Cross-Layer Optimization for Wireless Multihop Networks with Pairwise Intersession Network Coding

Abdallah Khreishah, Chih-Chun Wang, and Ness B. Shroff

***Abstract*—For wireless multi-hop networks with unicast sessions, most coding opportunities involve only two or three sessions as coding across many sessions requires greater transmission power to broadcast the coded symbol to many receivers, which enhances interference. This work shows that with a new flow-based characterization of *pairwise intersession network coding* (coding across two unicast sessions), an optimal joint coding, scheduling, and rate-control scheme can be devised and implemented using only the binary XOR operation. The new scheduling/rate-control scheme demonstrates provably graceful throughput degradation with *imperfect scheduling*, which facilitates the design tradeoff between the throughput optimality and computational complexity of different scheduling schemes. Our results show that pairwise intersession network coding improves the throughput of non-coding solutions regardless of whether perfect/imperfect scheduling is used. Both the deterministic and stochastic packet arrivals and departures are considered. This work shows a striking resemblance between pairwise intersession network coding and non-coded solutions, and thus advocates extensions of non-coding wisdoms to their network coding counterpart.**

***Index Terms*—Network coding, pairwise intersession network coding, imperfect scheduling, cross-layer optimization.**

I. Introduction

THE INTERFERENCE-HEAVY nature of wireless media presents a great challenge for designing high-throughput wireless multi-hop networks. Recently, many techniques have been developed for enhancing the throughput of wireless multi-hop networks, among which at least two techniques demonstrate promising throughput improvements. The first method is built around the existing "routing" (non-network-coding) concept and focuses on cross-layer design that considers the corresponding network utility maximization problem (see [28], [29] and the reference therein). The second method is network coding, which allows intermediate nodes to perform coding operations in addition to pure packet forwarding. Advantages of network coding are shown both theoretically [1], [15], [22], [25] and empirically [5], [7], [17], [33], [36], especially when the broadcast nature of wireless media is properly exploited. Network coding can be further classified into two different sub-categories: *intrasession* and *intersession network coding*, the former of which focuses on a single multicast session and coding is performed on packets from the same session. The latter considers multiple coexisting sessions and coding is performed on packets across different sessions.

Intrasession network coding is well-understood as the achievable multicast rate is characterized by the min-cut max-flow theorem [1]. This flow-based characterization leads to a natural extension of the cross-layer optimization framework to intrasession network coding, including throughput, utility, and energy optimization as in [8], [31], [35], [42], [43]. Unfortunately, for the most frequent scenario in which only *unicast sessions* are present, the throughput benefits of intrasession network coding vanish.

Intersession network coding provides performance improvement even when only unicast sessions are present. Although its throughput benefit is clearly demonstrated in the butterfly structure [17], [25], the much needed complete characterization of intersession network coding is less understood and some theoretic studies can be found in [10], [18], [24], [26], [30]. Most research hence focuses on capitalizing two particular network substructures that admit the intersession coding benefits: the butterfly structure of any sizes [38] and the wireless *one-hop* coding opportunity (also known as the "wireless cross" topology) [17]. For the butterfly-based approaches, the achievable rate region was studied in [34], [38] and the associated back-pressure algorithm was studied in [11], [14]. The one-hop coding opportunities were first studied and exploited by the COPE protocol in [17]. The simple one-hop nature and the empirical success of COPE has since motivated numerous subsequent works. Some examples include the centralized computation of the achievable rates with scheduling [36], the energy efficient scheduling with opportunistic coding [9], the power and throughput tradeoff between multicasting and unicasting [6], and the maximum number of overhearing opportunities under practical wireless settings [23]. By taking advantage of both the local butterfly structure and the one-hop coding opportunities, a hybrid, practical scheme has been proposed to further improve the throughput of wireless multi-hop networks [33].

In COPE [17], coding can be performed among ≥ 2 sessions especially when opportunistic listening and coding

Manuscript received 1 August 2008; revised 19 February 2009. This work was supported in part by the ARO MURI award W911NF-08-1-0238, AFOSR award FA 9550-07-1-0456, and by the NSF CAREER Award CCF-0845968. This paper was presented in part at the 42nd Conference on Information Sciences and Systems, Princeton, New Jersey, USA, March 19-21, 2008.

A. Khreishah and C.-C. Wang are with Center for Wireless Systems and Applications (CWSA), School of Electrical and Computer Engineering, Purdue University (e-mails: {akhreish,chihw}@purdue.edu).

N.B. Shroff is with the Departments of Electrical and Computer Engineering and Computer Science and Engineering, The Ohio State University (e-mail: shroff@ece.osu.edu).

Digital Object Identifier 10.1109/JSAC.2009.090604.

0733-8716/09/$25.00 © 2009 IEEE

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

is used. In its empirical study using 802.11 [17], 50% of the coding operations are performed over only two symbols. In a similar but energy-aware setting with scheduling [9], less than 1% coding operations is used to combine ≥ 3 symbols. The intuition behind is that coding more symbols together requires greater transmission power to broadcast the coded symbol to more receivers, which enhances interference and affects negatively the throughput of other traffic. As a result, in this work we consider *pairwise intersession network coding* (PINC) that allows network coding only between pairs of coexisting sessions.

In [39], [40], [41], a new necessary and sufficient condition is established for PINC. With a flow-based form similar to the min-cut max-flow theorem for multi-path routing and for intrasession network coding, the new characterization of PINC prompts tighter integration of cross-layer optimization and network coding. The main contribution of this work is two-fold. (i) Based the characterization of PINC, a new distributed, optimal, joint coding, scheduling, and rate-control scheme is devised, which uses only binary XOR operations, admits fully decoupled rate-control/scheduling, and achieves the optimal rates by identifying all pairwise coding opportunities that include the one-hop opportunity and the butterfly structure as special cases. (ii) Optimal scheduling is computationally expensive to achieve even in a pure non-coding paradigm, let alone with network coding. The PINC-based scheme demonstrates provably graceful throughput degradation for imperfect scheduling [28] that facilitates the design tradeoff between the throughput optimality and computational complexity of different scheduling schemes. Our results show that PINC improves the throughput of routing-based solutions regardless of whether perfect/imperfect scheduling is used. The striking resemblance between PINC and non-coding communications thus advocates extensions of non-coding wisdoms to their network coding counterpart.

The *characterization problem* of pairwise intersession network coding is studied in [39], [40], [41] for acyclic ([40], [41]) and cyclic networks ([39]). A new necessary and sufficient condition has been established, and extensive discussions have been made from coding, information-theoretic, and graph-theoretic (topological) perspectives in [39]. The new theoretic finding of [40], [41] is applied to the rate-control problem for *wireline networks* and a decoubled optimal rate-control algorithm is obtained in [19], [20] for static packet arrivals. In this paper we take further advantage of the PINC characterization results and consider the corresponding cross-layer scheduling and rate-control problem for multi-hop wireless networks. A general wireless setting is used in this work, including the static and dynamic arrivals of the packets. We also quantify the impact of *imperfect scheduling* on *pairwise intersession network coded* wireless multi-hop traffic for the first time. This work further bounds the complexity of the commonly-used wireless-to-wireline conversion technique that models wireless broadcast channel (first proposed for intrasession network coding in [31], [43]) in the context of pairwise intersession network coding for multi-hop networks. An XOR-based coding scheme that is more suitable for wireless networks than the one in [19], [20] is also developed in this paper.

Intersession network coding for wireless networks is also considered in [9] by converting the problem into finding intermediate nodes such that intrasession multicast is performed with these nodes being the sources and sinks of the mulicast sessions. Compared to the approach in this work, [9] does not consider all intersession network coding opportunities but on the other hand permits reencoding the decoded packets at the intermediate nodes which is not considered in this paper. To obtain a low complexity algorithm, [9] further limits the encoding and decoding nodes to be one hop away and assumes the node exclusive model. The resulting one-hop algorithm in [9] is a back-pressure algorithm, which generally incurs more delay and takes more time to converge than the path-based approach used in this work as observed in [12], [37]. For comparison, our path based approach fascilitates analyzing the impact of imperfect scheduling for both the deterministic and stochastic arrivals, which is not considered in [9]. On the other hand back-pressure techniques allow better modeling of lossy links as in [9].

The remainder of this paper is organized as follows. Section II introduces the model of wireless networks and the flow-based characterization for PINC. Concrete examples are provided to illustrate the benefits of PINC. For streamlining the discussion, a new distributed PINC code design is relegated to Section V, which is based on binary XOR operations. An optimal joint scheduling/rate-control scheme is provided in Section III, which admits decoupled implementations. Section IV studies the impact of imperfect scheduling on the proposed PINC solution for both deterministic and stochastic models of packet arrivals and departures. The throughput improvement of PINC and its performance under imperfect scheduling are verified by simulation in Section VI. Section VII concludes the paper.

II. Preliminary Results

A. Pairwise Intersession Network Coding (PINC) For Wireline Networks

Consider directed cyclic/acyclic wireline network $G = (V, E)$, in which each edge is able to carry one $\mathsf{GF}(q)$ symbol per unit time (say a second) and the propagation delay is also one second. High-rate links are modelled by parallel edges and long-delay links are modelled by long paths with added auxiliary intermediate nodes. A pair of coexisting unicast sessions (s_1, d_1) and (s_2, d_2) would like to transmit two strings of independently distributed $\mathsf{GF}(q)$ symbols $X_1, \cdots, X_T$ and $Y_1, \cdots, Y_T$ (one string for each session) simultaneously over a given duration of T seconds. Pairwise intersession network coding (PINC) is allowed and packets of these two strings $\{X_t, Y_t : t = 1, \cdots, T\}$ can be arbitrarily mixed in a linear or non-linear fashion.

We say a PINC solution exists for transmitting two rate-1 strings of packets (over the given unit-edge-capacity network), if given any $\epsilon > 0$, there exists a sufficiently large T such that

$$\frac{1}{T} I([X]_1^T; [M_{d_1}]_1^T) > (1-\epsilon)\log(q)$$
$$\text{and} \quad \frac{1}{T} I([Y]_1^T; [M_{d_2}]_1^T) > (1-\epsilon)\log(q),$$

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

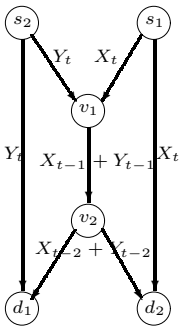


Fig. 1. An acyclic butterfly network that satisfies condition 2 of Theorem 1 and the corresponding coding operations.

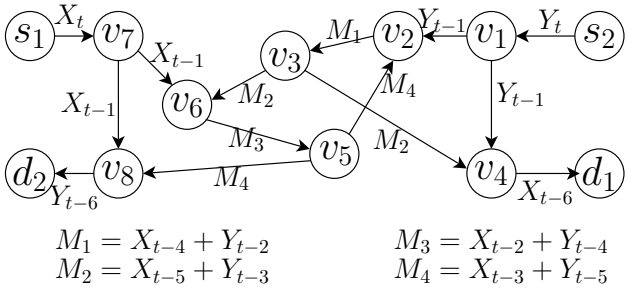


Fig. 2. A cyclic butterfly network that satisfies condition 2 of Theorem 1 and the corresponding coding operations.

where $[X]_1^T \triangleq \{X_1, \cdots, X_T\}$, $[Y]_1^T \triangleq \{Y_1, \cdots, Y_T\}$, $I(\cdot;\cdot)$ is the mutual information and $[M_{d_i}]_1^T$ is the symbols received by destination d_i for $i = 1, 2$.

In [39], a graph-theoretic characterization for the existence of a PINC solution has been established for directed cyclic networks. (Discussion for acyclic networks can be found in [40], [41].) For the following, we use $P_{u,v}$ to represent a path connecting nodes u and v.

Theorem 1: A PINC solution exists if and only if one of the following two conditions holds.

- Condition 1: There exist two edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2}.
- Condition 2: There exist six paths grouped into two sets $\mathcal{P} = \{P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}\}$ and $\mathcal{Q} = \{Q_{s_2,d_2}, Q_{s_1,d_1}, Q_{s_1,d_2}\}$ such that for all $e \in E$,

$$1_{\{e\in P_{s_1,d_1}\}} + 1_{\{e\in P_{s_2,d_2}\}} + 1_{\{e\in P_{s_2,d_1}\}} \leq 2$$
$$\text{and} \quad 1_{\{e\in Q_{s_2,d_2}\}} + 1_{\{e\in Q_{s_1,d_1}\}} + 1_{\{e\in Q_{s_1,d_2}\}} \leq 2,$$

where $1_{\{\cdot\}}$ is the indicator function.

If condition 1 of the theorem is satisfied, a non-coding solution is sufficient for the problem. On the other hand, if only condition 2 is satisfied, network coding is necessary to achieve simultaneous rate-1 transmission. For example, the acyclic network in Fig. 1 satisfies condition 2 by choosing $P_{s_1,d_1} = s_1v_1v_2d_1$, $P_{s_2,d_1} = s_2d_1$, $P_{s_2,d_2} = s_2v_1v_2d_2$, $Q_{s_1,d_1} = s_1v_1v_2d_1$, $Q_{s_1,d_2} = s_1d_2$, and $Q_{s_2,d_2} = s_2v_1v_2d_2$. A unit rate can be supported between (s_1, d_1) and (s_2, d_2), if the coding scheme represented in the figure is used. Fig. 2 contains a cyclic network that satisfies condition 2 of the theorem by choosing $P_{s_1,d_1} = s_1v_7v_6v_5v_2v_3v_4d_1$, $P_{s_2,d_1} = s_2v_1v_4d_1$, $P_{s_2,d_2} = s_2v_1v_2v_3v_6v_5v_8d_2$, $Q_{s_1,d_1} = s_1v_7v_6v_5v_2v_3v_4d_1$, $Q_{s_1,d_2} = s_1v_7v_8d_2$, and $Q_{s_2,d_2} = s_2v_1v_2v_3v_6v_5v_8d_2$. The corresponding coded symbols carried by each edge are also illustrated in Fig. 2, and rate-1 is sustainable for both unicast sessions (s_1, d_1) and (s_2, d_2). Note that the (ϵ, T) is essential to take into account the delay of each edge as seen in Figs. 1 and 2. Theorem 1 shows that as the existence of non-coding solutions is equivalent to finding edge-disjoint paths, the existence of PINC solutions is equivalent to finding paths with *controlled edge overlap*. The intuition is that when the packets/paths are not overly using any bottleneck edge (those edges used by three paths), coding enables the information to be transmitted simultaneously for both sessions. The sub-graph G' induced by any six paths satisfying condition 2 of Theorem 1 will be referred as a pairwise intersession coding configuration (PICC). *In a broad sense, a path is the smallest "graph unit" for non-coding multiple session communications while a* PICC *is the smallest "graph unit" when coding across two sessions is permitted.* This work will build an optimal decoupled coding/scheduling/rate-control scheme based on this new PICC unit and study its performance degradation when imperfect scheduling is used.

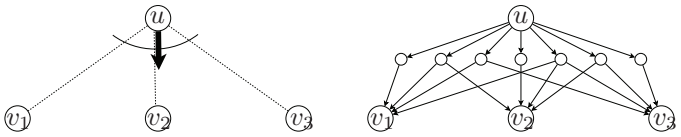
B. Analytical Framework for Wireless Multi-hop Networks — A Wireless to Wireline Conversion

An important feature of wireless multi-hop network is the broadcast nature of wireless media, which is termed the wireless multicast advantage (WMA). The WMA can be modelled as follows (see [31], [43] for details). For each node u with k neighbors $\{v_1, \cdots, v_k\}$, introduce $2^k - 1$ auxiliary nodes such that each auxiliary node corresponds to a non-empty element of the powerset of $\{v_1, \cdots, v_k\}$. Add $2^k - 1$ directed edges connecting u and each of the auxiliary nodes. For each auxiliary node, add directed edges from the auxiliary node to each node in the corresponding subset of neighbors. Fig. 3 illustrates one such conversion for a node with three neighbors. In a wireless network, every time a packet is about to be sent, the sender u chooses the target receiver(s) of the packet, which is equivalent to choosing the corresponding auxiliary node/link for transmission. Therefore, designing a wireless transmission scheme that exploits the advantages of the WMA is equivalent to designing a good routing/scheduling algorithm on its wireline counterpart *with the additional node-exclusive scheduling constraints that auxiliary nodes corresponding to the same u cannot be active simultaneously.* This framework takes into account the WMA and maps the wireless scheduling problem to a wireline scheduling problem while *the underlying interference model for the former is absorbed as scheduling constraints $\mathbf{\Theta}$ for the latter problem,* which will be clear in the later section.

C. Pairwise Intersession Network Coding for Wireless Networks

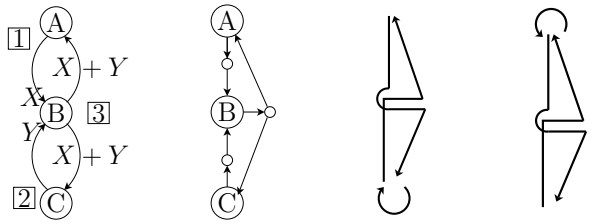
Based on a necessary and sufficient condition, the six-path-based PICC captures *all pairwise coding opportunities* once the aforementioned wireless to wireline conversion is properly exploited, which include the widely studied butterfly structure [11], [14], [33], [34], [38] and the one-hop coding

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.



(a) Wireless broadcast advantage. (b) The corresponding wireline counterpart.

Fig. 3. Modelling the wireless mulitcast advantage.



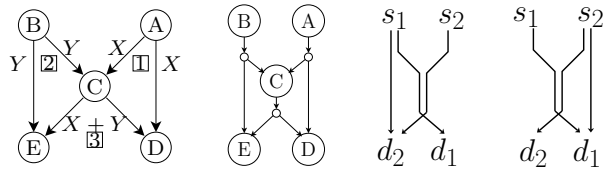
(a) (b) (c) (d)

Fig. 4. A simple wireless one-hop coding opportunity with two sessions: Session 1: $A \xrightarrow{X} C$; Session 2: $C \xrightarrow{Y} A$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1}, P_{s_2,d_2}, and P_{s_2,d_1}. (d) Paths Q_{s_2,d_2}, Q_{s_1,d_1}, and Q_{s_1,d_2}.

opportunities [9], [6], [17], [23], [36] as special cases when coding is permitted only between two sessions.

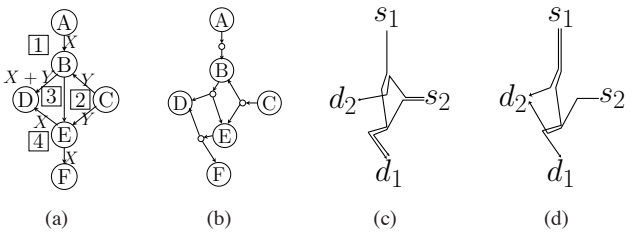
Fig. 4(a) is a classic example of the one-hop intersession coding opportunity for wireless networks. Node A would like to send symbol X to node C while C intends to send Y to A. Fig. 4(a) depicts how to send two symbols in three time slots. The 1 and 2 in the small boxes indicate that A sends X to B in the first time slot while C sends Y to B in the second time slot. In the third time slot, B broadcasts coded symbol $X + Y$ to A and C using the WMA. If we follow the wireless to wireline conversion, Fig. 4(a) is transformed to Fig. 4(b). By noticing the existence of the $\mathcal{P}$ and $\mathcal{Q}$ paths with controlled edge-overlap as in Figs. 4(c) and 4(d), this one-hop coding opportunity for coding across two sessions is captured by Theorem 1 and corresponds to an instance of PICC. Note that Fig. 4(b) also indicates that under a *node exclusive* model at least three time slots are necessary as the three involved auxiliary nodes cannot be scheduled simultaneously.

Similarly Fig. 5(a) describes the classic wireless cross-flows in which symbols X and Y can be sent from A to E and from B to D in three time slots. D and E use the overheard packets X and Y for decoding. Figs. 5(b) to 5(d) depict the corresponding wireless to wireline conversion and show that the wireless cross flows can again be captured as a special instance of PICC (which is actually a butterfly in the corresponding wireline network). Since the path-based characterization of PINC does not require that encoding and decoding happen at nodes that are 1-hop apart from each other, our formulation naturally takes into account coding opportunities over subgraphs of different "sizes," e.g. 2-hop butterflies used in [33]. Capturing *all* pairwise coding opportunities, the PICCs also prompt new wireless intersession coding opportunities different from Figs. 4 and 5. For example, in Fig. 6(a), a new type of wireless cross flows is identified, for which A sends symbol X to F while C sends symbol Y to D. This example is not captured by the traditional one-hop coding opportunity as node D does not overhear the original



(a) (b) (c) (d)

Fig. 5. The wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} E$; Session 2: $B \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1}, P_{s_2,d_2}, and P_{s_2,d_1}. (d) Paths Q_{s_2,d_2}, Q_{s_1,d_1}, and Q_{s_1,d_2}.



(a) (b) (c) (d)

Fig. 6. A new type of wireless cross flows with two sessions: Session 1: $A \xrightarrow{X} F$; Session 2: $C \xrightarrow{Y} D$. (a) The slot-by-slot wireless transmission. (b) Wireless to wireline conversion. (c) Paths P_{s_1,d_1}, P_{s_2,d_2}, and P_{s_2,d_1}. (d) Paths Q_{s_2,d_2}, Q_{s_1,d_1}, and Q_{s_1,d_2}.

symbol X sent by A but overhears the reconstructed symbol X decoded and sent by E. Figs. 6(c) and 6(d) illustrate the corresponding $\mathcal{P}$ and $\mathcal{Q}$ paths, which verify that this new type of wireless cross flows is a special instance of PICC (that is different than the classic butterfly structure). By including the existing coding opportunities as special cases and capturing additional ones, our PICC-based solution will enhance further the achievable capacity region of intersession network coding.

III. Optimal Joint Scheduling/Rate-Control With PINC

Following Section II-B, we model a wireless network by its wireline counterpart denoted by $G = (V, E)$ where V is the set of network nodes plus auxiliary nodes and E is the edge set. Consider slotted transmission, a scheduling policy Θ is a collection of active edges and the associated power levels. Under a given interference model, we use r_e^{Θ} to denote the rate that can be supported on edge e under the scheduling policy Θ, and we often use $\mathbf{r}^{\Theta}$ for the collective rate vector. Let $\mathbf{\Theta}$ denote the collection of all policies and let $\mathcal{R} \triangleq \{\mathbf{r}^{\Theta} : \forall \Theta \in \mathbf{\Theta}\}$ denote the corresponding rates. Any rate vector $\mathbf{r} \in \mathrm{Co}(\mathcal{R})$, the convex hull of $\mathcal{R}$, can be achieved via time sharing. Without loss of generality, we assume the rate region is bounded. There are N coexisting unicast sessions using the network to send data from source s_i to destination d_i where $i = 1, \cdots, N$. The utility function $U_i(x)$ for each session is strictly concave and monotonically increasing, where x is the end-to-end data rate for the session.

The utility optimization for multiple unicast sessions using PINC can be cast as follows.

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

$$\max_{0\le\mathbf{x}\le M_X,\mathbf{r}\in\mathrm{Co}(\mathcal{R})} \sum_{i=1}^{N} U_i\left(\sum_{k=1}^{|\mathcal{P}_i|} x_i^k + \sum_{j:j\neq i}\sum_{l=1}^{|\mathcal{PICC}_{ij}|} x_{ij}^l\right) \quad (1)$$

$$\text{subject to} \quad \sum_{i=1}^{N}\sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e)x_i^k + \sum_{(i,j):i\neq j}\sum_{l=1}^{|\mathcal{PICC}_{ij}|} \frac{H_{ij}^l(e)x_{ij}^l}{2} \le r_e, \forall e\in E \quad (2)$$

$$x_{ij}^l = x_{ji}^l, \quad \forall (i,j): i<j, \forall l \quad (3)$$

where $\mathcal{P}_i$ is the collection of paths from s_i to d_i along which packets will be routed without any coding operations and x_i^k is the rate assigned for the k-th path. $\mathcal{PICC}_{ij}$ is the collection of PICCs between sessions i and j on which intersession network coding will be performed and x_{ij}^l is the packet rate of source s_i that will be network coded using the l-th PICC of $\mathcal{PICC}_{ij}$. Without loss of generality, we further assume the indices of $\mathcal{PICC}_{ij}$ and of $\mathcal{PICC}_{ji}$ are consistent. Namely, for all l, the l-th PICC of $\mathcal{PICC}_{ij}$ is also the l-th PICC of $\mathcal{PICC}_{ji}$. Since in PINC, packets from s_i and s_j are coded bijectively with each other, the system requires the equal-rate constraint (3). Without loss of generality, we also assume that the rate vector $\mathbf{x}$ is bounded by a finitely large constant M_X.

In (2), $H_i^k(e)$ is the indicator function whether the k-th path in $\mathcal{P}_i$ uses edge e. $H_{ij}^l(e)$ is the indicator function whether the l-th PICC in $\mathcal{PICC}_{ij}$ uses edge e. PINC ensures that the two packet flows (with rates x_{ij}^l and x_{ji}^l respectively) jointly use only the max rate $\max(H_{ij}^l(e)x_{ij}^l, H_{ji}^l(e)x_{ji}^l)$ instead of the sum rate $H_{ij}^l(e)x_{ij}^l + H_{ji}^l(e)x_{ji}^l$. By the fact that the indicator function is symmetric by definition, i.e. $H_{ij}^l(e) = H_{ji}^l(e)$, and by the equal-rate constraint in (3), the rate consumption becomes $\max(H_{ij}^l(e)x_{ij}^l, H_{ji}^l(e)x_{ji}^l) = \frac{H_{ij}^l(e)x_{ij}^l + H_{ji}^l(e)x_{ji}^l}{2}$. For each edge e, summing over rates consumed by multi-path routing and by multi-PICC network coding leads to the capacity constraint (2). The non-negative rate vector $\mathbf{x}$, including all x_i^k and x_{ij}^l, is the subject of rate control and the edge rate vector $\mathbf{r}\in\mathrm{Co}(\mathcal{R})$ is the subject of optimal scheduling and time-sharing. One advantage of considering PICC is that unlike the existing butterfly-search approach [11], [14], the characterization of PICC is *path-based* rather than structure-based. One can thus use any path search algorithm to identify possible constituent paths and any six paths $P_{s_1,d_1}, P_{s_2,d_2}, \cdots, Q_{s_1,d_2}$ can serve as a PICC. It is worth pointing out that there is no need to strictly enforce Condition 2 of Theorem 1 during implementation. More explicitly, since each edge e knows whether itself participates in a given path during the path-search phase, e also knows its edge-overlap in the given six paths. Consider an edge e that is a *bottleneck*, i.e.

$$1_{\{e\in P_{s_1,d_1}\}} + 1_{\{e\in P_{s_2,d_2}\}} + 1_{\{e\in P_{s_2,d_1}\}} = 3 \quad (4)$$
$$\text{or } 1_{\{e\in Q_{s_1,d_1}\}} + 1_{\{e\in Q_{s_2,d_2}\}} + 1_{\{e\in Q_{s_1,d_2}\}} = 3.$$

We can still treat the given six paths as a PICC in our optimization problem (1) to (3) even though they do not satisfy Condition 2 of Theorem 1. To that end, we simply need to generalize the indicator function $H_{ij}^l(e)$ and let $H_{ij}^l(e) = 2$ (rather than 1) for any bottleneck edge. In this way, we allocate *double* the capacity for such e (see (2)), which thus resolves the corresponding bottleneck caused by (4). With the use of a generalized indicator $H_{ij}^l(e)$, searching for PICCs is equivalent to searching for paths plus combining six paths as a group, which can be achieved by any path-search algorithms. Note that the larger the path collection $\mathcal{P}_i$ and the PICC collection $\mathcal{PICC}_{ij}$, the higher achievable throughput will be. Depending on the available resources, there thus exists a complexity-performance tradeoff on how *exhaustive* the path-search algorithm should be.

The optimal solution of (1–3) can be achieved in a decoupled way by solving its dual problem via the sub-gradient method.

Algorithm A:

Rate Update For each s_i, update its rate vector $\mathbf{x}_i[t] = \{x_i^k[t], x_{ij}^l[t] : \forall k,j,l\}$ for the t-th time slot by

$$\mathbf{x}_i[t] = \arg\max_{0\le\mathbf{x}_i\le M_X} U_i\left(\sum_{k=1}^{|\mathcal{P}(i)|} x_i^k + \sum_{j:j\neq i}\sum_{l=1}^{|\mathcal{PICC}_{ij}|} x_{ij}^l\right) - \sum_{e\in E} q_e[t]\left(\sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e)x_i^k + \sum_{j:j\neq i}\sum_{l=1}^{|\mathcal{PICC}_{ij}|}\frac{H_{ij}^l(e)x_{ij}^l}{2}\right) - \sum_{l=1}^{|\mathcal{PICC}_{ij}|}\left(\sum_{j:j>i} q_{ij}^l[t]x_{ij}^l - \sum_{j:j<i} q_{ji}^l[t]x_{ij}^l\right) - \alpha_i\left(\sum_{k=1}^{|\mathcal{P}_i|}\left(x_i^k - y_i^k\right)^2 + \sum_{j:j\neq i}\sum_{l=1}^{|\mathcal{PICC}_{ij}|}\left(x_{ij}^l - y_{ij}^l\right)^2\right),$$

where $q_e[t]$ and $q_{ij}^l[t]$ are dual variables at the t-th time slot, whose values are feedback to s_i. The α_i are small constants and $\mathbf{y}_i = \{y_i^k, y_{ij}^l : \forall j,k,l\}$ are auxiliary variables of the proximal method in order to eliminate oscillation [4]. Periodically, $\mathbf{y}_i$ is set to $\mathbf{x}_i[t]$ and the iteration continues using the new $\mathbf{y}_i$.

Scheduling Update The network selects the optimal scheduling policy for the t-th time slot by

$$\mathbf{r}[t] = \arg\max_{\mathbf{r}\in\mathcal{R}} \sum_{e\in E} q_e[t]r_e. \quad (5)$$

Queue-length Update Each link e updates its dual variable $q_e[t+1]$ according to the following equation.

$$q_e[t+1] = \Big[q_e[t] + \beta_e\Big(\sum_{i=1}^{N}\sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e)x_i^k[t] + \sum_{(i,j):i\neq j}\sum_{l=1}^{|\mathcal{PICC}_{ij}|}\frac{H_{ij}^l(e)x_{ij}^l[t]}{2} - r_e[t]\Big)\Big]^+,$$

where $[\cdot]^+ \triangleq \max(\cdot, 0)$ is the projection operator and β_e is a small step size for the sub-gradient method.

Balance Update Each destination d_i updates the dual variable $q_{ij}^l[t+1]$ for all $j > i$. The dual variable q_{ij}^l accounts

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

the difference between packet rates of sources i and j that use the same PICC.

$$q_{ij}^l[t+1] = q_{ij}^l[t] + \beta_i \left(x_{ij}^l[t] - x_{ji}^l[t]\right), \forall j : j > i, \forall l, \quad (6)$$

where β_i is a small step size for the sub-gradient method.

Proximal Update Periodically, after every K time slots, set $\mathbf{y}_i \leftarrow \mathbf{x}_i[t]$. For notational simplicity, after the proximal update, we reset the timer value $t \leftarrow 0$.

The five different parts of Algorithm A are coupled implicitly via the queue lengths q_e and the balance information q_{ij}^l at the destinations. One important observation is that with PINC, only the rate and the balance updates, performed at the sources s_i and destinations d_i, differ from its non-coding counterpart (cf. [27]). The scheduling and queue-length updates remain identical. The impact of PINC on rate-control and scheduling is thus minimal and confined only in sources and destinations.

The complexity of Algorithm A depends mainly on the number of feedback messages $q_{ij}^l[t]$ that each source receives at each time slot t, which is proportional to the number of PICCs in the network. Therefore, the number of queue-length exchange messages is of the order $O(N \times |E| \times (\max_i \sum_{j:i\neq j} |\mathcal{PICC}_{ij}|))$. This is of similar complexity to that of traditional multipath routing with scheduling and congestion control [27] typically proposed for static mesh networks without mobility. Distributed methods that reduce the number of need-to-be-considered PICCs can be found in Section V and in [19], which mitigate the complexity of this algorithm and is executed only once in the initialization phase. Another approach for complexity reduction is to include the paths that form PICCs one by one in an adaptive way such that the number of control messages do not exceed a threshold as explained in [19]. For comparison, the number of multicast sessions used in the framework of [9] is $\binom{N}{2} \times |V| \times (|V|-1)$, and the number of queue-length exchange messages in the corresponding back-pressure algorithm is thus $O(N^2 \times |V|^3 \times \overline{\mathsf{nbs}})$, where $\overline{\mathsf{nbs}}$ is the average number of neighboring nodes for nodes in V.

A. Convergence Analysis of Algorithm A

Proposition 1: Consider a decreasing non-negative sequence $\{\beta_\tau\}$ such that $\sum_{\tau=1}^{\infty} \beta_\tau \to \infty$ and $\sum_{\tau=1}^{\infty} (\beta_\tau)^2 < \infty$. If in the beginning of each proximal iteration, we reset the step sizes $\beta_e = \beta_i = \beta_\tau$ with $\tau = 1$. As the inner iteration proceeds, we use β_τ, $\tau = 1, \cdots, K$ as the step sizes in the K inner iterations. Then when the update period K of the proximal variable $\mathbf{y}_i \leftarrow \mathbf{x}_i[t]$ is sufficiently large, Algorithm A converges to the optimal solution of (1–3), the optimal rate assignment of PINC.

A sketch of the proof is as follows. The boundedness of the rate region $\mathrm{Co}(\mathcal{R})$ and rate-vector $\mathbf{x}$ implies the boundedness of the sub-gradient of the dual problem of (1–3). Proposition 8.2.6 in [3] then guarantees the convergence. A detailed proof is relegated to [21]. The convergence with K bounded away from infinity and β_τ bounded away from zero is empirically verified during our simulations.

B. Stability of Algorithm A

Definition 1: A system load $\{w_i : i = 1, \cdots, N\}$ (we sometimes use $\{w_i\}_i$ as shorthand) can be stabilized by Algorithm A if there exists a non-negative vector $\mathbf{w} = \{w_i^k, w_{ij}^l : \forall i, j, k, l\}$ such that

$$w_i = \sum_{k=1}^{|\mathcal{P}_i|} w_i^k + \sum_{j:j\neq i} \sum_{l=1}^{|\mathcal{PICC}_{ij}|} w_{ij}^l, \forall i,$$
$$\text{and} \quad w_{ij}^l = w_{ji}^l, \forall (i,j) : i < j, \forall l. \quad (7)$$

Moreover, if we replace the "rate update" in Algorithm A by a fixed rate assignment $\mathbf{x}[t] = \mathbf{w}$, then the dual variables $q_e[t]$ and $q_{ij}^l[t]$ must stay bounded away from infinity when t tends to infinity.

Let Λ denote a set of system loads $\Lambda = \{\{w_i\}_i\}$ such that for any $\{w_i\}_i \in \Lambda$, there exists a rate vector $\mathbf{r} \in \mathrm{Co}(\mathcal{R})$, a non-negative vector $\mathbf{w} = \{w_i^k, w_{ij}^l : \forall i, j, k, l\}$ satisfying (7), and jointly $\mathbf{w}$ and $\mathbf{r}$ satisfy

$$\sum_{i=1}^{N} \sum_{k=1}^{|\mathcal{P}_i|} H_i^k(e) w_i^k + \sum_{(i,j):i\neq j} \sum_{l=1}^{|\mathcal{PICC}_{ij}|} \frac{H_{ij}^l(e) w_{ij}^l}{2} \leq r_e, \forall e \in E.$$

We then have the following stability result regarding the system load region Λ.

Proposition 2: Any system load $\{w_i\}$ that is in the interior of Λ can be stabilized by the optimal rate-control/scheduling in Algorithm A.

(Proposition 2 can be regarded as a corollary of Proposition 3 that will be introduced shortly after.)

IV. Pairwise Intersession Network Coding with Imperfect Scheduling

In general, it is computationally expensive to find the optimal scheduling decision satisfying (5) in Algorithm A. Depending on different interference models, finding the optimal scheduling $\mathbf{r}$ that maximizes $\sum_e q_e[t] r_e$ is NP-hard in many cases and generally requires centralized implementation. In practice, we would often have to resort to imperfect scheduling schemes that select the rate vector $\mathbf{r}[t]$ that achieves γ fraction of the maximum value. Namely, an imperfect scheduling policy choose $\mathbf{r}[t]$ satisfying

$$\sum_{e\in E} q_e[t] r_e[t] \geq \gamma \max_{\mathbf{r}} \sum_{e\in E} q_e[t] r_e, \quad (8)$$

where γ is a constant in $[0, 1]$. With imperfect scheduling ($\gamma < 1$), the tie between Algorithm A and the gradient method for the dual problem is severed and Algorithm A may not converge to any fixed-point solution. The following results show that even with imperfect scheduling, the proposed PINC scheme with cross-layer optimization Algorithm A still shows tractable performance in terms of the stability region.

Proposition 3: Any system load $\{w_i\}$ that is in the interior of $\gamma\Lambda$ can be stabilized by Algorithm A with γ-imperfect scheduling.

The sketch of the proof is provided in Appendix A, which covers Proposition 2 as a special case.

A. Networks with Dynamic Arrivals and Departures

In addition to networks with static arrivals and departures, we also consider the case of dynamic system loads with logarithmic utility functions. Consider N classes of users.

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

For all i, users in class i have a common logarithmic utility function $U_i(x) = \kappa_i \log(x)$ where $\kappa_i > 0$ are predefined system parameters. All users in class i will send packets from s_i to d_i and will use the same routing paths in $\mathcal{P}_i$ and the same PICCs in $\mathcal{PICC}_{ij}$ for transmission. We also assume that users of class i arrive according to a Poisson process with rate λ_i and each user needs to send a file whose size is exponentially distributed with mean $\frac{1}{\mu_i}$. The system load of this network with dynamic arrivals is then defined as $\left\{\left(\frac{\lambda_i}{\mu_i}\right) : \forall i\right\}$. The dynamic nature of this setting prompts a slightly different definition of stability.

Definition 2: A system load $\left\{\left(\frac{\lambda_i}{\mu_i}\right) : \forall i\right\}$ can be stabilized by Algorithm A if the dual variables $q_e[t]$ and $q_{ij}^l[t]$ are bounded away from infinity for each iteration with probability one.[1]

We then have the following stability result.

Proposition 4 (Stability for Dynamic Systems): Consider logarithmic utility functions $U_i(x) = \kappa_i \log(x)$. With sufficiently small α_i, β_e, and β_i, any system load $\left\{\left(\frac{\lambda_i}{\mu_i}\right) : \forall i\right\}$ that is in the interior of $\gamma\Lambda$ can be stabilized by Algorithm A with γ-imperfect scheduling.

The proof of Proposition 4 is sketched in Appendix B. Proposition 4 implies that although the instantaneous system load imposed on the network may well exceed the network capacity, as long as the average system load is within γ times the capacity, the queue lengths of the network are bounded away from infinity and the system is stable.

Proposition 4 shows that the graceful stability degradation that was previously known only for non-coding transmission (cf. [28]) also holds for PINC. Shifting from non-coding to network-coding solutions enhances the throughput without sacrificing the associated stability even with imperfect scheduling.

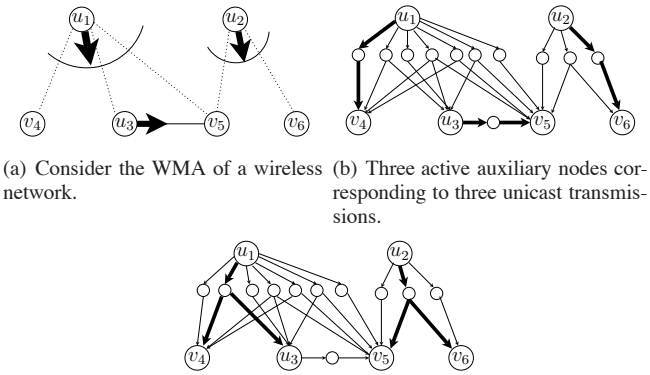
B. A Generalized Node Exclusive Interference Model With the WMA

The node exclusive model is a commonly used interference model for bluetooth or for FH-CDMA networks [2], [13], [32] that admits efficient and provably good approximation of the optimal scheduling policy. In the traditional node exclusive model (without taking advantages of the WMA), *the data rate of each link is fixed at c_e and each node can only send to or receive from one other node at any time*. The objective function of optimal scheduling is thus equivalent to

$$\max_{\mathbf{r}} \sum_{e\in E} q_e[t] r_e = \max_{\mathcal{M}} \sum_{e\in E} q_e[t] c_e 1_{\{e\in\mathcal{M}\}} = \max_{\mathcal{M}} \sum_{e\in\mathcal{M}} q_e[t] c_e, \tag{9}$$

where $\mathcal{M}$ is a matching of the underlying graph G. Finding the optimal scheduling of (9) thus becomes a maximum weighted matching problem.

[1]In contrast with Definition 1 where the rate update rule is modified for a static system load, for a dynamic system load, the optimal rate update is kept unchanged. Only the scheduling update will be changed to incorporate imperfect scheduling as in (8).



(a) Consider the WMA of a wireless network.

(b) Three active auxiliary nodes corresponding to three unicast transmissions.

(c) Two active auxiliary nodes corresponding to two broadcast transmission.

Fig. 7. Illustration of the node exclusive model when the WMA is taken into consideration.

Nonetheless, when the WMA is taken into account, i.e. with the auxiliary nodes added for the broadcast nature of wireless transmission as discussed in Section II-B, the objective function of scheduling becomes

$$\max_{\mathbf{r}} \sum_{e\in E} q_e[t] r_e = \max_{\mathcal{A}} \sum_{e\in E} q_e[t] c_e 1_{\{e \text{ is adjacent to some node in } \mathcal{A}\}}, \tag{10}$$

where $\mathcal{A}$ is a set of *active auxiliary nodes*. Since each network node can only send to or receive from one auxiliary node (due to the node-exclusiveness assumption), we require that the node set $\mathcal{A}$ satisfies that any node in $\mathcal{A}$ does not share any common neighbor with any other node in $\mathcal{A}$.

Fig. 7(a) depicts a wireless network of six nodes. Nodes u_1, u_2, and u_3 would like to transmit and the transmission can be overheard by more than one receivers (see Fig. 7(a)). Two possible scheduling policies (two different $\mathcal{A}$s) are illustrated. In Fig. 7(b), the wireline counterpart of Fig. 7(a), three auxiliary nodes are active and correspond to three *unicast transmissions* ((u_1, v_4), (u_2, v_6), and (u_3, v_5)) highlighted by thick edges. The other scheduling policy contains two active auxiliary nodes as in Fig. 7(c) that correspond to two *broadcast* transmissions ($(u_1, \{v_4, u_3\})$ and $(u_2, \{v_5, v_6\})$).

It can be shown that maximizing (10) is equivalent to solving a maximum weighted *hypergraph matching* (MWHM) problem. In [8] a greedy maximal hypergraph matching (GMHM) is proposed as an approximation of the MWHM. More explicitly, the network first selects an auxiliary node v_a that maximizes $\sum_{e \text{ is adjacent to } v_a} q_e[t] c_e$ and includes v_a as part of the scheduling policy $\mathcal{A}$. Remove v_a and its neighbors and then restart this greedy selection of auxiliary node until a maximal scheduling policy $\mathcal{A}$ is reached. In this way, GMHM guarantees to find a $\frac{1}{\max_{v_a} |\mathsf{nbs}(v_a)|}$-approximation of the MWHM, where $\mathsf{nbs}(v_a)$ is the set of neighbors around v_a. We can further sharpen the approximation ratio as follows.

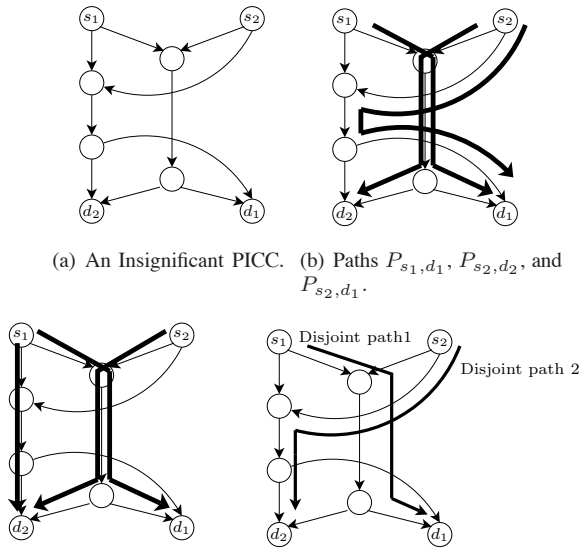
Proposition 5: For any given network, the GMHM is a $\frac{1}{5}$-approximation algorithm and can thus achieve at least $\frac{1}{5}$ of

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

the stability region Λ when used as an imperfect scheduling policy for the node-exclusive model.

Proof: Since each PICC consists of six paths, any auxiliary node participating in a PICC has at most six outgoing branches plus one incoming edge. As a result, during the wireless to wireline conversion, there is no need to include auxiliary nodes of $> (6+1)$ neighbors as the neighbors of those nodes will not fully participate in a PICC. The above reasoning shows that the GMHM can be made a $\frac{1}{7}$-approximation by eliminating the auxiliary nodes with > 7 neighbors. By further taking into account the edge-overlap conditions in Theorem 1, it can be shown that each auxiliary node needs to have at most four outgoing branches (two for the $\mathcal{P}$ paths and two for the $\mathcal{Q}$ paths). Therefore, the approximation ratio can be improved to $\frac{1}{5}$ by eliminating the auxiliary nodes with $> (4+1)$ neighbors. The proof is complete. ■

It is worth mentioning that the $\frac{1}{5}$-approximation is a lower bound of the performance of the GMHM. In our numerical study, GMHM has almost identical performance to the optimal MWHM solution. Other studies on the performance of greedy maximal matching for non-coding networks can be found in [16], [44].



(a) An Insignificant PICC. (b) Paths P_{s_1,d_1}, P_{s_2,d_2}, and P_{s_2,d_1}.

(c) Paths Q_{s_2,d_2}, Q_{s_1,d_1}, and Q_{s_1,d_2}.

(d) The embedded edge-disjoint paths.

Fig. 8. Illustration of an insignificant PICC that contains a pair of edge-disjoint paths as a strict subgraph.

V. Distributed Code Design for PINC

The rate control and link scheduling algorithms described thus far allocate optimal rates at each link so that the utility function can be maximized. The next question is what is the network coding scheme that can achieve the optimal rate assignment? For a given PICC, we proposed a coding scheme based on identifying distributedly special edges in the PICC in [20]. Specifically, carefully chosen *decoding* operations are performed on special decoding edges, while random network coding is performed on all other edges for the sake of scalability and distributiveness. In this paper we present a new, more efficient approach in which each edge e decides the coding operation based on the subset of the six paths of the given PICC that use e. Since each edge naturally knows whether itself participates in a given route/path or not (a byproduct of the initial path-search phase of Algorithm A), the corresponding coding operation can be decided locally without knowing the entire topology of the network. Moreover, this scheme only uses binary XOR operations, which has computational advantages over schemes based on a large finite field $\mathrm{GF}(2^8)$ or $\mathrm{GF}(2^{16})$. The new binary scheme achieves the same optimal throughput as that of [19] while the later requires the use of a larger field.

Practical network coding [7] uses the concept of "generations" that synchronize the network operations as coding is performed only within the same generation. With appropriate route/path selection and a carefully-designed generation-flushing policy, packets seldom cycle in the network. For the following, we thus restrict our attention to acyclic networks. In [19], it is observed that some PICCs have negligible impact on scheduling/rate-control and can be absorbed by a pair of edge-disjoint paths or by other PICCs. Fig. 8(a) represents one such insignificant PICC. The $\mathcal{P}$ and $\mathcal{Q}$ paths in Figs. 8(b) and 8(c) verify that Fig. 8(a) is indeed a PICC. However, within Fig. 8(a), there exists a pair of edge-disjoint paths P_{s_1,d_1} and P_{s_2,d_2} as illustrated in Fig. 8(d). One can send symbols X and Y along the edge-disjoint paths without using up all available bandwidth in the given PICC. From the throughput/cost perspective, the pair of edge-disjoint paths (Fig. 8(d)) *dominates* the given PICC (Fig. 8(a)), the latter of which is thus insignificant in the rate-control/scheduling analysis and can be removed from consideration without affecting the optimality of the solution. Let (s_1, d_1) and (s_2, d_2) denote the pair of unicast sessions of interest. For acyclic networks, the following four rules[2] identify the insignificant PICCs (each consisting of three $\mathcal{P}$ paths and three $\mathcal{Q}$ paths).

- **Rule 1:** If P_{s_2,d_1} and Q_{s_1,d_2} meet at any edge, the PICC is insignificant.

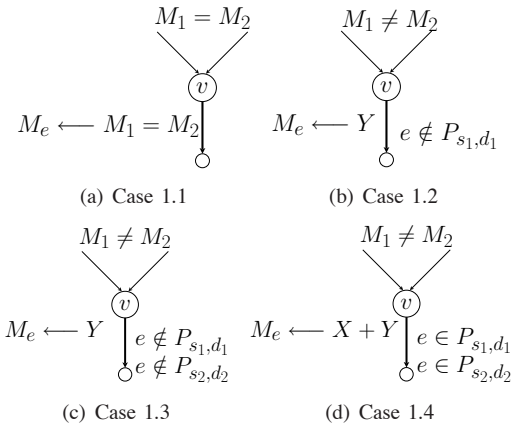
Depending on whether P_{s_1,d_1} and Q_{s_1,d_1} are the same path (and symmetrically whether P_{s_2,d_2} and Q_{s_2,d_2} are the same path), we have the following three more rules.

- **Rule 2.1:** Suppose $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. If there exists an edge e shared by all three paths P_{s_1,d_1}, P_{s_2,d_2}, and Q_{s_2,d_2}, then the PICC is insignificant.
- **Rule 2.2:** Suppose $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. If there exists an edge e shared by all three paths Q_{s_2,d_2}, Q_{s_1,d_1}, and P_{s_1,d_1}, then the PICC is insignificant.
- **Rule 2.3:** Suppose $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. Declare the PICC as insignificant.

Rules 1 to 2.3 can be implemented distributedly in the initialization phase by sending tokens along the paths to explore whether the paths share a given edge. For example, the insignificant PICC in Fig. 8(a) can be identified by Rule 1 and removed from consideration. Our new XOR-based scheme is then performed on the remaining PICCs that are not removed by the above four rules. The detailed description of the code construction is as follows.

[2]A detailed proof of the correctness of these four rules and the corresponding distributed implementation can be found in [21].

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.



(a) Case 1.1 (b) Case 1.2

(c) Case 1.3 (d) Case 1.4

Fig. 9. Cases 1.1 to 1.4 of the new distributed code construction using only the binary XOR operation. The thick arrow is the edge e in which the coding decision will be made. M_1 and M_2 are the coded symbols along the two incoming edges of v. M_e is the outgoing coded symbol along edge e.

The source: Source s_1 sends its own symbol X along P_{s_1,d_1}, Q_{s_1,d_1}, and Q_{s_1,d_2}, and s_2 sends its own symbol Y along P_{s_2,d_2}, Q_{s_2,d_2}, and P_{s_2,d_1}.

Each intermediate edge: At each edge e that is the outgoing edge of an intermediate network node $v \neq s_1, s_2$. Consider the following cases.

- **Case 1:** $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} \neq Q_{s_2,d_2}$. Consider four sub-cases. **Case 1.1:** If all incoming edges of v carry the same symbol, then forward that symbol. **Case 1.2:** If Case 1.1 is not satisfied and $e \notin P_{s_1,d_1}$, then send Y through e. **Case 1.3:** If Case 1.1 is not satisfied, $e \in P_{s_1,d_1}$, and $e \notin P_{s_2,d_2}$, then send X through e. **Case 1.4:** If Case 1.1 is not satisfied, $e \in P_{s_1,d_1}$, and $e \in P_{s_2,d_2}$, then send the binary XORed symbol $X + Y$ through e.
- **Case 2:** $P_{s_1,d_1} \neq Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. This is a symmetric case of Case 1. We perform the symmetric operations of Case 1 by swapping the roles of the first unicast session (s_1, d_1), X, and $\mathcal{P}$ with the roles of the second session (s_2, d_2), $\mathcal{Q}$ and Y.
- **Case 3:** $P_{s_1,d_1} = Q_{s_1,d_1}$ and $P_{s_2,d_2} = Q_{s_2,d_2}$. Perform the same operations as in Cases 1.1 to 1.4.

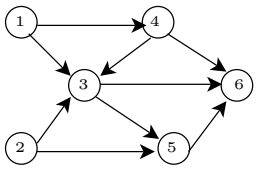
Fig. 9 illustrates Cases 1.1 to 1.4 for an outgoing edge e of an intermediate node v.

Proposition 6: For an acyclic network, consider a PICC that is not removed by Rules 1 to 2.3. Then destination d_1 (resp. d_2) is able to recover the designated symbol X (resp. Y) using the above locally computed binary coding scheme.

The proof is relegated to Appendix C.

VI. Numerical Experiments

We perform simulations under the linear signal-to-noise-&-interference-ratio (SINR) model. The objective is to compare the non-coded and the network coding solutions with both perfect and imperfect scheduling. To implement imperfect scheduling, we maintain a small pool imperfect scheduling policy $\tilde{\mathbf{\Theta}} \subsetneq \mathbf{\Theta}$ and choose the imperfect scheduling from the smaller policy pool $\tilde{\mathbf{\Theta}}$ in a similar way as in [28] according to the following. Every scheduling policy $\theta \in \tilde{\mathbf{\Theta}}$ is associated with a rate vector r_e^θ. Further assume that every $\theta \in \tilde{\mathbf{\Theta}}$ is associated with a set of queue lengths $\{q_e^\theta : \forall e\}$ such



node	location	Class	Source	Destination
1	(0,0)	1	1	6
2	(0,2)	2	2	6
3	(1,1)	3	4	5
4	(2.2,0)			
5	(2.5,2)			
6	(3.5,1)			

Fig. 10. The network topology, node locations, and three classes of unicast traffic used in the simulations.

that the policy θ is a γ_θ-approximation policy satisfying $\sum_{e\in E} q_e^\theta r_e^\theta \geq \gamma_\theta \max_{\mathbf{r}} \sum_{e\in E} q_e^\theta r_e$. If the following condition holds in the t-th time slot

$$\max_{\theta\in\tilde{\mathbf{\Theta}}} \sum_{e\in E} q_e[t] r_e^\theta \geq \gamma \min_{\theta\in\tilde{\mathbf{\Theta}}} \left(\sum_{e\in E} [q_e[t] - q_e^\theta]^+ r_e^{\max} + \frac{\sum_{e\in E} q_e^\theta r_e^\theta}{\gamma_\theta} \right),$$

for some γ, where $r_e^{\max}$ is the maximum possible rate along edge e, then policy θ_a^* that maximizes the left-hand side is a γ-approximation of the optimal scheduling policy with weights $q_e[t]$ on each edge. We can use such a scheduling policy θ_a^* in the reduced policy pool $\tilde{\mathbf{\Theta}}$ without the computationally expensive step of computing the optimal scheduling policy θ^* in the right-hand side of (8). If no such θ_a^* exists, we compute directly one $\theta[t]$ satisfying (8) and store this new $\theta[t]$ and the associated $q_e[t]$ in the small pool $\tilde{\mathbf{\Theta}}$.

We assume that the total power assigned to node u at any time slot is bounded by $P_{u,\max}$. To achieve the optimal throughput, in each time slot, each node u should either transmit at full power $P_{u,\max}$ or remain silent. For any unicast transmission from u to v, the data rate $r_{u,v}$ is assumed to be proportional to the SINR level[3] at the receiver v, which is formally expressed as.

$$r_{uv} = W \frac{G(u,v) 1_{\{(u,v) \text{ is activated}\}} P_{u,\max}}{N_0 + \sum_{w:w\neq u} G(w,v) 1_{\{\text{node } w \text{ is sending}\}} P_{w,\max}},$$

where N_0 is the background noise, W is the bandwidth of the system, and $G(u,v)$ is the path gain between nodes u and v which is set to $(\text{dist}(u,v))^{-4}$, where $\text{dist}(u,v)$ is the Euclidean distance between nodes u and v. With network coding, the data rate of the broadcast link with multiple receivers is proportional to the minimum of the SINR levels at those receivers. More precisely, if node u is broadcasting to nodes $v_1, \cdots, v_n$, the data rate of this broadcast link, $r_{u,\{v_1,\ldots,v_n\}}$, becomes

$$r_{u,\{v_1,\ldots,v_n\}} = W \min_{\{i=1,\ldots,n\}} \left\{ \frac{G(u,v_i) 1_{\{(u,\{v_1,\ldots,v_n\}) \text{ is activated}\}} P_{u,\max}}{N_0 + \sum_{w:w\neq u} G(w,v_i) 1_{\{w \text{ is sending}\}} P_{w,\max}} \right\}.$$

We run the simulations on the topology in Fig. 10. The X- and Y-coordinates of the six network nodes are specified in the figure. We simulate three classes of users and each class is allowed to use multi-path or multi-PICC communications.

[3]The linear SINR model can be viewed as a first order approximation of the information-theoretic $W \log(1 + SINR)$ model.

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

The source and destination pair of each class is also shown in the figure. A logarithmic utility function $U(\cdot) = \log(\cdot)$ is assumed for all classes. In our simulation, we use $W = 10$, $N_0 = 1$, $P_{u,\max} = 1$, $\forall u$, the proximal coefficient $\alpha_i = 0.01$, the step sizes $\beta_e = 0.01$, $\forall e$, $\beta_i = 0.01$, $\forall i$, and 10 inner iterations within each proximal iteration $K = 10$. Fig. 11 represents the results for the case of deterministic arrival and departure. Using the non-coded solution with perfect scheduling the rates of classes 1 and 2 converge to about 0.377 and the rate of class 3 converges to 0.325. When imperfect scheduling is used without network coding and $\gamma = 0.6$ the rates of classes 1 and 2 remain the same as the perfect scheduling case and the rate of class 3 is reduced by only 0.02. The number of time slots in which new schedules need to be computed is 18 out of totally 5000 time slots (500 proximal iterations). When we further reduce γ to $\frac{1}{3}$, the rate of class 1 is 4.0, the rate of class 2 is 3.5, and the rate of class 3 is 3.2 which shows a deviation from the optimal rates. The number of time slots in which new schedules need to be computed is 7 out of totally 5000 time slots.

The PINC solution with both the perfect and imperfect scheduling with $\gamma = 0.6$, $\frac{1}{3}$ achieves strict fairness as the data rates of all classes converge to 4 as shown in Fig. 11. Using imperfect scheduling from a reduced pool of scheduling policies, the new schedules need to be computed in only 21 time slots when $\gamma = 0.6$ and in only 13 time slots when $\gamma = \frac{1}{3}$. With network coding, the computationally efficient imperfect scheduling method outperforms the non-coding solution with optimal scheduling from both the throughput and fairness perspectives. To show that the performance gain of PINC is universal for other channel models, we have also simulated the same topology with a $W \log(1 + SINR)$ model. Similar performance gain is observed in Fig. 12.

For the dynamic arrival and departure, we simulated the arrival of files for each class whose size is exponentially distributed with average size ($\frac{1}{\mu} = 100$). Each file arrives according to a Poisson process with rate λ. We vary the rate λ and report in Fig. 13 the average number of users in the system with respect to the system load per user $\rho \stackrel{\Delta}{=} \frac{\lambda}{\mu}$. As shown in the figure network coding with imperfect scheduling outperforms the non-coded solution with perfect scheduling by a significant 20%.

VII. Conclusion

For intersession network coding, coding across many sessions requires greater transmission power to broadcast the coded symbol to many receiver, which results in higher interference in the wireless multi-hop network. In both empirical and analytical studies, it has been shown that for an interference/energy aware network coding scheme, most of the coding opportunities involve only two sessions, referred herein as pairwise intersession network coding (PINC).

In this work, we have proposed a jointly optimal coding, scheduling, and rate-control scheme for wireless multi-hop networks based on the recent theoretical finding of PINC. The corresponding coding, scheduling, and rate-control components are decoupled by the use of queue lengths and the introduction of rate-balance dual variables. Our results have proven that in a wireless multi-hop network, the throughput advantage of PINC can be achieved without sacrificing the stability conditions. Moreover, PINC has minimal impact on the optimal rate-control/scheduling as the only new component necessary for scheduling PINC traffic is the *balance update* performed at the receivers. Following this new formulation, we have also studied the impact of γ-imperfect scheduling on PINC-based rate-control algorithm and for the corresponding distributed greedy hypergraph matching algorithm.

Numerical experiments have also been conducted for the linear and the logarithmic signal-to-noise/interference-ratio (SINR) models, which shows that the achievable rates using PINC and efficient imperfect scheduling outperforms that of non-coding transmission with computationally expensive optimal scheduling. We provide the following future directions to conclude this paper.

1) PINC admits a flow-based characterization similar to that for non-coding communications, which prompts further extension of the traditional wisdoms for non-coding communications to PINC. We will study the benefit/impact of PINC for different network objectives, including distributed scheduling policy for general interference models and the joint scheduling and energy minimization. In [41], the PINC has been extended from two unicast sessions to two multicast sessions. We will develop the corresponding coding/scheduling schemes for multicast traffic as well.
2) Another important ingredient for practical wireless network coding is the opportunistic way of exploiting overhearing and coding opportunities. Opportunistic listening/coding/routing takes advantages of the randomness of the channel, which can be viewed as another type of imperfect scheduling. Specifically, with opportunistic routing, the network users can control only part of the *routing/scheduling policy* and the random nature of the wireless channel will decide randomly which receivers will receive which packet transmission. This opportunistic behavior of wireless multi-hop network provides a unique challenge for the analysis of intersession network coding as the coded symbols intended for certain receivers may be received by a different set of receivers instead. We will build upon our understanding of imperfect scheduling and investigate the opportunistic coding opportunities with PINC.

Appendix A
Stability Results with Deterministic Arrivals

Sketches of the proof of Proposition 3: We first notice that by (6), q_{ij}^l is a constant and is thus bounded away from infinity. For the following, choose the following Lyapunov function $V(\mathbf{q}) = \sum_e \frac{(q_e)^2}{2\beta_e}$ for the dual variables $\mathbf{q} = \{q_e : \forall e\}$. Then

$$V(\mathbf{q}[t+1]) - V(\mathbf{q}[t]) = \sum_e q_e[t]\Big(\sum_i \sum_k H_i^k(e) x_i^k + \frac{1}{2} \sum_{(i,j):i\neq j} \sum_l H_{ij}^l x_{ij}^l - r_e\Big) + \text{const}, \tag{11}$$

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

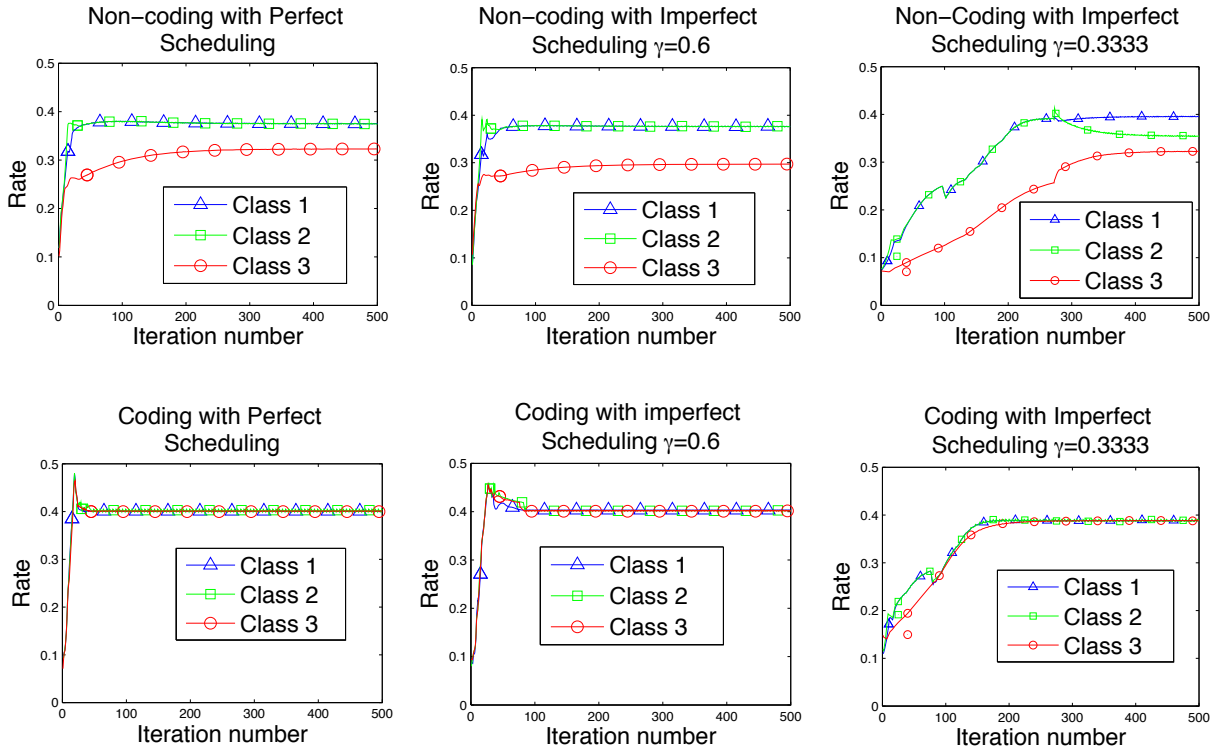


Fig. 11. The convergence results for the case of deterministic arrival with the linear SINR-based interference model.

where const is a constant bounded away from positive and negative infinity.

Since the fixed rate assignment $\mathbf{x}$ is in the interior of $\gamma\Lambda$, the first term of (11) can be rewritten as

$$-\epsilon \sum_e q_e[t] \left(\sum_i \sum_k H_i^k(e) x_i^k + \frac{1}{2} \sum_{(i,j):i\neq j} \sum_l H_{ij}^l x_{ij}^l \right) \tag{12}$$

for some $\epsilon > 0$. Combining (11) and (12), we have that any component of $\mathbf{q}$ tends to infinity will lead to negative difference of the Lyapunov function $V(q)$. As a result, all dual variables q_e and q_{ij}^l are bounded away from infinity. Since the primal variables have only bounded domains, the proof is complete. ■

APPENDIX B
STABILITY RESULTS WITH STOCHASTIC ARRIVALS

Sketches of the proof of Proposition 4: Let $\rho_i \triangleq \frac{\lambda_i}{\mu_i}$. Since $\{\rho_i\}_i$ is in the interior of $\gamma\Lambda$, by the definition of Λ, we can find ϵ, ρ_i^k and ρ_{ij}^l satisfying

$$\rho_i = \sum_k \rho_i^k + \sum_{j,l:j\neq i} \rho_{ij}^l, \forall i, \rho_{ij}^l = \rho_{ji}^l, \forall i,j,l,$$

$$\text{and } (1+2\epsilon) \sum_i \left(\sum_k H_i^k(e)\rho_i^k + \frac{1}{2} \sum_{j,l:j\neq i} H_{ij}^l(e)\rho_{ij}^l \right) \leq \gamma r_e^* \quad \text{for some } \mathbf{r}^* \in \mathrm{Co}(\mathcal{R}). \tag{13}$$

Let n_i denote the number of users in the system. The probability law of n_i is determined by a Markov process. Its transition rate is given by:

$$\begin{cases} n_i[t] \to n_i[t]+1 & \text{with rate } \lambda_i \\ n_i[t] \to n_i[t]-1 & \text{with rate } \mu_i \left(\sum_k x_i^k[t] + \sum_{j,l:j\neq i} \sum_l x_{ij}^l[t] \right) n_i[t] \end{cases}$$

A heuristic fluidity model argument is provided as follows. By the rate and scheduling update rules, we have

$$\begin{aligned} \mathbf{x}_i[t] = \arg\max_{\mathbf{x}} & U_i \left(\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l \right) \\ & - \sum_e q_e[t] \left(\sum_k H_i^k(e) x_i^k + \frac{1}{2} \sum_{j,l:j\neq i} H_{ij}^l(e) x_{ij}^l \right) \\ & - \left(\sum_{j:j>i} \sum_l q_{ij}^l[t] x_{ij}^l - \sum_{j:j<i} \sum_l q_{ji}^l[t] x_{ij}^l \right) \\ & - \frac{\alpha_i}{2} \left(\sum_k \left(x_i^k - y_i^k \right)^2 + \sum_{j,l:j\neq i} \left(x_{ij}^l - y_{ij}^l \right)^2 \right) \\ \sum_e q_e[t] r_e[t] \geq \gamma \max_{[r]} & \sum_e q_e[t] r_e \end{aligned} \tag{14}$$

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

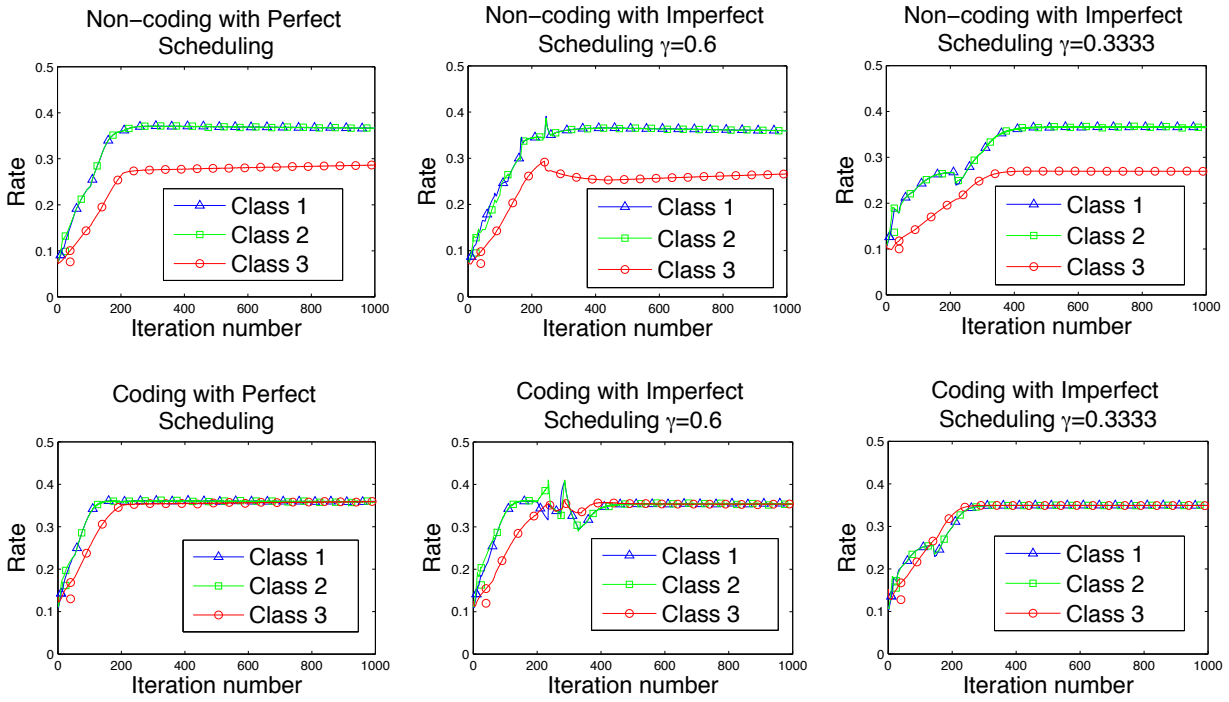


Fig. 12. The convergence results for the case of deterministic arrival with the $W\log(1+SINR)$-based interference model.

The first order derivatives of n_i, q_e, and q_{ij}^l become

$$\frac{d}{dt}n_i[t] = \lambda_i - \mu_i n_i[t]\left(\sum_k x_i^k[t] + \sum_{j,l:j\neq i} x_{ij}^l[t]\right)$$

$$\frac{d}{dt}q_e[t] = \begin{cases} \beta_e\left(\sum_i n_i[t]\left(\sum_k H_i^k(e)x_i^k[t] + \frac{1}{2}\sum_{j,l:j\neq i} H_{ij}^l(e)x_{ij}^l[t]\right) - r_e[t]\right) & \text{if positive} \\ & \text{or } q_e[t] > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{d}{dt}q_{ij}^l[t] = \beta_i\left(n_i[t]x_{ij}^l[t] - n_j[t]x_{ji}^l[t]\right)$$

Consider the following $V(\cdot,\cdot)$ function that will be used as the Lyapunov function of the system.

$$V(\mathbf{n},\mathbf{q}) = V_n(\mathbf{n}) + V_q(\mathbf{q})$$

where

$$V_n(\mathbf{n}) = \sum_i\left(\frac{1}{2}\frac{K\kappa_i n_i^2}{\lambda_i} + \frac{\alpha_i n_i}{\mu_i}\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right)$$

$$V_q(\mathbf{q}) = \sum_e \frac{(q_e)^2}{2\beta_e} + \sum_{i,j:i<j,l}\frac{(q_{ij}^l)^2}{2\beta_i}$$

By the fluidity model, we have the following expression for any positive constant K.

$$\frac{dV_n(\mathbf{n}[t])}{dt} = \sum_i\left(\frac{K\kappa_i n_i[t]}{\rho_i} + \alpha_i\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right)\left(\rho_i - n_i[t]\left(\sum_k x_i^k[t] + \sum_{j,l:j\neq i} x_{ij}^l\right)\right)$$

Therefore

$$\begin{aligned}\frac{dV_n}{dt} =& -\epsilon\sum_i\left(K\kappa_i n_i + \alpha_i\rho_i\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right) \\ &+ \sum_i\left(\frac{K\kappa_i n_i}{\rho_i} + \alpha_i\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right)\left((1+\epsilon)\rho_i - n_i\left(\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l\right)\right) \\ =& -\epsilon\sum_i\left(K\kappa_i n_i + \alpha_i\rho_i\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right) \\ &+ \sum_i\left(\frac{K(1+\epsilon)\kappa_i}{\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l} + \alpha_i\left(\sum_k y_i^k + \sum_{j,l:j\neq i} y_{ij}^l\right)\right)\left((1+\epsilon)\rho_i - n_i\left(\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l\right)\right) \\ &+ (A),\end{aligned}$$

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

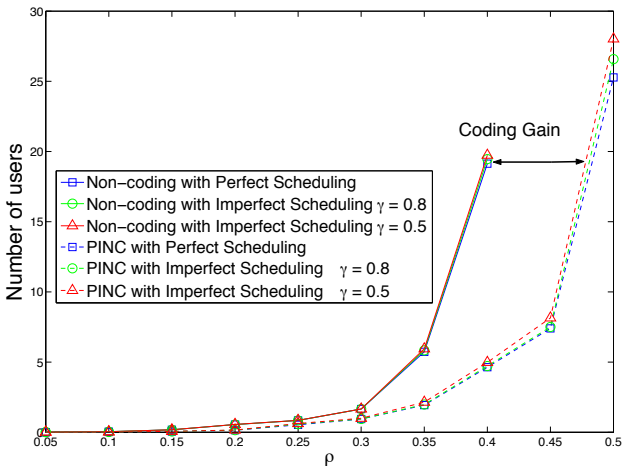


Fig. 13. The number of users in the system versus the system load for the case of dynamic arrival with the linear SINR-based interference model.

where

$$(A) \triangleq -\sum_i K\kappa_i \left(\frac{(1+\epsilon)}{\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l} - \frac{n_i}{\rho_i}\right) \left((1+\epsilon)\rho_i - n_i\left(\sum_k x_i^k + \sum_{j,l:j\neq i} x_{ij}^l\right)\right) \leq 0.$$

In the following we will use the following notation $y_{i(j)}^{k/l}$ to express that we are iterating over all y_i^k and y_{ij}^l. For example $\sum_{k,l,j,:j\neq i} y_{i(j)}^{k/l} \triangleq \sum_{k=1}^{|\mathcal{P}_i|} y_i^k + \sum_{j:j\neq i}\sum_{l=1}^{|\mathcal{PICC}_{ij}|} y_{ij}^l$. Similar notation is used for other variables such as ρ and q. We define F_0 and F_1 such that:

$$F_0 = (1+\epsilon)\left(\sum_i \alpha_i \sum_{k,j,l:j\neq i} y_{i(j)}^{k/l} \sum_{k',j',l':(k',j',l')\neq(k,j,l)} \rho_{i(j')}^{k'/l'}\right) - \epsilon\sum_i \alpha_i\rho_i \sum_{k,j,l} y_{i(j)}^{k/l} + (A)$$

$F_1 = (1+\epsilon)\sum_i \alpha_i M_{\Sigma x}\sum_{k,j,l:j\neq i}\rho_{i(j)}^{k/l}$. Here $M_{\Sigma x}$ is the maximum rate assigned to any user.

Since $\mathbf{x}_i[t]$ solves (14) and $U_i(\cdot) = \kappa_i\log(\cdot)$, there exist $\delta_i^k, \delta_{ij}^l \geq 0$ and $\delta_{i,M} \geq 0$ such that

$$\frac{\kappa_i}{\sum_{k',j',l':j'\neq i} x_{i(j')}^{k'/l'}} - \alpha_i(x_{i(j)}^{k/l} - y_{i(j)}^{k/l}) + \delta_{i(j)}^{k/l} - \delta_{i,M} = q_{e,i(j)}^{k/l}, \quad i,j,k,l : i\neq j$$

where

$$q_{e,i}^k = \sum_e q_e H_i^k(e)$$

$$q_{e,ij}^l = \begin{cases} \frac{1}{2}\sum_e q_e H_{ij}^l(e) + q_{ij}^l & \text{if } i<j \\ \frac{1}{2}\sum_e q_e H_{ij}^l(e) - q_{ji}^l & \text{if } i>j. \end{cases}$$

We have $\delta_{i(j)}^{k/l} x_{i(j)}^{k/l} = 0$ and $\delta_{i,M}\left(\sum_{k,j,l:j\neq i} x_{i(j)}^l - M_{\sum x}\right) = 0$ due to the complementary slackness conditions. In our online technical report [21] for each i we have explicitly construct a positive integer J_i such that

$$\delta_{i,M} \leq \frac{\kappa_i}{M_{\Sigma x}} + \alpha_i\frac{\sum_{k,j,l:i\neq j} y_{i(j)}^{k/l}}{J_i}, \forall i. \tag{15}$$

By choosing $F_2 = \max_i\{\frac{\kappa_i}{M_{\Sigma x}} + \alpha_i\frac{\sum_{k,j,l} y_{i(j)}^{k/l}}{J_i}\}$, F_2 is an upper bound for all $\delta_{i,M}$. Let $K = \frac{1}{1+\epsilon}$, we have

$$\begin{aligned}\frac{dV_n}{dt} \leq & -\epsilon\sum_i K\kappa_i n_i \\ & + \sum_e q_e\left((1+\epsilon)\sum_i\left(\sum_k H_i^k(e)\rho_i^k + \frac{1}{2}\sum_{j,l:j\neq i} H_{ij}^l(e)\rho_{ij}^l\right)\right. \\ & \left. - \left(\sum_i\sum_k n_i H_i^k(e)x_i^k + \frac{1}{2}\sum_{j,l:j\neq i} n_i H_{ij}^l(e)x_{ij}^l\right)\right) \\ & + \sum_{(i,j):i<j}\sum_l\left(q_{ij}^l(1+\epsilon)\left(\rho_{ij}^l - \rho_{ji}^l\right) - q_{ij}^l\left(n_i x_{ij}^l - n_j x_{ji}^l\right)\right) \\ & + F_0 + F_1 + F_2(1+\epsilon)\sum_i\sum_{k,j,l:j\neq i}\rho_{i(j)}^{k/l}. \end{aligned} \tag{16}$$

The details of how (16) is obtained in our online technical report [21]. Since

$$\begin{aligned}\frac{dV_q(\mathbf{q})}{dt} = & \sum_e q_e\left(\sum_i n_i\left(\sum_k H_i^k(e)x_i^k + \frac{1}{2}\sum_{j,l:j\neq i} H_{ij}^l(e)x_{ij}^l\right) - r_e\right) \\ & + \sum_{(i,j):i<j}\sum_l q_{ij}^l\left(n_i x_{ij}^l - n_j x_{ji}^l\right),\end{aligned}$$

and $\rho_{ij}^l = \rho_{ji}^l$, the overall drift $\frac{dV}{dt} = \frac{dV_n}{dt} + \frac{dV_q}{dt}$ becomes

$$\begin{aligned}\frac{dV}{dt} \leq & -\epsilon\sum_i K\kappa_i n_i - \epsilon\sum_e q_e\sum_i\left(\sum_k H_i^k(e)\rho_i^k + \frac{1}{2}\sum_{j,l:j\neq i} H_{ij}^l(e)\rho_{ij}^l\right). \\ & + F_0 + F_1 + F_2(1+\epsilon)\sum_i\sum_{k,j,l:j\neq i}\rho_{i(j)}^{k/l}.\end{aligned}$$

Here, we used (13). The Lyapunov function will have a negative drift and the system is stable. A full proof that takes into account the second-order variation can be obtained accordingly.

■

APPENDIX C
THE VALIDITY OF THE DISTRIBUTED CODE CONSTRUCTION

Proof of Proposition 6: We prove this theorem by induction. We perform coding operations sequentially from the most upstream edges to the most downstream edges. Let M_e represent the symbol transmitted along edge e. We have

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

the following induction hypothesis: If Case 1 is satisfied, then for any edge e we have:

$$M_e = \begin{cases} X \text{ or } X+Y & \text{if } e \in P_{s_1,d_1} \\ Y \text{ or } X+Y & \text{if } e \in P_{s_2,d_2} \\ X \text{ or } Y & \text{if } e \in P_{s_2,d_1} \text{ or } e \in Q_{s_2,d_2}. \end{cases} \quad (17)$$

Case 2 is a symmetric version of Case 1 and the discussion is thus omitted.

If Case 3 is satisfied, for any edge e we have:

$$M_e = \begin{cases} X \text{ or } X+Y & \text{if } e \in P_{s_1,d_1} \\ Y \text{ or } X+Y & \text{if } e \in P_{s_2,d_2} \\ X \text{ or } Y & \text{if } e \in P_{s_2,d_1} \text{ or } e \in Q_{s_1,d_2} \end{cases} \quad (18)$$

Both hypotheses (17), (18) are satisfied on the immediate outgoing edges of sources s_1 and s_2 which carry X and Y respectively. To show that the induction holds for all edges, we need to consider 13 scenarios when Case 1 is satisfied as in Table I and 5 scenarios when Case 3 is satisfied as in Table II. The entries in the second column in both Table I and II represent the paths that share edge e and the entries in the third column represent the corresponding coding operation that will be performed on edge e.

Since Case 3 is simpler than Case 1, we disccuss Case 3 first and then move on to Case 1.

For Case 3 we have $P_{s_1,d_1} = Q_{s_1,d_1}$, $P_{s_2,d_2} = Q_{s_2,d_2}$. Therefore, we have four distinct paths $P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}$, and Q_{s_1,d_2} in the PICC. If an edge is shared by three paths, it is either the case that the PICC is insignificant by Rule 1 or condition 2 of Theorem 1 is not satisfied. There are $\binom{4}{2} = 6$ scenarios in which two paths meet at a single edge. Since by Rule 1, P_{s_2,d_1} and Q_{s_1,d_2} do not meet, we are left with 5 scenarios to consider as in Table II

In the following, we prove that the induction hypothesis follows for the 5 scenarios in Case 3.

Scenario 1 as in Table II The paths that meet at edge e are P_{s_1,d_1} and P_{s_2,d_2}. The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be $X+Y$ according to the hypothesis, which is the intersection of the first two cases of (18). The coded symbol $X+Y$ will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can decode both X and Y and compute the coded symbol $X+Y$ according to Case 3.4 (or equivalently Case 1.4) in Section V and send $X+Y$ along e. The hypothesis holds in this scenario that $e \in P_{s_1,d_1} \cap P_{s_2,d_2}$.

Scenario 2 and 3 as in Table II The paths that meet at edge e are P_{s_1,d_1} and Q_{s_1,d_2} (P_{s_1,d_1} and P_{s_2,d_1}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be X according to the hypothesis, which is the intersection of the first and third cases of (18). The symbol X will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute X according to Case 3.3 (or equivalently Case 1.3) in Section V and send X along e. The hypothesis holds in this scenario that $e \in P_{s_1,d_1} \cap Q_{s_1,d_2}$ ($e \in P_{s_1,d_1} \cap P_{s_2,d_1}$).

Scenario 4 and 5 as in Table II The paths that meet at edge e are P_{s_2,d_2} and Q_{s_1,d_2} (P_{s_2,d_2} and P_{s_2,d_1}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be Y according to the hypothesis, which is the intersection of the last two cases of (18). The symbol Y will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute the symbol Y according to Case 3.2 (or equivalently Case 1.2) in Section V and send Y along e. The hypothesis holds in this scenario that $e \in P_{s_2,d_2} \cap Q_{s_1,d_2}$ ($e \in P_{s_2,d_2} \cap P_{s_2,d_1}$).

For Case 1 we have $P_{s_1,d_1} = Q_{s_1,d_1}$. Therefore, we have five distinct paths $P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}, Q_{s_2,d_2}$, and Q_{s_1,d_2} in the PICC. By Rule 1 P_{s_2,d_1} and Q_{s_1,d_2} will not meet at a single edge. Also $P_{s_1,d_1}, P_{s_2,d_2}, Q_{s_2,d_2}$ will not meet at a single edge due to Rule 2.1. Therefore, any scenario in which an edge is used by five or four of the paths is impossible. We have $\binom{5}{3} = 10$ different scenarios in which edge e is used by three distinct paths. The scenarios in which an edge is used by $P_{s_1,d_1}, P_{s_2,d_2}, P_{s_2,d_1}$ or $P_{s_1,d_1}, Q_{s_2,d_2}, Q_{s_1,d_2}$ violate condition 2 of Theorem 1. The scenario in which an edge is used by $P_{s_1,d_1}, P_{s_2,d_2}, Q_{s_2,d_2}$, will be removed because it satisfies Rule 2.1. The scenarios in which an edge is used by $(P_{s_1,d_1}, P_{s_2,d_1}, Q_{s_1,d_2})$, $(P_{s_2,d_2}, Q_{s_2,d_1}, Q_{s_1,d_2})$, or $(P_{s_2,d_1}, Q_{s_2,d_2}, Q_{s_1,d_2})$ will be removed because it satisfies Rule 1. As a result, we need to only consider 4 scenarios in which e is used by three distinct paths (Scenarios 4, 5, 11, 12 in Table I). Since by Rule 1 P_{s_2,d_1} and Q_{s_1,d_2} do not use the same edge we have only 9 scenarios in which e is used by two paths. The total is 13 scenarios as in Table I.

In the following, we prove that the induction hypothesis holds for the 13 scenarios of Case 1.

Scenarios 1, 3, 4, 7 and 12 as in Table I The paths that meet at edge e are (P_{s_2,d_1} and P_{s_2,d_2}), (P_{s_2,d_1} and Q_{s_2,d_2}), (P_{s_2,d_1}, P_{s_2,d_2}, and Q_{s_2,d_2}), (P_{s_2,d_2} and Q_{s_2,d_2}), or (Q_{s_1,d_2}, P_{s_2,d_2}, and Q_{s_2,d_2}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be Y according to the hypothesis, which is the intersection of the last two cases of (17). The symbol Y will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute the symbol Y according to Case 1.2 in Section V and send Y along e. The hypothesis holds in these scenarios.

Scenarios 2, 5 and 13 as in Table I The paths that meet at edge e are (P_{s_2,d_1} and P_{s_1,d_1}), (P_{s_2,d_1}, P_{s_1,d_1}, and Q_{s_2,d_2}), or (P_{s_1,d_1} and Q_{s_2,d_2}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be X according to the hypothesis, which is the intersection of the first and third cases of (17). The symbol X will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute X according to Case 1.3 in Section V and send X along e. The hypothesis holds in these scenarios.

Scenarios 6 and 11 as in Table I The paths that meet at edge e are (P_{s_1,d_1} and P_{s_2,d_2}) or (Q_{s_1,d_2}, P_{s_1,d_1}, and P_{s_2,d_2}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

TABLE I
THE LIST OF POSSIBLE CODING OPERATIONS A NODE HAS TO PERFORM IF THE PICC SATISFIES CASE 1

PICC satisfies Case 1					
	Paths sharing edge e	Symbols transmitted on edge e		Paths sharing edge e	Symbols transmitted on edge e
Scenario 1	$P_{s_2,d_1}\ P_{s_2,d_2}$	Y	Scenario 8	$Q_{s_1,d_2}\ P_{s_1,d_1}$	X or $X+Y$
Scenario 2	$P_{s_2,d_1}\ P_{s_1,d_1}$	X	Scenario 9	$Q_{s_1,d_2}\ P_{s_2,d_2}$	Y or $X+Y$
Scenario 3	$P_{s_2,d_1}\ Q_{s_2,d_2}$	Y	Scenario 10	$Q_{s_1,d_2}\ Q_{s_2,d_2}$	Y or $X+Y$
Scenario 4	$P_{s_2,d_1}\ P_{s_2,d_2}\ Q_{s_2,d_2}$	Y	Scenario 11	$Q_{s_1,d_2}\ P_{s_1,d_1}\ P_{s_2,d_2}$	$X+Y$
Scenario 5	$P_{s_2,d_1}\ P_{s_1,d_1}\ Q_{s_2,d_2}$	X	Scenario 12	$Q_{s_1,d_2}\ P_{s_2,d_2}\ Q_{s_2,d_2}$	Y
Scenario 6	$P_{s_1,d_1}\ P_{s_2,d_2}$	$X+Y$	Scenario 13	$P_{s_1,d_1}\ Q_{s_2,d_2}$	X
Scenario 7	$P_{s_2,d_2}\ Q_{s_2,d_2}$	Y			

TABLE II
THE LIST OF POSSIBLE CODING OPERATIONS A NODE HAS TO PERFORM IF THE PICC SATISFIES CASE 3

PICC satisfies Case 3		
	Paths sharing edge e	Symbols transmitted on edge e
Scenario 1	$P_{s_1,d_1}\ P_{s_2,d_2}$	$X+Y$
Scenario 2	$P_{s_1,d_1}\ Q_{s_1,d_2}$	X
Scenario 3	$P_{s_1,d_1}\ P_{s_2,d_1}$	X
Scenario 4	$P_{s_2,d_2}\ Q_{s_1,d_2}$	Y
Scenario 5	$P_{s_2,d_2}\ P_{s_2,d_1}$	Y

must be $X+Y$ according to the hypothesis, which is the intersection of the first two cases of (17). The coded symbol $X+Y$ will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can decode both X and Y and compute the coded symbol $X+Y$ according to Case 1.4 in Section V and send $X+Y$ along e. The hypothesis holds in these scenarios.

Scenarios 9 and 10 as in Table I The paths that meet at edge e are (Q_{s_1,d_2} and P_{s_2,d_2}), or (Q_{s_1,d_2} and Q_{s_2,d_2}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be either Y or $X+Y$ according to the hypothesis. The symbol Y or $X+Y$ will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute the symbol Y according to Case 1.2 in Section V and send Y along e. The hypothesis holds in these scenarios.

Scenario 8 as in Table I The paths that meet at edge e are (Q_{s_1,d_2} and P_{s_1,d_1}). The symbols carried by the paths on the respective previous edges can be the same or not. If they are the same, the symbols must be either X or $X+Y$ according to the hypothesis. The symbol X or $X+Y$ will be forwarded and the invariant holds for the target edge e. If the symbols that enter edge v, the tail of e are different, then node v can compute X according to Case 1.3 in Section V and send X along e. The hypothesis holds in this scenario. ■

REFERENCES

[1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. Yeung, "Network information flow," *IEEE Trans. Inform. Theory*, vol. 46, no. 4, pp. 1204–1216, July 2000.

[2] D. Baker, J. Wieselthier, and A. Ephremides, "Distributed algorithm for scheduling the activation of links in a self-organizing mobile radio network," in *Proc. IEEE Int'l Conf. Commun.*, 1982.

[3] D. Bertsekas, A. Nedić, and A. Ozdaglar, *Convex Analysis and optimization*. Athena Scientific, 2003.

[4] D. Bertsekas and J. Tsitsiklis, *Parallel and distributed computation: Numerical methods*. Athena Scientific, 1997.

[5] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*. Kyoto, Japan, August 2007.

[6] P. Chaporkar and A. Proutiere, "Adaptive network coding and scheduling for maximizing throughput in wireless networks," in *Proc. Annual ACM Int'l. Conf. on Mobile Computing & Networking (MobiCom)*. Montréal, Canada, September 2007, pp. 315–146.

[7] P. Chou, Y. Wu, and K. Jain, "Practical network coding," in *Proc. 41st Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, October 2003.

[8] T. Cui, L. Chen, and T. Ho, "Distributed optimization in wireless networks using broadcast advantage," in *Proc. 46th IEEE Conf. Decision and Contr.* New Orleans, December 2007.

[9] ——, "Energy efficient opportunistic network coding for wireless networks," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

[10] R. Dougherty, C. Freiling, and K. Zeger, "Insufficiency of linear coding in network information flow," *IEEE Trans. Inform. Theory*, vol. 51, no. 8, pp. 2745–2759, August 2005.

[11] A. Eryilmaz and D. Lun, "Control for inter-session network coding," in *Proc. 3rd Workshop on Network Coding, Theory, & Applications (NetCod)*, January 2007.

[12] P. Gupta and T. Javidi, "Towards Throughput and Delay-Optimal Routing for Wireless Ad-Hoc Networks," *Asilomar Conference on Signals, Systems and Computers*, November 2007.

[13] B. Hajek and G. Sasak, "Link scheduling in polynomial time," *IEEE Trans. Inform. Theory*, vol. 34, no. 5, September 1988.

[14] T. Ho, Y. Chang, and K. Han, "On constructive network coding for multiple unicasts," in *Proc. 44th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, October 2006.

[15] T. Ho, M. Médard, R. Koetter, D. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413–4430, October 2006.

[16] C. Joo, X. Lin, and N. Shroff, "Understanding the capacity region of the greedy maximal scheduling algorithm in multi-hop wireless networks," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

[17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "XORs in the air: Practical wireless network," in *Proc. ACM Special Interest Group on Data Commun. (SIGCOMM)*, 2006.

[18] A. Keshavarz-Haddad and R. Riedi, "Bounds on the benefit of network coding: Throughput and energy saving in wireless networks," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

[19] A. Khreishah, C.-C. Wang, and N. Shroff, "Optimal rate control with inter-session network coding," *IEEE/ACM Trans. Networking*, submitted.

[20] ——, "Optimization-based rate control for communication networks with inter-session network coding," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

[21] ——, "Cross-layer optimization for wireless multihop networks with pairwise intersession network coding," Purdue University, online Technical Report, 2008, http://web.ics.purdue.edu/∼akhreish/JSAC08TR.pdf.

[22] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, vol. 11, no. 5, pp. 782–795, October 2003.

[23] J. Le, J. Lui, and D. Chiu, "How many packets can we encode? — an analysis of practical wireless network coding," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

[24] A. Lehman, "Network coding," Ph.D. dissertation, MIT, 2005.

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.

[25] S.-Y. Li, R. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, no. 2, pp. 371–381, February 2003.
[26] Z. Li and B. Li, "Network coding: the case of multiple unicast sessions," in *Proc. 42nd Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, Illinois, USA, September 2004.
[27] X. Lin and N. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proc. IEEE Conf. Decision & Contr.* Paradise Island, Bahamas, December 2004.
[28] ——, "The impact of imperfect scheduling on cross-layer congestion control in wireless networks," *IEEE Trans. Networking*, vol. 14, no. 2, pp. 302–315, April 2006.
[29] ——, "Utility maximization for communication networks with multipath routing," *IEEE Trans. Automat. Contr.*, vol. 51, no. 5, pp. 766–781, May 2006.
[30] J. Liu, D. Goeckel, and D. Towsley, "Bounds on the gain of network coding and bradcasting in wireless networks," in *Proc. 26th IEEE Conference on Computer Communications (INFOCOM)*. Anchorage, Alaska, USA, May 2007.
[31] D. Lun, N. Ratnakar, M. Médard, R. Koetter, D. Karger, T. Ho, E. Ahmed, and F. Zhao, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 2608–2623, June 2006.
[32] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider Guide to an Open Specification for Global Wireless Communications*. Prentice Hall, 2000.
[33] S. Omiwade, R. Zheng, and C. Hua, "Practical localized network coding in wireless mesh networks," in *Proc. IEEE Commun. Society Conf. Sensor & Ad Hoc Commun. & Networks (SECON)*. San Francisco, USA, June 2008.
[34] J. Price and T. Javidi, "Network coding for resource redistribution in a unicast network," in *Proc. 45th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, September 2007.
[35] Y. Sagduyu and A. Ephremides, "On joint MAC and network coding in wireless ad hoc networks," *IEEE Trans. Inform. Theory*, vol. 53, no. 10, October 2007.
[36] S. Sengupta, S. Rayanchu, and S. Banerjee, "An analysis of wireless network coding form unicast sessions: the case for coding aware routing," in *Proc. 26th IEEE Conference on Computer Communications (INFOCOM)*. Anchorage, USA, May 2007.
[37] V. Subramanian and D. Leith, "Draining-time Based Scheduling Algorithm," *IEEE CDC*, Dec 2007.
[38] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. IEEE Int'l Symp. Inform. Theory*. Seattle, USA, July 2006.
[39] C.-C. Wang and N. Shroff, "Pairwise intersession network coding on directed networks," *IEEE Trans. Inform. Theory*, submitted and under review.
[40] ——, "Beyond the butterfly — a graph-theoretic characterization of the feasibility of network coding with two simple unicast sessions," in *Proc. IEEE Int'l Symp. Inform. Theory*. Nice, France, June 2007.
[41] ——, "Intersession network coding for two simple multicast sessions," in *Proc. 45th Annual Allerton Conf. on Comm., Contr., and Computing*. Monticello, IL, September 2007.
[42] Y. Wu, P. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906–1918, November 2005.
[43] Y. Wu, P. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network planning in wireless ad hoc networks: a cross-layer approach," *IEEE J. Selected Areas of Commun.*, vol. 23, no. 1, pp. 136–150, January 2005.
[44] G. Zussman, A. Brzezinski, and E. Modiano, "Multihop Local Pooling for Distributed Throughput Maximization in Wireless Networks," in *Proc. 27th IEEE Conference on Computer Communications (INFOCOM)*. Phoenix, USA, April 2008.

Abdallah Khreishah received the B.S. degree from Jordan University of Science and Technology (JUST), Irbid, Jordan in 2004, and the M.S. degree from the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2006. He is currently working toward the Ph.D. degree at Purdue University. His research interests include network coding, congestion control, and cross layer design in wireless networks.

Chih-Chun Wang joined the School of Electrical and Computer Engineering in January 2006 as an Assistant Professor. He received the B.E. degree in E.E. from National Taiwan University, Taipei, Taiwan in 1999, the M.S. degree in E.E., the Ph.D. degree in E.E. from Princeton University in 2002 and 2005, respectively. He worked in Comtrend Corporation, Taipei, Taiwan, as a design engineer during in 2000 and spent the summer of 2004 with Flarion Technologies, New Jersey. In 2005, he held a post-doc position in the Electrical Engineering Department of Princeton University. He received the National Science Foundation Faculty Early Career Development (CAREER) Award in 2009. His current research interests are in the graph-theoretic and algorithmic analysis of iterative decoding and of network coding. Other research interests of his fall in the general areas of optimal control, information theory, detection theory, coding theory, iterative decoding algorithms, and network coding.

Ness B. Shroff received his Ph.D. degree from Columbia University, NY, in 1994 and joined Purdue university as an Assistant Professor. At Purdue, he became Professor of the School of Electrical and Computer Engineering in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University as the Ohio Eminent Scholar of Networking and Communications, a chaired Professor of ECE and CSE. His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, pricing, and security of these networks.

Dr. Shroff is a past editor for IEEE/ACM Trans. on Networking and the IEEE Communications Letters and current editor of the Computer Networks Journal. He has served as the technical program co-chair and general co-chair of several major conferences and workshops, such as the IEEE INFOCOM 2003, ACM Mobihoc 2008, IEEE CCW 1999, and WICON 2008. He was also a co-organizer of the NSF workshop on Fundamental Research in Networking, held in Arlie House Virginia, in 2003.

Dr. Shroff is a fellow of the IEEE. He received the IEEE INFOCOM 2008 best paper award, the IEEE INFOCOM 2006 best paper award, the IEEE IWQoS 2006 best student paper award, the 2005 best paper of the year award for the Journal of Communications and Networking, the 2003 best paper of the year award for Computer Networks, and the NSF CAREER award in 1996 (his INFOCOM 2005 paper was also selected as one of two runner-up papers for the best paper award).

Authorized licensed use limited to: Purdue University. Downloaded on June 12, 2009 at 20:02 from IEEE Xplore. Restrictions apply.