node i. These sessions are said to be *locally symmetric* at node i if $d_i(A) = s_i(B)$ and $s_i(A) = d_i(B)$. In this case, node i can XOR packets from sessions A and B, and send the XOR-ed packet to $d_i(A)$ and $d_i(B)$. The latter nodes will be able to decode the XOR-ed packet as (i) holds.

As illustrated in Example 1, COPE, associated with an arbitrary scheduling policy, may not provide any throughput gain. This calls for the design of a joint NC and scheduling policy that will guarantee that the gains expected from COPE can actually be met. To this aim, we apply the framework of Section 3 and derive a throughput optimal policy adapted to COPE.

4.2 An Optimal Scheduling for COPE

Let us first characterize the set of valid schedules $\mathcal{L}_{\text{COPE}}$ compatible with COPE. Note that COPE is correct only if at most two packets corresponding to locally symmetric sessions are XOR-ed (Theorem 4.1 of [7]). Hence, the set of schedules compatible with COPE is defined as follows: A schedule $\ell \in \mathcal{L}_{\text{COPE}}$ is defined as a subset of $\mathcal{E} = \cup_{e \in E}(e \times \overline{\mathcal{S}(e)})$, where $\overline{\mathcal{S}(e)} = \mathcal{S} \cup \{A \oplus B : A, B$ locally symmetric at i, $e = (i, d_i(A))\}$. Notation $(e, A) \in \ell$ means that the link $e = (i,j)$ is active and serves queue $q_{i,A}$; $(e, A \oplus B) \in \ell$ means that link $e = (i,j)$ is active and serves XOR-ed packets from locally symmetric sessions A and B. Now, schedule ℓ belongs to $\mathcal{L}_{\text{COPE}}$ if it satisfies the following constraints: $\forall A \in \mathcal{S}$ and $\forall e = (i, d_i(A))$,

- if $(e, A) \in \ell$, then for all B, $(e, A \oplus B) \notin \ell$;
- if $(e, A \oplus B) \in \ell$, then $(e', A \oplus B) \in \ell$ where $e' = (i, d_i(B))$, and $(e, A) \notin \ell$, $(e', B) \notin \ell$.

In addition to the above constraints, any schedule ℓ in $\mathcal{L}_{\text{COPE}}$

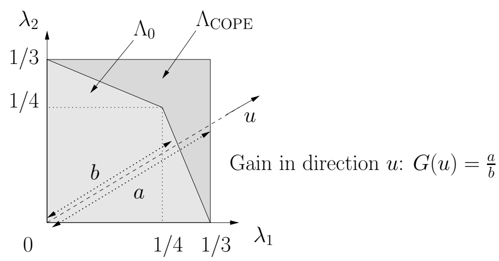

Figure 4: The throughput regions with or without COPE for the network of Figure 3 - The Phy layer follows the Protocol model, and interfering nodes are the 1-hop neighbors only.

has to satisfy the Phy and MAC constraints as illustrated in Section 2.2.

Consider the scheduling policy Δ^*_{COPE} that depending on the queue lengths and link rates, selects, in slot t, schedule ℓ defined as follows:

$$\ell^{\Delta^*_{\text{COPE}}}(t) = \arg \max_{\ell \in \mathcal{L}_{\text{COPE}}} \left\{ \sum_{A,k} R_{e^A_k}(\ell) \partial Q_{k,A}(t) \right\}.$$

We prove that Δ^*_{COPE} has the largest throughput region within the class $\mathcal{C}_1$ of the joint NC and scheduling policies with correct NC and that do not use OL.

THEOREM 3. *The policy Δ^*_{COPE} is throughput optimal in $\mathcal{C}_1$.*

PROOF. Since any correct NC scheme without OL can XOR two packets from locally symmetric sessions only, any $\Delta \in \mathcal{C}_1$ selects schedules from $\mathcal{L}_{\text{COPE}}$. Thus, from Theorem 2, Δ^*_{COPE} is throughput optimal in $\mathcal{C}_1$. □

4.3 Throughput gains of COPE

In general, quantifying the throughput gain achieved with NC is difficult as it depends on many parameters that include the network topology, the underlying Phy and MAC layers and the relative throughput requirements of the sessions. We define the throughput gain by comparing the throughput region of the set of scheduling policies with NC, and the throughput region Λ_0 of policies without NC. The gain achieved by COPE for the network of Figure 3 with a Phy layer satisfying the Protocol model is illustrated in Figure 4. There, $G(u)$ is the gain in direction u, where u is the unit vector representing the relative throughput requirements of the sessions A and A'. The throughput gain is then defined as $\max_u G(u)$.

In [16, 17], the authors characterize the maximum throughput region of 1D networks with NC. In [11], upper bounds on the throughput gains for large random networks are derived. Characterizing the throughput gain with NC for more general topologies is quite challenging. However, even for an arbitrary network, one can use Theorem 1 to characterize the throughput region with or without NC, and numerically compute the throughput gain. A similar approach is used in [19].

Now for COPE, one can easily derive a crude bound on its gain in networks whose Phy layers follow the Protocol model [7]. The maximum gain is 2, and it is achievable. It is achieved for a 1D symmetric network as described in Figure 3 with a large number N of nodes, when the throughput requirements of the two sessions are the same (the gain is achieved in direction $u_1 = u_2$), and when the link rates are all identical. To achieve a gain of 2, it is also necessary that all links interfere with each other (i.e., only one node can transmit at a time), which is a quite unrealistic assumption. In that case, the transmission of one packet of each session from its source to its destination requires $2(N-1)$ slots in absence of COPE and N slots with COPE. The throughput gain is then close to 2 when N is large. Now assume that in this 1D network, each link interferes with its closest neighbors only. Then it can be readily shown that the gain reduces to 4/3 whatever the number N is.

Finally, note that all the studies mentioned above agree that the gain achieved by NC is always upper bounded by a constant, lying typically between 1 and 2, depending on the network topology and the Phy/MAC layers considered.

5. XOR-SYM: A SIMPLIFIED NC SCHEME

In this section, we design a NC scheme that requires a minimal change in the present network architecture and yet provides similar performance benefits as COPE. To this aim, we enforce the following constraint on the type of NC used in the network.

C1: Decoding at Destination Only. A packet corresponding to session A is decoded at $d(A)$ only, and not at any other node.

Many of the NC strategies proposed in the literature (e.g., COPE) require that packets are decoded at each node. Thus, each node has to maintain the packets received and transmitted successfully in the past in order to decode the packets that will arrive in the future. Moreover, whenever an encoded packet arrives, in order to decode it, a node has to perform look-up in its buffer for all but one packets that compose the incoming packet. The look-up may be computationally expensive, if the node has many packets in its buffer. We eliminate this potential bottleneck for scalability of NC schemes by imposing the constraint C1. Intermediate nodes can then remain simple: they only need to perform bit-wise XOR of HoL packets; the required additional functionality can be incorporated without adversely affecting scalability.

In the following, we propose XOR-Sym, a correct NC scheme satisfying the constraint C1 and yet providing throughput benefits.

5.1 The XOR-Sym coding scheme

Figures 5 and 6 provide the pseudo codes for XOR-Sym in the cases of fixed and adaptive rate systems. The key feature of XOR-Sym is that it XORs packets corresponding to symmetric sessions only. Contrast this with COPE which XORs packets corresponding to *locally* symmetric sessions at each node. Due to space limitations, we only describe XOR-Sym for fixed rate systems. Consider the network of Figure 3, whose Phy layer follows the Protocol model and with negotiated link rates all equal to 1 packet/slot (refer to Figure 5 for systems with heterogeneous rates). If the scheduling scheme decides to serve session A only on link $(2,3)$ in slot t, then node 2 transmits $P_{2,A}(t)$. If $P_{2,A}(t)$ is successfully received at node 3, then node 2 discards this packet and replace it with a new packet at the HoL position in $q_{2,A}$. $P_{2,A}(t)$ is queued at the end of $q_{3,A}$. If $P_{2,A}(t)$ is not

```
XOR-Sym for Fixed Rate Systems
begin
  Assume (e, A) ∈ ℓ^Δ(t), where e = (i, d_i(A));
  for j = 1, ..., R_e do
    Transmit P_{i,A}(t);
    if d_i(A) successfully receives the packet then
      Discard P_{i,A}(t);
      P_{i,A}(t) is replaced by the next in line packet in q_{(i,A)};
    end if
  end for
  Assume (e, A ⊕ A') ∈ ℓ^Δ(t), where e = (i, d_i(A));
  for j = 1, ..., min{R_e, R_{e'}} do
    P ← P_{i,A}(t) ⊕ P_{i,A'}(t);
    Broadcast P;
    if Both d_i(A) and d_i(A') successfully receive P then
      Discard P;
      P_{i,A}(t) and P_{i,A'}(t) are replaced by the next in line packets
      in q_{(i,A)} and q_{(i,A')} respectively;
    else if Only d_i(A) successfully receives P then
      Retain P_{i,A'}(t) at HoL position in q_{(i,A')};
      P_{i,A}(t) is replaced by the next in line packet in q_{(i,A)};
    else if Only d_i(A') successfully receives P then
      Retain P_{i,A}(t) at HoL position in q_{(i,A)};
      P_{i,A'}(t) is replaced by the next in line packet in q_{(i,A')};
    else
      Retain both the packets at HoL positions in their respective
      queues;
    end if
  end for
end
```

Figure 5: Pseudo code of XOR-Sym NC scheme with scheduling policy Δ for fixed rate systems - These tasks are performed in each slot.

successfully received at node 3, then it is retained at the HoL position in $q_{2,A}$. Now, suppose that the scheduling scheme decides to broadcast an XOR-ed packet from node 2 on links $(2,1)$ and $(2,3)$. Then, 2 broadcasts $P = P_{2,A}(t) \oplus P_{2,A'}(t)$. Three cases arise. (i) Both 1 and 3 receive P successfully. Then, 2 discards these packets, and new packets come to the HoL positions in $q_{2,A}$ and $q_{2,A'}$. P is decoded at node 1, while it is queued at the end of $q_{3,A}$. (ii) Only one of the intended recipients, say node 3, receives P correctly. Then, $P_{2,A}(t)$ is discarded from $q_{2,A}$ and is replaced by a new packet at the HoL position of $q_{2,A}$, while $P_{2,A'}(t)$ is retained at the HoL position in $q_{2,A'}$. P is queued at the end of $q_{3,A}$. The case when only 1 receives P correctly is similar. (iii) Both 1 and 3 do not receive P correctly. Then, both $P_{2,A}(t)$ and $P_{2,A'}(t)$ are retained at HoL positions in $q_{2,A}$ and $q_{2,A'}$.

Since intermediate nodes do not decode packets, encoded packets can be XOR-ed again. For example, in (ii) above, the XOR-ed packet P is queued in $q_{3,A}$ as it is. When P comes to the HoL position in $q_{3,A}$, it can be XOR-ed again with a packet from $q_{3,A'}$. Thus, it is not clear whether the destinations can decode the received packets. In the following lemma, we show that XOR-Sym is correct given that: for each session A, source $s(A)$ keeps all the packets of A that it has already transmitted, and destination $d(A)$ keeps all the packets of A that it could correctly decode.

Lemma 1. *The NC scheme XOR-Sym is correct.*

Proof. We sketch the proof. Let $\{P_k^A\}_{k\geq 1}$ denote the ordered sequence of session A's packets, i.e., P_k^A is transmitted by $s(A)$ before P_{k+1}^A and after P_{k-1}^A. Let $P(k) = P_1 \oplus \cdots \oplus P_{m(k)}$ denote the first packet containing P_k^A arriving at $d(A)$ in slot $t(k)$. Because of FIFO service, clearly, $t(k) \leq t(k+1)$ for every $k \geq 1$. Now, the result follows from the following claim: For every $k \geq 1$, there does not exist $u \in \{1, \ldots, m(k)\}$ and $v > k$ such that $P_v^A = P_u$. Thus, $P(k)$ can only contain packets from session A' or the packets transmitted before P_k^A. The correctness follows by induction on k. □

```
XOR-Sym for Adaptive Rate Systems
begin
  Assume (e, A) ∈ ℓ^Δ(t), where e = (i, d_i(A));
  Transmit R_e(ℓ(t)) packets from q_{(i,A)};
  Discard all the transmitted packets;
  Assume (e, A ⊕ A') ∈ ℓ^Δ(t), where e = (i, d_i(A));
  R ← min{R_e(ℓ⃗(t)), R_{e'}(ℓ⃗(t))}, where e' = (i, d_i(A'));
  for j = 1, ..., R do
    Broadcast P_{i,A}(t) ⊕ P_{i,A'}(t) (at rate R);
    Discard P_{i,A}(t) and P_{i,A'}(t);
    P_{i,A}(t) and P_{i,A'}(t) are replaced by the next in line packets in
    q_{(i,A)} and q_{(i,A')} respectively;
  end for
end
```

Figure 6: Pseudo code of XOR-Sym NC scheme with policy Δ for adaptive rate systems - These tasks are performed in each slot.

5.2 An Optimal Scheduling for XOR-Sym

Since XOR-Sym combines packets only from symmetric sessions, the set of all possible schedules $\mathcal{L}_{\text{XOR-Sym}}$ is as follows. A schedule $\ell \in \mathcal{L}_{\text{XOR-Sym}}$ is a subset of $E \times \overline{\mathcal{S}}$, where $\overline{\mathcal{S}} = \mathcal{S} \cup \{A \oplus A', A \in \mathcal{S}\}$. In addition, if $\ell \in \mathcal{L}_{\text{XOR-Sym}}$, it satisfies the following constraints:

- if $(e, A) \in \ell$, then $(e, A \oplus A') \notin \ell$;
- if $(e, A \oplus A') \in \ell$, then $(e', A \oplus A') \in \ell$ where $e' = (i, d_i(A'))$, and $(e, A) \notin \ell$, $(e', A') \notin \ell$.

Now, consider the scheduling policy $\Delta^*_{\text{XOR-Sym}}$ that, depending on the queue lengths and link rates, selects, in slot t, schedule ℓ defined as follows:

$$\ell^{\Delta^*_{\text{XOR-Sym}}}(t) = \arg \max_{\ell \in \mathcal{L}_{\text{XOR-Sym}}} \left\{ \sum_{A,k} R_{e_k^A}(\ell) \partial Q_{k,A}(t) \right\}.$$

Now, we prove that $\Delta^*_{\text{XOR-Sym}}$ has the largest throughput region within the class $\mathcal{C}_2$ of the joint NC and scheduling schemes with correct NC and that satisfies the constraint C1. Note that $\mathcal{C}_2$ also contains off-line policies.

Theorem 4. *The policy $\Delta^*_{\text{XOR-Sym}}$ is throughput optimal in $\mathcal{C}_2$.*

In view of Theorem 2, the above result follows from the fact that any scheme in $\mathcal{C}_2$ chooses schedules from $\mathcal{L}_{\text{XOR-Sym}}$ in each slot, which is a consequence of the following lemma.

Lemma 2. *Consider a NC scheme satisfying the constraint C1, and assume that it XORs packets from sessions A and B, where $B \neq A'$. Then the NC scheme is not correct.*

Proof. Consider a NC scheme that allows XOR-ing of packets from sessions A and B at node i, where $B \neq A'$ and

$i \in \mathcal{R}_A \cap \mathcal{R}_B$. Also, without loss of generality, let $s(B) \neq d(A)$, and $\mathcal{R}_A = \{a_0, \ldots, a_{N_A}\}$ and $\mathcal{R}_B = \{b_0, \ldots, b_{N_B}\}$ with $a_k = b_j = i$ for some k and j. Furthermore, let P_A be the first packet arriving in $q_{s(A),A}$ at time t. Now, our aim is to construct a sequence of valid schedules such that the first packet containing P_A arriving at $d(A)$ can not be decoded correctly. Then, the result will follow from Definition 2. The construction is as follows: First, let $q_{i,B}$ is empty at t. Then, find the largest $u < j$ such that $q_{b_u,B}$ is non-empty. If no such u exists, then do not schedule any link until a packet (say P_B) arrives in $q_{s(B),B}$. Note that a new packet will arrive at $s(B)$ in finite time w.p. 1 as $\lambda_B > 0$. When a packet arrives in $q_{s(B),B}$, we can choose $u = 0$. Once the value of u is determined, schedule e_u^B, e_{u+1}^B and so on, one at a time until a packet arrives in $q_{i,B}$. Next, schedule a sequence of links $e_0^A, \ldots, e_{k-1}^A$ one at a time so that P_A arrives in $q_{i,A}$. Now, multicast $P_A \oplus P_B$ on e_k^A and e_j^B, which is possible as NC allows XOR-ing of these packet at i. Finally, schedule $e_{k+1}^A, \ldots, e_{N_A-1}^A$ one at a time so that the XOR-ed packet $P_A \oplus P_B$ arrives at $d(A)$. Now, note that because of the FIFO service, P_B is not available at $d(A)$. Thus, P_A can not be recovered at $d(A)$ upon its arrival. This proves the required. □

5.3 Throughput gains of XOR-Sym

Note that for any network, $\mathcal{L}_0 \subseteq \mathcal{L}_{\mathrm{XOR-Sym}} \subseteq \mathcal{L}_{\mathrm{COPE}}$, where $\mathcal{L}_0$ is the set of all feasible schedules without NC. Thus, $\Lambda_0 \subseteq \Lambda_{\mathrm{XOR-Sym}} \subseteq \Lambda_{\mathrm{COPE}}$: the throughput gain achieved with XOR-Sym over policies that do not use NC is greater than 1, but it may be less than that achieved with COPE. The scalability of XOR-Sym compared to that of COPE is obtained at the expense of a smaller throughput region. Note however, that the maximum gain achieved by XOR-Sym and COPE are identical, and are achieved in the 1D network as described at the end of Section 4. Moreover, under XOR-Sym, the computational complexity at intermediate nodes and the throughput gain can be traded by splitting sessions into several logical sessions. For example, consider a network where packets of sessions A and B follow the routes $\mathcal{R}_A = \{1, 2, 3, 4, 5\}$ and $\mathcal{R}_B = \{6, 4, 3, 2, 1\}$. A and B are not symmetric as $d(A) \neq s(B)$, but both these sessions traverse through nodes 1, 2, 3 and 4. Now, let us split each of these sessions into two logical sessions as follows: $A_1 = (1, 4)$, $A_2 = (4, 5)$ and $B_1 = (4, 1)$, $B_2 = (6, 4)$. Note that now A_1 and B_1 are symmetric and their packets can be XOR-ed under XOR-Sym. Thus, splitting sessions will provide a larger throughput region. But, now the intermediate node 4 has to decode packets, increasing its complexity. Note that XOR-Sym and COPE are identical if the sessions are split into several logical sessions, each traversing exactly one link. A technical difficulty with this approach is that the arrivals at the sources of the logical sessions are not i.i.d.; however, the analysis in Section 3 can be extended to this case. Finally, we believe that creating 1-hop logical sessions everywhere (as in COPE) is not necessary to ensure optimal throughput, because most often only few links are bottlenecks in the network. It may be sufficient to define logical sessions so as to maximize the NC opportunities around these links. The logical sessions may also be created adaptively based on the queue length information.

5.4 Limitation of XOR-Sym

In NC schemes, to ensure decodability, the header of each packet contains the identities of all the packets XOR-ed in this packet. For a packet $P = P_1 \oplus \cdots P_m$, we say that its packet header size is m. Now, if two packets of header sizes m and n are XOR-ed, then the header length of the resulting packet is at most $m + n$. With XOR-Sym, since packets are decoded at destinations only, the header sizes can be quite large. Theoretically, it is possible to construct an example where the header size can become arbitrarily large even for networks with simple topologies as in Figure 3; however, as shown in Section 6, we have verified using simulations that in fact, the header size remains modest unless the network becomes heavily loaded. In Section 6, we also propose some solutions to limit the header sizes.

6. NUMERICAL EXPERIMENTS

In this section, we present some numerical experiments verifying the analytical results of the previous sections. We give the performance of XOR-Sym and of the associated optimal scheduling policy $\Delta^*_{\mathrm{XOR-Sym}}$. Due to space limitations, we present results in the case of simple 1D networks. Refer to [?] for results on networks with more general topologies.

Consider a 1D network as depicted in Figure 3 but with N nodes. Interference follows the protocol model, and we assume that the reception at a node is interfered by the transmission of the 1-hop neighbors, i.e., for instance, using the notation of Section 2.3.1, $K_{(i,i+1)} = \{i+2\}$. The negotiated link rates are all equal to 1. It is then easy to prove (see [?]) that the throughput regions with and without XOR-Sym are independent of N and represented in Figure 4. In this example, the NC gain is maximized when the arrival rates of the two symmetric sessions are equal, $\lambda_A = \lambda_{A'}$, and COPE and XOR-Sym provide similar throughput gains.

Figure 7 (top-left) provides the mean end-to-end packet delay as a function of the session rate for $\Delta^*_{\mathrm{XOR-Sym}}$. The results are compared with those obtained without NC, but with a throughput optimal policy. Note that as expected, these schemes achieve maximum throughput, i.e., the mean packet delay is finite for all $\lambda_A < 1/3$ with XOR-Sym, and for all $\lambda_A < 1/4$ without NC. In Figure 7 (top-right) we present the mean packet header size using XOR-Sym. When the network size is small, e.g. $N = 4$, the mean header size remains small unless the system load approaches the stability limit. The header size increases with N.

To reduce the number of packets XOR-ed into a single packet, we associate a cost to the XOR-ing procedure: for any schedule ℓ chosen at time slot t, we denote by $f(\ell, t)$ the total number of packets involved in XORs under ℓ (e.g., if under ℓ, only packets $P_1 \oplus P_2$ and P_3 are XOR-ed, the cost is 3). Note that this cost function does not strictly correspond to the framework of Section 3; but the latter can be readily modified to account for this kind of costs. Figure 7 (Bottom) presents the mean packet header size using the optimal policy $\Delta^*_{\mathrm{XOR-Sym}}(\kappa)$, for different values of κ in a network of $N = 8$ nodes. The choice of κ allows us to tune the trade-off between packet header size and delay.

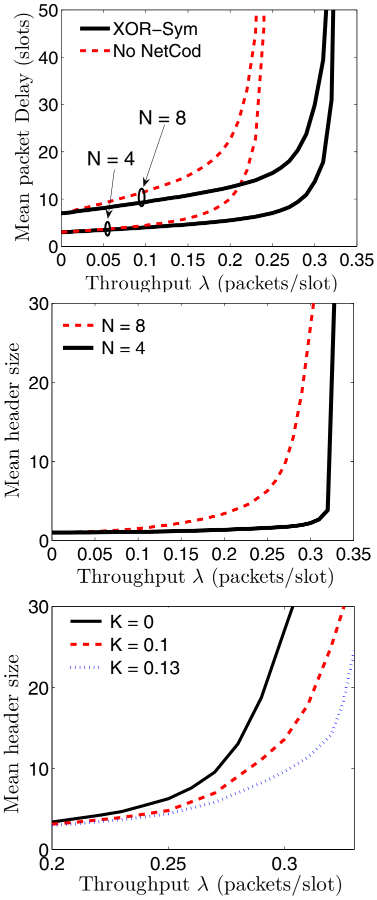

Figure 7: [Top] Mean packet delay as a function of session throughputs $\lambda = \lambda_A = \lambda_{A'}$ with or without XOR-Sym and policy $\Delta^*_{\text{XOR-Sym}}$ - [Middle] Mean packet header size using $\Delta^*_{\text{XOR-Sym}}$ - [Bottom] Mean packet header size using $\Delta^*_{\text{XOR-Sym}}(\kappa)$ for $\kappa = 0$, 0.1, and 0.13, $N = 8$.

7. GENERALIZATIONS

Here we present two generalizations of the framework developed in Section 3. The first extension aims at taking random fading into account, and the second one at providing a joint scheduling and congestion control scheme maximizing some network utility.

7.1 Accounting for fading

In wireless networks, the links are subjected to fading variations, and as a result, the packet loss probability or the achievable rate of these links are time-varying. Here, as in Section 3, we restrict the analysis to adaptive rate systems. To model fading variations, we introduce a finite set of fading states Γ. A fading state $\boldsymbol{\gamma} \in \Gamma$ describes the radio conditions of *all* links, so that the model can capture correlations between the radio conditions on various links. Denote by $R_{e_k^A}(\ell, \boldsymbol{\gamma})$ the service rate of session A on link e_k^A when the fading state is $\boldsymbol{\gamma}$ and when the schedule ℓ is chosen. Now, the fading state evolves according to a stochastic process $\{\boldsymbol{\gamma}(t)\}_{t\geq 1}$ assumed to be stationary ergodic, with equilibrium distribution $[\pi_{\boldsymbol{\gamma}} : \boldsymbol{\gamma} \in \Gamma]$. Denote by $\mathcal{C}_{\mathcal{L}}$ the class of policies that choose schedules in $\mathcal{L}$. Then as in the absence of fading, we can characterize the throughput region of $\mathcal{C}_{\mathcal{L}}$, and provide an optimal adaptive scheduling scheme.

The throughput region of $\mathcal{C}_{\mathcal{L}}$ is $\mathcal{X}_{\mathcal{L}}$, the set of all arrival rate vectors $\boldsymbol{\lambda}$ for which there exist vectors $\boldsymbol{\alpha}_{\boldsymbol{\gamma}} = [\alpha_{\boldsymbol{\gamma},1} \ \cdots \ \alpha_{\boldsymbol{\gamma},L}]$, $\boldsymbol{\gamma} \in \Gamma$ such that for all $\ell, \boldsymbol{\gamma}$, $\alpha_{\boldsymbol{\gamma},\ell} \geq 0$, $\sum_{\ell\in\mathcal{L}} \alpha_{\boldsymbol{\gamma},\ell} = 1$, and

$$\sum_{\ell\in\mathcal{L}} \sum_{\boldsymbol{\gamma}\in\Gamma} \pi_{\boldsymbol{\gamma}} \alpha_{\boldsymbol{\gamma},\ell} R_{e_k^A}(\ell, \boldsymbol{\gamma}) \geq \lambda_A, \ \forall k < N_A \text{ and } \forall A \in \mathcal{S}.$$

An optimal scheduling policy $\Delta^*(\kappa)$ chooses in slot t, the schedule defined by:

$$\ell^{\Delta^*(\kappa)}(t) = \arg\max_{\ell\in\mathcal{L}} \left\{ \sum_{A,k} R_{e_k^A}(\ell, \boldsymbol{\gamma}(t)) \partial Q_{k,A}(t) - \kappa f(\ell) \right\}.$$

7.2 Maximizing network utility

When the network handles data traffic, the rates $\boldsymbol{\lambda}$ of sessions should be adapted to the level of congestion in the network. Usually, optimization approaches are used to analyze or design congestion control algorithms. More precisely, these algorithms are assumed (or designed so as) to maximize a certain network utility: $\sum_{A\in\mathcal{S}} U(\lambda_A)$, where U is an increasing and strictly concave function. The framework developed in Section 3 can be extended so as to design a joint congestion control, NC and scheduling policy that maximizes the network utility while stabilizing all buffers. This type of extensions has been already proposed in the literature in case of networks without NC, see e.g. [10]. This policy is obtained by adding the following rate control algorithm to the scheduling strategy $\Delta^*(0)$ (no cost function is considered). Each session A releases, in slot t, an amount of packets $\lambda_A(t)$ defined by:

$$\lambda_A(t) = \arg\max_{0\leq r\leq c} \left\{ U(r) - r \sum_{k=0}^{N_A-1} Q_{a_k,A}(t) \right\}.$$

8. CONCLUSION

We have investigated the use of network coding (NC) in wireless multi-hop networks for unicast sessions. Surprisingly, we could build simple and realistic examples of networks where NC reduces the throughput performance. This happens when the NC schemes are greedy in the sense that all opportunities to combine and broadcast packets are exploited. We have also observed that if NC and scheduling are designed separately, then the throughput gain expected from NC may not be achieved.

These observations have demonstrated the need of adaptive schemes that use NC opportunities only when they can provide performance benefits. It seems also critical that the scheduling choices and the NC decisions should be coupled. Hence, we have developed a generic framework to design joint optimal NC and scheduling schemes. We have applied this framework to propose an optimal scheduling scheme adapted to COPE, a recent and popular NC scheme. We have also designed XOR-Sym, a new NC scheme, and its associated optimal scheduling scheme. XOR-Sym exhibits a lower complexity than that of COPE but yet offers similar performance gains.

Acknowledgments

The authors would like to thank Prof. Michael Johansson for many fruitful discussions in the course of this work, and also for funding the visit of Dr. Prasanna Chaporkar to KTH, Stockholm, Sweden. Dr. Alexandre Proutiere was with KTH while preparing this paper.

9. REFERENCES

[1] R. Ahlswede, N. Cai, S. Li, and R. Yeung. Network information flow. *IEEE Trans. on Information Theory*, 46(4):1204–1216, 2000.

[2] A. Brzezinski, G. Zussman, and E. Modiano. Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach. In *Proc. of ACM Mobicom*, 2006.

[3] P. Chaporkar and S. Sarkar. Stable scheduling policies for maximizing throughput in generalized constrained queueing. In *Proc. of IEEE Infocom*, 2006.

[4] A. Gupta, X. Lin, and R. Srikant. Low-complexity distributed scheduling algorithms for wireless networks. In *26th IEEE International Conference on Computer Communications(INFOCOM'07)*, pages 1631 – 1639, May 2007.

[5] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Trans. on Information Theory*, 46(2):388–404, 2000.

[6] T. Ho and R. Koetter. Online incremental network coding for multiple unicasts. In *Proc. of DIMACS Working Group on Network Coding*, 2005.

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

[8] S.-Y. R. Li, R. W. Yeung, and N. Cai. Linear network coding. *IEEE Trans. on Information Theory*, 49(2):371–381, 2003.

[9] Z. Li and B. Li. Network coding: The case of multiple unicast sessions. In *Proc. of Allerton Conference*, 2004.

[10] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Trans. on Networking*, 14(2):302–315, 2006.

[11] J. Liu, D. Goeckel, and D. Towsley. Bounds of the gain of network coding and broadcasting in wireless networks. In *Proc. of IEEE Infocom*, 2007.

[12] D. S. Lun, N. Ratnakar, R. Koetter, M. Medard, E. Ahmed, and H. Lee. Achieving minimum-cost multicast: A decentralized approach based on network coding. In *Proc. of IEEE Infocom*, 2005.

[13] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proc: of ACM Sigmetrics / IFIP Performance*, 2006.

[14] M. Neely. Energy optimal control for time varying wireless networks. In *IEEE Infocom*, 2005.

[15] A. Ramamoorthy, J. Shi, and R. Wesel. On the capacity of network coding for random networks. *IEEE Trans. on Information Theory*, 51(8):2878–2885, 2005.

[16] Y. E. Sagduyu and A. Ephremides. Network coding in wireless queueing networks: Tandem network case. In *Proc. of IEEE ISIT*, 2006.

[17] Y. E. Sagduyu and A. Ephremides. Some optimization trade-offs in wireless network coding. In *Proc. of CISS*, 2006.

[18] S. Sanghavi, L. Bui, and R. Srikant. Distributed link scheduling with constant overhead. In *ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, pages 313 – 324, San Diego, CA, 2007.

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

[20] A. Stolyar. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queueing Systems*, 50(4):401–457, 2005.

[21] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multi-hop radio networks. *IEEE Trans. on Automatic Control*, 37(12):1936–1949, 1992.

[22] 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 Research, 2004.

APPENDIX

A. PROOFS OF THEOREMS 1 AND 2

First we state the supporting lemmas that we use to prove Theorems 1 and 2. Due to space constraints, proofs for some of the lemmas are omitted.

LEMMA 3. *The throughput region satisfies* $\Lambda_{\mathcal{L}} \subseteq \mathcal{X}_{\mathcal{L}}$.

Let $\overline{\mathcal{C}_{\mathcal{L}}}$ denote the space of all the policies that are allowed to schedule transmission from $q_{i,A}$ in slot t even when $Q_{i,A}(t) = 0$. Scheduling a transmission from an empty queue corresponds to transmitting a pseudo packet. The pseudo packets are immediately discarded by the receiving node. We note that we use such policies only to obtain a compact proof of the optimality of Δ^*, and we do not allow policies to schedule transmissions from empty queues. Note that $\mathcal{C}_{\mathcal{L}} \subseteq \overline{\mathcal{C}_{\mathcal{L}}}$.

LEMMA 4. *If* $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^{\circ}$, *then there exists* $\delta_1 > 0$ *such that for every* $0 < \delta < \delta_1$, *there exists* $\boldsymbol{\alpha} = [\alpha_1 \ \cdots \ \alpha_L]$ *that satisfies*

$$\sum_{\ell=1}^{L} \alpha_\ell R_{e_k^A, A}(\ell) = \lambda_A + (k+1)\delta, \ \forall\, k < N_A \text{ and } A \in \mathcal{S}$$

$$\sum_{\ell=1}^{L} \alpha_\ell = 1, \text{ and } \ \alpha_\ell \geq 0 \ \text{ for every } \ell.$$

PROOF. The result follows from Assumption 1, and the fact that for every $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^{\circ}$, there exists $\delta_1 > 0$ and such that for every $0 < \delta < \delta_1$, $\boldsymbol{\lambda} + \delta \in \mathcal{X}_{\mathcal{L}}^{\circ}$. □

For rest of the section, fix any $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^{\circ}$. Also, fix $\delta_1 > 0$ that satisfies the conditions in Lemma 4. Now, we define a randomized policy $\Delta_1(\delta) \in \overline{\mathcal{C}_{\mathcal{L}}}$ as follows. Under $\Delta_1(\delta)$, $\ell^{\Delta_1(\delta)}(t) = \ell$ w.p. α_ℓ in every slot t independent of the queue lengths and the decisions in the previous slots. The vector $\boldsymbol{\alpha}$ is a solution of the following Linear Program (LP) for $0 < \delta < \delta_1$.

LP(δ) :- Minimize: $U(\delta) = \sum_{\ell=1}^{L} \alpha_\ell f(\ell)$
Subject to:
1) $\sum_{\ell=1}^{L} \alpha_\ell R_{e_k^A, A}(\ell) = \lambda_A + (k+1)\delta$ for every $k < N_A$ and $A \in \mathcal{S}$
2) $\sum_{\ell=1}^{L} \alpha_\ell = 1$, and $\alpha_\ell \geq 0$ for every ℓ.

LEMMA 5. *For every $\epsilon > 0$ there exists $\delta_\epsilon > 0$ such that for every $\delta < \delta_\epsilon$, $F_{\min}^{\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})} \geq U(\delta) - \epsilon$ w.p. 1.*

PROOF. First, we show that $U(0) \leq F_{\min}^{\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})}$ w.p. 1. Let $\Delta \in \overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})$. Such Δ exists because of Lemma 4. Fix any non-trivial sample path. Let γ_ℓ denote the fraction of time ℓ is scheduled by Δ. Clearly, by stationarity of Δ,

$$\sum_{\ell=1}^{L} \gamma_\ell = 1, \text{ and } \gamma_\ell \geq 0 \text{ for every } \ell. \qquad (3)$$

Then, by stability of Δ, we also know that

$$\lambda_A = \sum_{\ell=1}^{L} \gamma_\ell R_{e_k^A, A}(\ell), \forall\, k < N_A \text{ and } A \in \mathcal{S}, \qquad (4)$$

$$F^\Delta = \sum_{\ell=1}^{L} \gamma_\ell f(\ell). \qquad (5)$$

Equations (3) and (4) show that $\boldsymbol{\gamma}$ is a feasible solution for **LP(0)**. Moreover, (5) is the objective function for **LP(0)**. Since Δ is an arbitrary policy in $\overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})$, we conclude that

$$U(0) \leq F_{\min}^{\overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})} \text{ w.p. 1.} \qquad (6)$$

Since the feasible set of **LP(δ)** is convex and compact, and $f(\ell)$ is a bounded function, by continuity, we conclude that $U(\delta) \to U(0)$ as $\delta \to 0$. Thus, for every $\epsilon > 0$ there exists $\delta_\epsilon > 0$ such that for every $\delta < \delta_\epsilon$, $U(\delta) \leq U(0) + \epsilon$. From (6), we conclude that

$$U(\delta) \leq F_{\min}^{\overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})} + \epsilon \text{ w.p. 1.} \qquad (7)$$

Now, since $\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda}) \subseteq \overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})$, the result follows. ☐

Note that $F^{\Delta_1(\delta)} = U(\delta)$. Thus, Lemma 5 forges the first link between the costs under policies in $\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})$ and that under $\overline{\mathcal{C}_{\mathcal{L}}}(\boldsymbol{\lambda})$. Let us denote the drift in the backlog of $q_{k,A}$ in slot t under Δ as $\partial R_{k,A}^\Delta(t)$, i.e.,

$$\partial R_{k,A}^\Delta(t) = \begin{cases} R_{e_{k-1}^A, A}(\ell^\Delta(t)) - R_{e_k^A, A}(\ell^\Delta(t)) & : k \neq 0, N_A \\ \Lambda_A(t) - R_{e_k^A, A}(\ell^\Delta(t)) & : k = 0 \\ 0 & : k = N_A. \end{cases}$$

Now, consider any $\Delta \in \mathcal{C}_{\mathcal{L}}$ and observe that

$$Q_{a_k,A}^\Delta(t+1) = \max\left\{Q_{a_k,A}^\Delta(t) + \partial R_{k,A}^\Delta(t), 0\right\}. \qquad (8)$$

$$\text{Let } \xi^\Delta(t) \stackrel{\text{def}}{=} \sum_{k,A} \left[\left(Q_{a_k,A}^\Delta(t+1)\right)^2 - \left(Q_{a_k,A}^\Delta(t)\right)^2\right].$$

With (8) and some elementary algebra, it follows that

$$\begin{aligned} \mathbb{E}[\xi^\Delta|\boldsymbol{Q}] \leq\ & Z + 2\sum_A Q_{a_0,A}\lambda_A - 2\kappa\mathbb{E}[f(\ell^\Delta)|\boldsymbol{Q}] \\ & -2\mathbb{E}\left[\sum_{k,A} R_{e_k^A,A}(\ell^\Delta)\partial Q_{k,A} - \kappa f(\ell^\Delta)|\boldsymbol{Q}\right] \end{aligned} \qquad (9)$$

where $Z = |\mathcal{S}|(c^2 + R_{\max}^2)$. We have omitted t for notational simplicity. Then,

LEMMA 6. *Given the queue lengths, $\Delta^*(\kappa)$ maximizes the last term in (9) among all the policies in $\overline{\mathcal{C}_{\mathcal{L}}}$.*

Now, from Lemma 6 and (9), we conclude that

$$\begin{aligned} & \mathbb{E}[\xi^{\Delta^*(\kappa)}|\boldsymbol{Q}] \\ & \leq Z + 2\sum_A Q_{a_0,A}\lambda_A - 2\kappa\mathbb{E}[f(\ell^{\Delta^*(\kappa)})|\boldsymbol{Q}] \\ & \quad -2\mathbb{E}\left[\sum_{k,A} R_{e_k^A,A}(\ell^{\Delta_1(\delta)})\partial Q_{k,A} - \kappa f(\ell^{\Delta_1(\delta)})|\boldsymbol{Q}\right] \end{aligned} \qquad (10)$$

Since the choice of schedule is independent of the queue lengths under $\Delta_1(\delta)$, it follows that

$$\mathbb{E}\left[f(\ell^{\Delta_1(\delta)})|\boldsymbol{Q}\right] = \mathbb{E}\left[f(\ell^{\Delta_1(\delta)})\right] = \sum_\ell \alpha_\ell f(\ell) = U(\delta).$$

$$\mathbb{E}\left[R_{e_k^A,A}(\ell^{\Delta_1(\delta)}))|\boldsymbol{Q}\right] = \sum_\ell \alpha_\ell R_{e_k^A,A}(\ell) = \lambda_A + (k+1)\delta.$$

Substituting the above quantities in (10), we obtain

$$\begin{aligned} & \mathbb{E}[\xi^{\Delta^*(\kappa)}|\boldsymbol{Q}] \\ & \leq Z - 2\kappa\mathbb{E}[f(\ell^{\Delta^*(\kappa)})|\boldsymbol{Q}] - 2\delta\sum_{k,A} Q_{a_k,A} + 2\kappa U(\delta). \end{aligned} \qquad (11)$$

Note that the process $\{\boldsymbol{Q}^{\Delta^*(\kappa)}(t)\}_{t\geq 1}$ is a Markov chain. Thus, to show stability under $\Delta^*(\kappa)$, it suffices to show that the queue length process is positive recurrent.

LEMMA 7. *For every $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^\circ$, $\{\boldsymbol{Q}^{\Delta^*(\kappa)}(t)\}_{t\geq 1}$ is positive recurrent for every $\kappa < \infty$.*

PROOF. Note that $\mathbb{E}[\xi^{\Delta^*(\kappa)}|\boldsymbol{Q}]$ denote the expected Lyapunov drift. From (11), $\kappa < \infty$ and finite support of $f(\cdot)$, it follows that $\mathbb{E}[\xi^{\Delta^*(\kappa)}|\boldsymbol{Q}] < \infty$ for every $\boldsymbol{Q}$. Moreover, $\mathbb{E}[\xi^{\Delta^*(\kappa)}|\boldsymbol{Q}] < -1$ whenever $\sum_{k,A} Q_{a_k,A} > (Z + \kappa U(\delta) + 1)/2\delta$. Thus, the positive recurrence follows from Foster's Theorem. ☐

LEMMA 8. *For all $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^\circ$, $F^{\Delta^*(\kappa)}(\boldsymbol{\lambda}) \leq \frac{Z}{2\kappa} + U(\delta)$ w.p. 1.*

PROOF. From Lemma 7, for every $\boldsymbol{\lambda} \in \mathcal{X}_{\mathcal{L}}^\circ$, the queue length is stationary under $\Delta^*(\kappa)$. Thus, the result follows by taking the expectation in (11) with respect to stationary distribution of the queue length process, and observing from the Renewal Reward Theorem that $\mathbb{E}[f(\ell^{\Delta^*(\kappa)})] = F^{\Delta^*(\kappa)}(\boldsymbol{\lambda})$ w.p. 1. ☐

Now, we prove Theorems 1 and 2.

A.1 Proof of Theorem 1

PROOF. (Theorem 1) From Lemma 7, $\mathcal{X}_{\mathcal{L}}^\circ \subseteq \Lambda^{\Delta^*(\kappa)} \subseteq \Lambda_{\mathcal{L}}$ for every $\kappa \geq 0$. Thus, the result follows from Lemma 3. ☐

A.2 Proof of Theorem 2

PROOF. (Theorem 2) From Lemma 7, $\Delta^*(\kappa)$ is throughput optimal for every $\kappa < \infty$. Now, we show ϵ-optimality. Fix $\epsilon > 0$. From Lemma 5, choose $\delta > 0$ such that $U(\delta) - F_{\min}^{\mathcal{C}_{\mathcal{L}}(\boldsymbol{\lambda})} \leq \epsilon/2$. Also, choose $\widehat{\kappa}$ such that $Z/\widehat{\kappa} = \epsilon$. Now, ϵ-optimality follows from Lemma 8 for every $\kappa > \widehat{\kappa}$. ☐