A Policy-Aware Enforcement Logic for Appropriately Invoking Network Coding

Ahmed Osama Fathy Atya, *Student Member, IEEE*, Ioannis Broustis, *Member, IEEE*, Shailendra Singh, *Member, IEEE*, Dimitris Syrivelis, Srikanth V. Krishnamurthy, *Fellow, IEEE*, and Thomas F. La Porta, *Fellow, IEEE*

***Abstract*—Network coding has been shown to offer significant throughput benefits over certain wireless network topologies. However, the application of network coding may not always improve the network performance. In this paper, we first provide an analytical study, which helps in assessing when network coding is preferable to a traditional store-and-forward approach. Interestingly, our study reveals that in many topological scenarios, network coding can in fact hurt the throughput performance; in such scenarios, applying the store-and-forward approach leads to higher network throughput. We validate our analytical findings via extensive testbed experiments. Guided by our findings as our primary contribution, we design and implement PACE, a Policy-Aware Coding Enforcement logic that enables network coding only when it is expected to offer performance benefits. Specifically, PACE leverages a minimal set of periodic link quality measurements in order to make per-flow online decisions with regards to when network coding should be activated, and when store-and-forward is preferable. It can be easily embedded into network-coding-aware routers as a user-level or kernel-level software utility. We evaluate the efficacy of PACE via: 1) ns-3 simulations, and 2) experiments on a wireless testbed. We observe that our scheme wisely activates network coding only when appropriate, thereby improving the total network throughput by as much as 350% in some scenarios.**

***Index Terms*—Measurements, network policy, rate adaptation, simulation, testbed, wireless network coding.**

I. INTRODUCTION

WIRELESS network coding (NC) exploits the broadcast nature of the wireless medium toward increasing the capacity of the network by encoding the information contained in multiple packets into a set of fewer packets at intermediate wireless routers [1]. With this, in conducive topologies, NC has been shown to offer significant throughput benefits, compared to a traditional store-and-forward router approach. On the other hand, studies suggest that when NC is blindly applied, it can cause severe degradation of the network throughput, especially in multirate environments [2]. In this paper, we show via analysis as well as measurements that while network coding is not a magical solution for all wireless network topologies, regulating the use of network coding and store-and-forward together can result in improved long-term throughput benefits.

Performance Improvement Due to NC: Now You See It, Now You Don't: Wireless network coding has been examined both theoretically and experimentally over the past decade, under many different deployment and traffic scenarios [1]–[8]. Two primarily identified factors affect the performance improvements due to wireless NC:

1) the network topology, which determines the ability of neighboring devices to successfully overhear each other's transmissions in order to decode encoded packets;
2) the traffic patterns of the different users, which dictate the number of packet encoding opportunities at the routers.

Let us consider the simple topology of Fig. 1, where Alice sends data to Bob, while Jim sends data to Emma, all routed via Jack. From among the above two factors, it is easy to see that #2 dictates the network coding gain: If Alice has a much higher application data rate than Jim, then the router (Jack) will rarely be able to encode Alice's and Jim's packets together. However, the application of NC here will never degrade the overall throughput[1] due to factor #2. On the other hand, factor #1 can be the reason for significant throughput degradation in the presence of NC. In particular, let us assume that both links *Alice-Jack* and *Jim-Jack* have a packet delivery ratio (PDR) equal to 1 at 54Mb/s, while the overhearing link *Alice-Emma* has a PDR equal to 0.2 at this rate. In this topological scenario, Jack will receive packets from Alice and Jim at similar bit rates. However, if Jack decides to constantly apply NC given the high availability of candidate packets, this will cause a tremendous degradation in the overall network throughput (compared to simply applying store-and-forward). This is because Emma will not be able to decode 80% of the delivered encoded packets by Jack, regardless of the coding gain, due to the poor link quality that she maintains with Alice. In other words, the throughput achieved with store-and-forward would be higher than that achieved with NC here. If Jack greedily prefers NC to store-and-forward whenever native packets from Alice and Jim are available in Jack's queues, his strategy will backfire and hurt the network performance. Clearly, if Alice adapts her transmission rate in order for the PDR on the link *Alice-Emma* to increase, then the long-term

Manuscript received November 01, 2013; revised November 15, 2014; accepted May 26, 2015; approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor A. Markopoulou. Date of publication June 19, 2015; date of current version August 16, 2016.

A. O. F. Atya, I. Broustis, S. Singh, and S. V. Krishnamurthy are with the University of California, Riverside, Riverside, CA 92521 USA (e-mail: a.osama@ieee.org).

D. Syrivelis is with the University of Thessaly, Volos 382 21, Greece.

T. F. La Porta is with the Pennsylvania State University, University Park, PA 16802 USA.

This work has supplementary downloadable material available at http://ieeexplore.ieee.org.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNET.2015.2438775

[1]We assume here for the sake of the discussion that the system overheads imposed due to NC do not affect the performance.

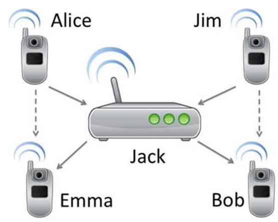


Fig. 1. Five-node topology that may potentially benefit from NC.

throughput due to NC may end up being higher than that with store-and-forward, or it may not. Note that recent studies have proposed novel bit rate adaptation algorithms that take into consideration the existence of network coding [3], [4]. However, they have not examined whether a store-and-forward strategy would still be more beneficial than NC in diverse channel environments, even when such NC-aware rate control protocols are applied, as we discuss in Section II.

Designing an Adaptive Decision-Making Engine: Given the above discussion, in this paper we design and implement PACE, a Policy-Aware Coding Enforcement logic for NC-capable wireless routers. PACE leverages a small set of periodic link quality measurements in order to decide whether NC or store-and-forward must be applied, for each particular data flow that traverses the router, in real time. The logic of PACE is guided by our analytical study, which considers different topological scenarios, and provides insights on which approach (NC or store-and-forward) is expected to offer higher long-term network throughput. More specifically, our contributions in this paper are the following.

1) *Analysis of the gains from network coding*: We perform an analytical assessment of the achieved network throughput with NC and store-and-forward, for a general topological class. We verify the accuracy of our analytical assessments via extensive testbed experiments, using a novel network coding software platform [9]. Our analysis, in conjunction with our testbed measurements, provides recommendations on when it is preferable to apply NC, and when store-and-forward is a wiser choice, in multirate settings. With this, we construct a concrete set of NC application guidelines for each considered topological scenario.
2) *Design of PACE*: We use our guidelines to design PACE, our Policy-Aware Coding Enforcement logic. PACE performs periodic link quality measurements to assess the potential effectiveness of NC. It leverages our analytical model to make online router decisions regarding when and how NC or store-and-forward should be enabled and what transmission bit rates are to be used in conjunction.
3) *Measurements and simulations*: We implement PACE on our wireless testbed, and we evaluate its efficacy via measurements over numerous different topologies and diverse link qualities. Moreover, we develop PACE in ns-3 and perform extensive simulations over large-scale single- and multihop network deployments. Our simulations and experiments demonstrate that PACE always follows the right strategy with regards to the application of NC; with this, it results in throughput improvements of up to 350% in some cases, i.e., by switching on NC when appropriate.

Our Work in Perspective: In this paper, we primarily focus on local NC topologies where an encoded packet by a router is decoded at the next hop. As we discuss in Section VI, this has direct applications in wireless LANs. We do not consider problems such as topology discovery, selection of which packets to code together, or aggregated ACK/NACK packets. Such issues have been addressed by other studies such as [1]. Our proposed scheme is directly applicable in previously proposed NC frameworks, such as [1], [7], [10], etc. To show the applicability of our approach in more generic settings, we rely on simulations in scenarios wherein packets traverse multiple wireless hops with potential opportunities to use network coding at each such hop (as in mesh networks).

The rest of the paper is organized as follows. In Section II, we discuss related work. In Section III, we present our generic analytical model and its validation. In Section IV, we develop the PACE algorithmic logic, which we evaluate in Section V. In Section VI, we discuss the scope of our study. Finally, our conclusions form Section VII.

II. Related Work

In this section, we discuss previous relevant NC studies. The idea of NC first appeared in [11], where Ahlswede *et al.* showed that by performing algebraic bit-wise operations in routers and by forwarding mixtures of packets, one can tremendously increase the transfer capacity of a multicast network. This work motivated the subsequent generation of a plurality of efforts to understand this concept and further exploit it in improving the network performance. As examples, Li *et al.* [12] show that linear codes are sufficient for achieving the optimal capacity in multicast traffic scenarios, although they are not sufficient for arbitrary network demands. Gkantsidis and Rodriguez[13] propose a platform for peer-to-peer content distribution, which depends on application-layer NC operations. Koetter and Medard[14] present polynomial-time algorithms for realizing network operations of encoding and decoding packets. Ho *et al.* [15] extend the algorithms of [14] and discuss distributed NC schemes. In addition, [16]–[18] show that for many specific scenarios, NC results in better throughput than pure forwarding. In [19], the authors study the capacity of 2-hop relay networks and propose a near-optimal coding scheme that can make a linear number of decisions in terms of which packets to combine using network coding. In the paper by Sharma *et al.*, the authors examine if network coding can benefit cooperative communications (CC) with multiple simultaneous flows. They consider analog network coding (as opposed to simple XOR operations as we do here) and quantify the noise at a node that aggregates packets. They demonstrate analytically that this noise can diminish the benefits of analog network coding. In contrast, our work examines the impact of link qualities and traffic load on the benefits achievable with XOR-based NC. Furthermore, we do not consider either cooperative communications or analog aggregation. Finally, unlike in [20], which focuses on theoretical analysis of NC with CC, we focus on experimentation and the design of a policy to only apply network coding in conducive conditions.

A. NC Applicability Assessment

The study that is mostly relevant to our work is by Chaporkar and Proutiere [2]. Similar to our work, they show that in multirate settings, systems with NC may have smaller throughputs than without coding. They argue that unless appropriate scheduling is applied, NC may lead to performance degradation in many scenarios. They further propose a generic framework that characterizes the throughput region with NC and enables the design of adaptive joint NC and scheduling schemes. However, they do not provide any generic guidelines or an online method for adaptively activating NC when it is expected to increase throughput. Moreover, [2] does not involve real network experimentation and/or measurements.

In [21], the authors analytically studied the capacity region of M unicast sessions in single-hop relay networks by modeling the broadcast packet erasure stationary channels. This work does not consider store-and-forward or involve any real network experimentation and/or measurements.

In [1], COPE is proposed for practical wireless coding. COPE does turn off network coding if packet loss rates are higher than 20%. However, this value is empirical.

Unlike in the above efforts, our contribution is in quantifying the degradation in the performance when the overhearing links PDR is low. Specifically, unlike in other efforts [1], we make a determination on at what point the quality is detrimental to network coding.

B. Analytical and Simulation Studies on Wireless NC

Liu and Xue in [22] analytically characterize the achievable rate regions with NC for a basic 3-node topology wherein no overhearing is involved. Vieira *et al.* [23] examine how the combination of NC and bit rate diversity affects the performance of broadcasting protocols. Scheuermann *et al.* [24] propose noCoCo, a deterministic scheduling scheme for NC to operate on two-way multihop traffic flows. Seferoglu *et al.* [25] propose code selection schemes that consider the properties of video traffic. Le *et al.* [6] provide an upper bound on the number of packets that can be coded together. Lun *et al.* [26] show that the problem of minimizing the communication cost can be formulated as a linear program and solved in a distributed manner. There has also been some work on NC-aware data rate control at the transport layer [5], [27]. However, *these studies do not address the problem of choosing between NC and store-and-forward toward improving the long-term network throughput.*

C. Experimental Work on Wireless Coding

Katti *et al.* [1] propose COPE, the first seminal implementation of wireless NC. Since one of the goals of COPE is to increase the number of encoding opportunities, low transmission rates are favored in order for native packets to be overheard by as many neighbors as possible. Their experiments with COPE show that even with very simple encoding operations, NC can provide significant capacity gains. However, they do not study cases where store-and-forward is preferable to NC. Rozner *et al.* in [10] present ER, a scheme that adopts the design of COPE and employs NC to perform efficient packet retransmissions. Rayanchu *et al.* [7] propose CLONE, a suite of algorithms for NC that take into account channel losses. Both [10] and [7] follow COPE's logic regarding the application of NC; they do not propose any policies for multirate settings.

Srinivasan *et al.* [28] propose a network metric that takes into consideration the interlink interference on packet reception probability. However, they do not consider transmissions at multiple bit rates or rate adaptation.

MORE [29] is a routing protocol that performs a random mixing of packets, right before they are forwarded. With this, routers that overhear a transmission can decide not to forward the same overhear packets. However, no decisions on NC versus store-and-forward are made. MIXIT [8] encodes symbols rather than packets. Relays use hints from the PHY layer in order to infer which symbols within a packet are correctly received with high probability. Note here that all of these studies are transmission-rate-unaware.

Kim *et al.* [3] study the performance of NC in multirate settings. They show that unless rate adaptation is NC-aware, NC may not offer significant performance benefits. They further design an NC-aware rate control algorithm for local topologies. Kumar *et al.* [4] take the same path but propose a different NC-aware rate control algorithm.

Hulya *et al.*, in [30], proposed I^2NC to overcome the non-negligible loss rates by combining intersession and intrasession network coding. In their work, intrasession specifies the amount of redundancy required to compensate for errors and intersession chooses the number of flows to code together. In [31], the authors proposed CLONE to improve the performance of NC by being aware of potential losses and introducing redundancy to cope with these losses. Neither [30] nor [31] considers potential throughput benefits with network coding in multirate settings.

The idea of using redundancy with network coding (as in [30]) is orthogonal to the question we are asking: *"When should network coding be applied in multirate settings?"*

All the above papers implicitly assume that NC should be applied whenever possible. However, as we discuss in Section III, this should not always be the case as it may lead to performance degradation.

NCRAWL [9] is a multirate network coding library. NCRAWL does not provide any insights by itself on when to (or when not to) apply network coding. We use NCRAWL as the underlying framework over which we implement PACE. In Section V, we provide more details on these implementation aspects.

D. NC on Wireline Networks

Finally, a large body of studies has investigated NC for wireline networks (e.g., [14], [15], and [18]). However, they do not account for the inherent properties of the wireless medium.

In short, very limited steps have been taken toward assessing the applicability of practical NC in relation with the inherent characteristics of the wireless medium. As we discuss, the application of wireless NC should be regulated.

III. Profiling the Applicability of NC

In this section, we discuss our analytical model for assessing whether/when the application of NC is preferable to store-and forward. Furthermore, we verify the accuracy of our analytical model via experiments on our wireless testbed.

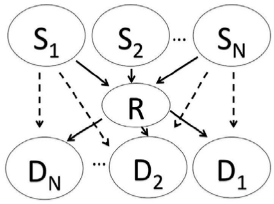


Fig. 2. N-node topology wherein network coding may offer performance benefits when the link qualities among the different users are conducive.

A. Designing Our Analytical Model

Our goal is to derive the throughput with both store-and-forward and NC in various single-hop topological scenarios. For this, we design an analytical model, which essentially computes the average number of packet transmissions in a multirate environment, for both the cases and from that, the throughput. We consider the general topological scenario in Fig. 2, which consists of N packet senders and N receivers (a total of N source–destination pairs) communicating via a relay node R.

1) Model Assumptions: Our considered setting involves links with heterogeneous qualities in terms of packet error rate (PER). At a specific time instance, each participating device (node) uses a specific transmission bit rate R_{node}, while every link between a sender and a receiver has a PER equal to $p_{\text{Sender}-\text{Receiver}}$; we assume that the network is quasi-stationary, wherein PER values remain unchanged for relatively long periods. This assumption is realistic in cases with slowly varying channels. This happens when there is not much mobility and the interference patterns are quasi-static (e.g., an office space). If the fading conditions are much more dynamic, it is hard assess channel qualities. This limitation is not unique to our problem; even rate adaptation schemes may underperform [32] in such scenarios. Given that we are not designing new channel quality estimation methods, we defer solving this problem to future work. Without loss of generality, in order to make the analysis tractable, we assume that data packet lengths have a fixed size equal to L bits. Note that our analysis focuses on the generic topology of Fig. 2, wherein encoded packets (i.e., mixtures or two or more native packets) constructed at a router, are decoded at the next hop. We do not consider cases where encoded packets traverse more than one hop. We elaborate on this assumption later in Section VI. Table I summarizes the notation used in our analysis.

The "packet" transmission rate between nodes i and j is $X_{ij} = \frac{R_i}{L}$ packets/s, while the transmission time, T_{ij}, of a packet is equal to $\frac{1}{X_{ij}}$. In what follows, we first analytically derive the average number of transmissions for each individual forwarding strategy separately. We use these derivations to estimate the average throughputs with the two cases. The analysis provides a quick and efficient way of understanding when to use NC and when to use store-and-forward.

2) Case for Store-and-Forward: The average time taken to deliver a packet from a source to a destination in the case of store-and-forward, sf, is given by

$$T_{\text{avg}}^{\text{sf}} = \frac{\sum_{k=1}^{N} \left(\rho_{S_kR}^{\text{sf}} T_{S_kR} + \rho_{RD_k}^{\text{sf}} T_{RD_k}\right)}{N} + Q_{\text{sf}}. \quad (1)$$

TABLE I
DEFINITIONS OF NOTATIONS

Symbol	Definition
R_i	Transmission rate for node i
L	Packet length
n	Number of flows to be coded together (equals 1 in case of store-and-forward).
$\rho_{S_kR}^{sf}$	Average number of transmissions from Sender S_k towards the Relay (case of store-and-forward).
$\rho_{RD_k}^{sf}$	Average number of transmissions from the Relay towards Destination D_k (case of store-and-forward).
$\rho_{S_kR}^{nc}$	Average number of transmissions from Sender S_k towards the Relay (case of NC).
$\rho_{RD_k}^{nc}$	Average number of transmissions from the Relay towards Destination D_k (case of NC).
ρ_R	Average number of transmissions for encoded packets from the Relay towards all intended recipients.
T_{S_kR}	Average transmission time between a sender node S_k and the relay node R.
T_{RD_k}	Average transmission time between the relay node R and a destination node D_k.
T_R	Average transmission time for the encoded packet from the relay R towards the selected MAC-level destination.
Q_{sf}	Average queuing time in the store-and-forward (sf) case.
Q_{nc}	Average queuing time in the NC (nc) case.
P_{nc}	Average processing time overhead in the NC (nc) case.
p_{AB}	Probability of error of the link between nodes A and B.
M	Maximum number of transmissions.

In the above expression, the numerator represents the total time taken to transfer "one" packet, on average, from each source to its destination. We take the average over all source–destination pairs. In addition, we include the average packet queuing time experienced by the packet, prior to its transmission attempts. The throughput of the store and forward scheme is then simply

$$\tau^{\text{sf}} = \frac{1}{T_{\text{avg}}^{\text{sf}}}. \quad (2)$$

The average number of transmissions for a packet from the source to its destination depends on the value of PER between the sender and the relay as well as on the PER between the relay and the destination. Thus, we need to consider the links *sender-relay* and *relay-destination* in our analysis. For instance in the X-topology in Fig. 1, we consider individually the direct links *Alice-Jack, Jim-Jack, Jack-Bob*, and *Jack-Emma*. Taking into account the PER on each of these individual links, the average number of transmissions in the store-and-forward case on the link between the sender and the relay can be computed as (details are in the supplementary Appendix)

$$\rho_{S_kR}^{\text{sf}} = \frac{\left(1 - (M+1)p_{S_kR}^{M} + M p_{S_kR}^{M+1}\right)}{1 - p_{S_kR}} \quad (3)$$

while the average number of transmissions between the relay and the destination can be computed as

$$\rho_{RD_k}^{\text{sf}} = (1 - p_{S_kR}) \frac{\left(1 - (M+1)p_{RD_k}^{M} + M p_{RD_k}^{M+1}\right)}{1 - p_{RD_k}} \quad (4)$$

where M is the maximum number of transmission attempts (including retransmissions) on any given link (we set $M = 7$ in this work, as suggested in [33]).

3) Case for Network Coding: As with store-and-forward, the average number of transmissions with NC depends on the PER on the *sender-relay* and *relay-destination* links. However, in addition, the likelihood of correct reception also depends on the value of PER on overhearing links (e.g., Alice → Emma). Given this, the average number of transmissions in the NC case for the generic topology between the N senders and the relay node in Fig. 2 is (see the Appendix for the full derivation)

$$\rho^{\mathrm{nc}}_{S_k R} = \frac{\left(1-(M+1)p^{M}_{S_k R} + M p^{M+1}_{S_k R}\right)}{1-p_{S_k R}} \tag{5}$$

and the average number of transmissions between the relay node and the destinations is given by

$$\rho^{\mathrm{nc}}_{RD_k} = \left(\prod_{i=1}^{N}(1-p_{S_i R})\right)\frac{\left(1-(M+1)p^{M}_{RD_k} + M p^{M+1}_{RD_k}\right)}{1-p_{RD_k}} \tag{6}$$

where D_k is the chosen 802.11-level destination for the encoded packet by the relay.[2] In our work, the relay selects D_k to be the recipient with the lowest PDR value. With this, all intended recipients obtain the encoded packet with high probability. Hence, for ease of notation $\rho^{\mathrm{nc}}_{RD_k}$ is simply referred to as ρ^{nc}_{R} here forth. The product $(\prod_{i=1}^{N}(1-p_{S_i R}))$ represents the success probability of packets being delivered from the transmitter(s) to the relay nodes. In the case of store and forward, we compute the success probability of each individual packet (in reaching the relay node from a sender k). In the network coding case, there are two or more incoming links to the relay; all packets need to be successfully delivered for encoding. We assume that the probabilities of success on each of the ingress links to the relay are independent (this is realistic if they experience independent interference patterns).

The average throughput in the case of NC depends not only on the average transmission times of native and encoded packets, but also on the overhearing probabilities at receivers. Based on the details in the Appendix, the average packet transmission time is given by

$$T^{\mathrm{nc}}_{\mathrm{avg}} = \frac{\left(\sum_{k=1}^{k=N}\rho^{\mathrm{nc}}_{S_k R} T_{S_k R}\right) + \rho^{\mathrm{nc}}_{R} T_R}{N} + Q_{\mathrm{nc}} + P_{\mathrm{nc}} \tag{7}$$

where T_R is the average transmission time for the encoded packet from the relay R toward the selected MAC-level destination. Thus, T_R is the reciprocal of transmission rate used by the relay for its target destination, and the decoding probability is given by

$$p^{\mathrm{dec}}_{i} = \prod_{k=1}^{n-1}\left(1-p^{M}_{S_k R}\right)\left(1-p_{S_k R}\right) \times\left(\frac{1-p^{M}_{S_k R}}{1-p_{S_k R}} + p_{S_k D_i}\frac{1-(p_{S_k R}\cdot p_{S_k D_i})^{M}}{1-p_{S_k R}\cdot p_{S_k D_i}}\right). \tag{8}$$

The throughput with NC is then computed to be

$$\tau^{\mathrm{nc}} = \frac{p^{\mathrm{dec}}_{i}}{T^{\mathrm{nc}}_{\mathrm{avg}}}. \tag{9}$$

In (7), $T^{\mathrm{nc}}_{\mathrm{avg}}$ is the sum of: 1) the average transmission time; 2) the average queuing time; and 3) the processing time with NC. In our evaluations, we use the measured values of queuing and processing times [9] in computing our analytical results since it is hard to derive exact expressions for these.

B. Experimental Validation and Inferences

Next, we validate the results from our analysis with real WiFi testbed measurements. We also draw inferences from our observations toward later formulating a set of guidelines on which PACE, our encoding logic, is based.

We defer a detailed description of our wireless testbed [34] to Section V. In a nutshell, our testbed consists of Soekris net4826 boxes that run 802.11a/g. The NC functionality in our experiments is managed by the novel NCRAWL software platform[9], which has been designed specifically to accommodate multiple bit rate NC measurements. While we have cross-verified a part of our measurements with the COPE platform [1], we omit those results here in the interest of space. Each experiment lasts for approximately 5 min and is repeated 20 times, for both the NC and store-and-forward cases; for each run, we log the achieved average throughput with each strategy.

We conduct an extensive set of testbed experiments across different topologies in terms of node populations and link qualities. In particular, we examine local topologies (with a single relay node) wherein we vary the packet delivery ratio ($\mathrm{PDR} = 1-\mathrm{PER}$) and the bit rate on both direct and overhearing links; we consider PDR values that range between 0.2 and 0.8, and bit rate values ranging between 6 and 12 Mb/s.

We observe that the analytical results match the experimental results fairly well in all the scenarios considered (as seen in the corresponding figures discussed below).

1) Varying the PDR on a Overhearing Link With Fixed Rate: We consider various settings wherein we fix the bit rate to a specific value, while we vary the PDR on one of the overhearing links. The PDR on all other links is "1"; we adjust the transceiver positions to ensure that this is the case. Our goal here is to observe how the quality of overhearing links affects the efficacy of NC. For example, in the network setting of Fig. 1, we vary the PDR on the link *Alice-Emma*.

a) Case of Two Overhearing Links (the X Topology): Fig. 3 (left) depicts the average per-user throughput versus PDR for the overhearing link between Alice and Emma, when the transmission rate (on all links) is 6 Mb/s. We observe that

[2]Encoded packets are unicast to a specific node; all other recipients that successfully decode it report to the relay the identities of the native packets that they have successfully obtained from decoding, as in [1] and [9].

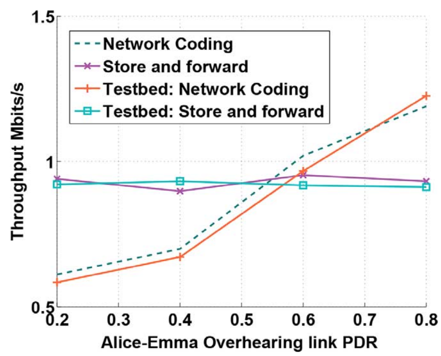


Fig. 3. Varying the PDR for the overhearing link between Alice and Emma.

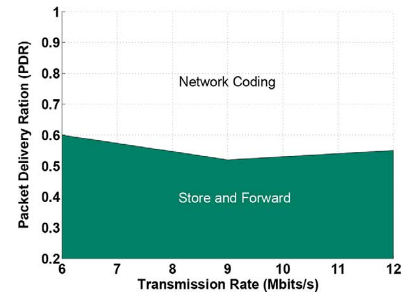


Fig. 4. Value of the PDR on overhearing links affects the efficacy of NC (X-topology).

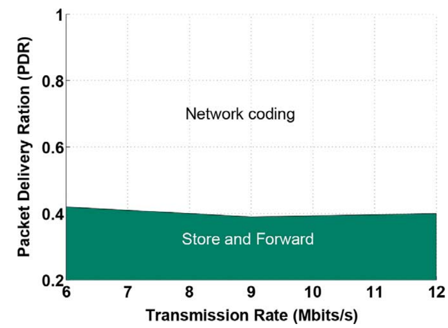


Fig. 5. Value of the PDR on overhearing links affects the efficacy of NC (Wheel topology).

the throughput with NC is higher than that with store-and-forward when the PDR on the link *Alice-Emma* is above 0.5; for values lower than 0.5, the application of NC hurts the average long-term throughput. This drop in the throughput with NC occurs due to the inability of Emma to successfully overhear Alice's transmissions. This renders the decoding of Jack's encoded packets impossible for her. We vary the PDRs on the other links, but find that if the overhearing link is poor, it makes no difference, i.e., NC always performs worse than store-and-forward (we do not present additional results here). Similar results are also seen with the other transmission bit rates. Based on these experiments, we conclude that *the decision on whether to apply NC should consider the qualities of the overhearing links; if the PDR is low on such links, NC is likely to degrade the network throughput.*

b) Cases With Higher Numbers of Overhearing Links (Wheel Topology): Next, we compare the requirements on the overhearing link qualities with the X (2 flows) and wheel (3flows) topologies for the range of rates considered. Figs. 4 and 5 depict results for different transmission rates versus varying PDR values for overhearing links. In particular, we seek to find the sweet spot where NC outperforms store-and-forward. The x-axis depicts the transmission rate, and the y-axis indicates the various PDRs achievable at those rates. The green (dark) region depicts the PDR (for each rate) where store and forward outperforms network coding. If PDRs are in the white (light) region, network coding should be chosen for operations. We observe that NC gains compensate for some of the losses due the link errors. Specifically, in the X topology (Fig. 4), NC outperforms store-and-forward when the PDR is greater than 60% on all overhearing links. In the wheel topology (Fig. 5), NC outperforms store-and-forward when PDR is greater than 40%. This is because in the wheel topology case, with NC the relay typically encodes two or three packets together, and thus the required number of outgoing transmissions is reduced; this compensates for the overhearing link losses to some extent. On the other hand, one would expect that the existence of more overhearing links with low PDR in wheel topologies increases the probability of erroneous overhearing and thereby decreases the achieved throughput with NC. However, our experiments demonstrate that a PDR $>$ 0.4 on all overhearing links is sufficient for NC to perform better than store-and-forward; the reduced number of required transmissions compensates for the link losses on the overhearing links. Hence, we conclude that *the performance of NC when applied on wheel topologies is less sensitive to the PER (on the overhearing links) than when NC is applied on X topologies.*

c) Varying the PDR on the the Direct (Nonoverhearing) Links: We classify the direct links as incoming (e.g., Jim $\rightarrow$ Jack) or outgoing (e.g., Jack $\rightarrow$ Emma) depending on their relative positions with respect to the relay. We perform an exhaustive set of experiments and make several observations, but do not present all the results here. We only present a key set of results instead with a transmission bit rate of 12 Mb/s (similar results are seen with other rates). In all these experiments, we maintain a good quality for the overhearing links (i.e., PDR $= 1$).

d) Case With High-Quality Incoming Links: Our first observation is that if all the links are of high quality, NC provides significant gains over the store-and-forward approach. We do not show this result explicitly for space constraints. However, this is captured in the next result that we show in Fig. 6. Here, we maintain high-quality incoming links, but we vary the quality of one of the outgoing links (Jack $\rightarrow$ Bob). We see that NC outperforms store-and-forward always, and especially when the considered link is of high quality. The reason for this is the following. Due to the fair allocation nature of 802.11, store-and-forward can only provide a throughput equivalent to the poorest outgoing link from Jack (e.g., see [35]); thus, transmitting encoded packets at a rate that satisfies this poor receiver is the best one can do (it saves on the transmission over the better link). This is also reflected in Fig. 7, wherein we vary the PDR on both of the outgoing links while maintaining the high quality of the incoming links. *In summary, if the PDR on incoming links to the relay is high, it is always better to use NC. In other words,*

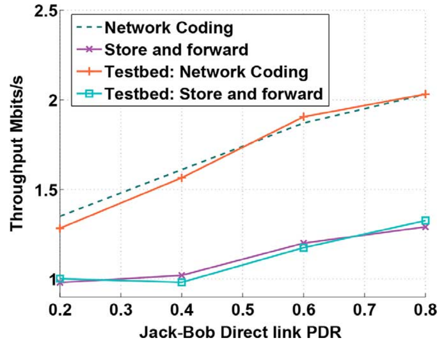


Fig. 6. High PDR values on the relay's incoming links favor NC even if one outgoing link is poor.

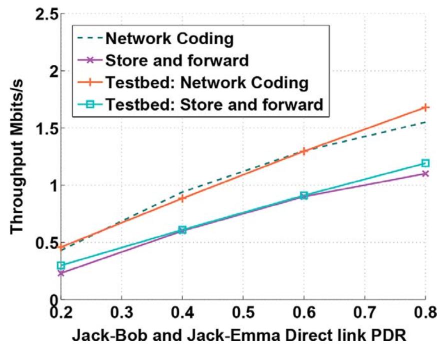


Fig. 7. High PDR on the relay's incoming links favor NC even if both outgoing links are poor.

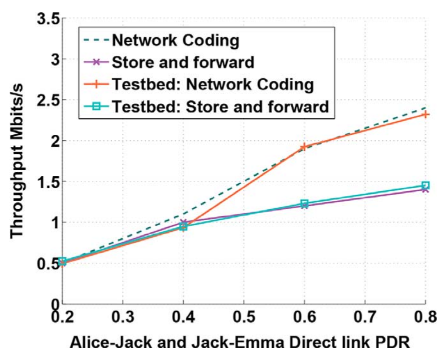


Fig. 8. Low PDR values on the relay's incoming links affects the potential for encoding negatively.

the quality of the outgoing links does not matter as long as the quality of the incoming links is good.

e) Case With Poor Quality Incoming Links: Next, we vary the quality of one of the incoming links (Jim $\rightarrow$ Jack) and one of the outgoing links (Jack $\rightarrow$ Bob) simultaneously. We find that this causes a performance degradation with NC (see Fig. 8) if the pair of links have low PDR. This is because the mismatch in the quality of the incoming links causes a queue imbalance at the relay (Jack). Thus, the likelihood of encoding even if NC is applied by default is very low. As a consequence, there are simply no gains to be had. The processing with NC slightly hurts performance compared to store-and-forward. As the link qualities improve and we approach a regime where all links are again good, the gains due to NC are apparent.

In the final experiment in this section, we vary the quality of both the incoming links to the router, Jack. The overhearing

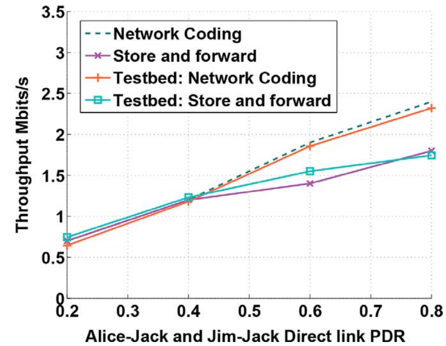


Fig. 9. High PDR values on the relay's outgoing links do not help if the incoming links are poor.

links and the outgoing links are all of good quality, i.e., PDR $\approx$ 1. The throughput results with NC and the store-and-forward scheme are presented in Fig. 9. Again, we notice that when the PDR on the incoming links is low, there are no gains from NC relative to the store-and-forward case. The reason for this is that the input rate to Jack's queues from Alice and Jim are low due to poor PDR. Therefore, Jack typically does not find packets from both senders and, thus, is rarely able to encode packets. As the PDR increases on the incoming links, the benefits due to NC begin to increase. Again, when these links are of good quality ($\mathrm{PDR} = 0.8$), NC outperforms store-and-forward by about 30% in terms of the achieved throughput (as expected, since one transmission is gained relative to the store-and-forward case).

Based on these experiments, we conclude that *the decision on whether to apply NC should consider the qualities of the relay's incoming links; when the PDR on the incoming links is low, coding opportunities may be infrequent. Perfect overhearing does not necessarily imply that network coding should be performed if the quality of incoming links is poor.*

Other Conclusions: Our experiments also lead to two other conclusions (implicit in our discussions above). *a) The outgoing links of the router are a nonfactor in determining whether or not NC should be applied. b) The dependence on the transmission bit rate is not explicit. However, the choice of the bit rate implicitly affects the quality of the overhearing and incoming links and, thus, it would affect the decision on whether NC should be applied or not.*

2) Summary and Scope: While we have presented results with simple topologies, the results hold for more complex wheel topologies (many overhearing links) that inherently present opportunities for NC. The string topology is a special case of the X topology; it is a case where no overhearing is necessary since the native packets are already available at the end destinations. In such cases, as long as the links are of good quality, NC helps; if these links are of poor quality, there are no gains to be had compared to store-and-forward, although there is no significant hit in performance either.

IV. Designing PACE

To summarize our study in Section III, we draw two main conclusions: For any given rate, NC should not be applied when: 1) the quality of the overhearing links is poor ($\mathrm{PDR} < 0.6$ with two overhearing links and $\mathrm{PDR} < 0.4$ with more than

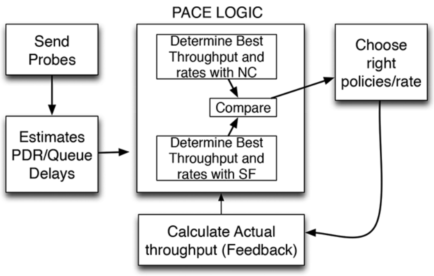


Fig. 10. Main components of PACE.

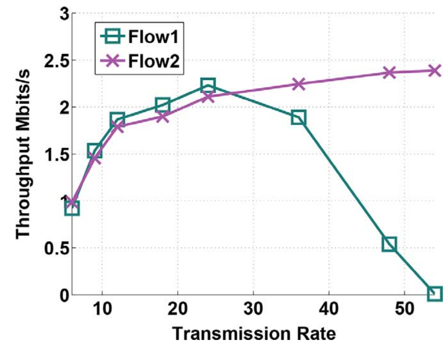


Fig. 11. Throughput versus transmission rates for two store-and-forward flows.

two overhearing links); and 2) the overhearing links are of good quality, but any of the incoming links to the router are of poor quality ($\mathrm{PDR} < 0.4$). Based on these observations, we design our decision logic engine, PACE, next (see Fig. 10).

A. Design Overview

The goal of PACE is to regulate the use of NC at routers. However, it is difficult to apply the above rules directly in a multirate setting since the properties of the links depend on the transmission bit rate in use. We describe how PACE determines whether or not NC is to be applied and the specific transmission rates to be used at each local topology where NC can be potentially invoked.

B. Assessing the Quality of Links

The first step in the process is to acquire the quality of the different links (in terms of PDR) in a local topology. Here, PACE leverages the ETT probing mechanism [36]. Specifically, each node periodically transmits probes at different rates and reports the percentage of received probes from each neighbor. With this, the relay obtains accurate information about the PDR for every rate, on each link in the neighborhood. In addition, since we use the model derived in Section III, and the average queueing delays are needed here, we modify the probe formats to allow each node to report its average queuing delay over the past 10 packets to the router.

C. Determining the Best Throughput With NC

Next, PACE seeks to determine the best throughput with NC. As the transmission rates increase, the quality of the overhearing links could potentially degrade. PACE determines the highest rate (say $\mathcal{R}_{\mathrm{NC}}$) at which the link qualities satisfy the requirements mandated by our guidelines above for invoking NC. It is easy to verify that this rate $\mathcal{R}_{\mathrm{NC}}$ provides the best case for NC. Specifically, at lower rates, lower throughputs are achieved. More importantly, the NC throughput is most likely higher than the store-and-forward case at rate $\mathcal{R}_{\mathrm{NC}}$ (since the conditions mandated by the guidelines for applying NC are satisfied at this rate). However, if the rate is further increased, the store-and-forward approach could deliver higher throughputs than NC (but this is not known at this point). The router (e.g., Jack) then applies our analytical model to compute this best case throughput (with rate $\mathcal{R}_{\mathrm{NC}}$) for NC (say $\mathcal{T}_{\mathrm{NC}}$).

D. Choosing the Policy

Now that PACE has determined the highest throughput with NC, the question that has to be answered is the following: *"Is a higher throughput possible with store-and-forward with higher rates?"* At rates higher than $\mathcal{R}_{\mathrm{NC}}$, the store-and-forward approach may provide higher throughputs than it would at rate $\mathcal{R}_{\mathrm{NC}}$. We seek to examine if the higher throughput with store-and-forward exceeds the throughput achieved with NC at rate $\mathcal{R}_{\mathrm{NC}}$.

If we examine the store-and-forward throughput with different topologies, we see the following behavior. The throughput first increases upon increasing the rate. Either this behavior continues until we hit the maximum transmission rate (e.g., 54 Mb/s with 802.11a) or begins to drop beyond a point. The reason for this is that as we increase rates, the PDR will drop (causing more retransmissions and delays). However, the packet transmission time decreases. Initially, the second factor dominates. At some point it is possible (depending on the topology) that the first factor begins to dominate and, thus, the throughput falls. This behavior is shown for two example store-and-forward flows in a simulated X topology in ns3 (Fig. 11). We could not validate this experimentally since our implementation only supports rates up to 12 Mb/s.

Given the behavior, we do the following. If $\mathcal{R}_{\mathrm{NC}}$ is the highest rate possible (e.g., 54 Mb/s with 802.11a), we simply decide to use NC. If not, and there is just one higher rate (say $\mathcal{R}_{\mathrm{NC}} = 48$ Mb/s with 802.11a), we simply check the throughput of the store-and-forward case at this higher rate using our analytical model. If this computed throughput (say $\mathcal{T}_{\mathrm{SF}}$) is higher than $\mathcal{T}_{\mathrm{NC}}$, we choose store-and-forward; otherwise we choose NC. If there are two or more higher rates, we begin with the rate that is immediately higher than $\mathcal{R}_{\mathrm{NC}}$ and compute the store-and-forward throughput with that rate and the next higher rate. If the throughput is increasing, we keep checking the throughputs at the higher rates, until we hit the peak throughput (as suggested by Fig. 11) or the maximum rate. The throughput with the store-and-forward at that rate is now the highest throughput possible with that approach ($\mathcal{T}_{\mathrm{SF}}$). We compare the values of $\mathcal{T}_{\mathrm{SF}}$ and $\mathcal{T}_{\mathrm{NC}}$ as before and choose the winning policy. Algorithm 1 shows the algorithmic steps followed by PACE. The counter represents the number of incoming flows to the relay. If they are less than 2, then the relay cannot encode. If they are exactly 2, then we consider a certain threshold value. If there are more than two incoming flows, we consider a different value for the threshold. These values were shown to give the best performance via our measurements.

Note that the PACE logic is transparent to other NC procedures, such as cumulative packet acknowledgments [1], [9], de-

Algorithm 1: PACE Algorithm.

Input: ETT neighbor reports ($< \mathcal{R}_k, p_{S_i R}$ for rate $\mathcal{R}_k >$, $< \mathcal{R}_k, p_{RD_i}$ for rate $\mathcal{R}_k >$, Q_{sf} (at sources)) $\forall i$ and $p_{S_i D_j}$ $\forall i, j$ such that $i \neq j$ and $\mathcal{R}_k = \{6,\ 9,\ 12,\ 18,\ 24,\ 36,\ 54\}$;

Output: select_NC and rate vector $\vec{r}$;

Initialization: counter $\leftarrow 0$, $k \leftarrow \{1\}$,stop $\leftarrow$ false and select_NC $\leftarrow$ true;

//Check if the guidelines with regards to links hold
while ! stop **do**
  $k \leftarrow$ Next k ;
  // k is an index and Next k is not always $= k + 1$
  for $i \leftarrow 0$ **to** n **do**
    if $(1 - p_{S_i R}) > 0.4$ **then**
      counter $\leftarrow$ counter + 1;
  if counter < 2 **then**
    stop $\leftarrow$ true ;
  else if counter == 2 **then**
    // Two flows case
    for $c = 1$ **to** n **do**
      for $j = 1$ **to** n **do**
        // Check that the overhearing link PDR > 0.6
        if $c \neq j \bigcap (1 - p_{S_c D_j}) \geq 0.6$ **then**
          stop $\leftarrow$ true;
  else
    // Three flows or more case
    Repeat the Else if part but for PDR > 0.4;
// Compare best case NC and store and forward throughputs
if $k == 7$ **then**
  select_NC $\leftarrow$ false;
else
  Calculate τ^{nc} at k ;
  Calculate τ^{sf}_{prev}, τ^{sf}_{next} at next k ;
  while $\tau^{sf}_{prev} < \tau^{sf}_{next}$ **do**
    $\tau^{sf}_{prev} \leftarrow \tau^{sf}_{next}$;
    $k \leftarrow$ Next k ;
    Calculate τ^{sf}_{next} ;
    if $\tau^{nc} < \tau^{sf}_{next}$ **then**
      select_NC $\leftarrow$ false;
Set $\vec{r}$;

cisions on which packets to code together, etc., as we discuss in Section VI. Also note that the router (Jack) locally computes the processing delays with NC and uses the computation to estimate the throughput with our analytical model as above. Moreover, PACE may employ alternative schemes for accumulating PDR information at the relay [37].

Remark: Coding a subset of flows instead of coding all-or-nothing is a possible extension for our current work. Our solution will have to be extended in the following ways. First, we need to tag outgoing transmissions or notify destinations about who are the coding participants. Second, the additional constraint of what to code in addition to when to code significantly increases the search space. We defer such possibilities for the future.

V. Evaluating Our Framework

In this section, we evaluate PACE via 1) ns-3 simulations on large-scale topologies, and 2) experiments on our wireless testbed over various topological settings and bit rates. While we have run a very large set of simulations and testbed measurements, we discuss only a subset here. As a high-level observation, our assessment reveals that PACE wisely activates NC on a per-data-flow basis, thereby offering throughput improvements of as much as 350% in some specific use-cases.

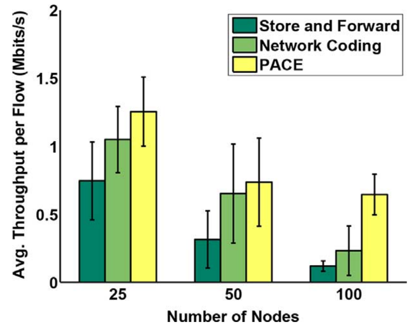


Fig. 12. Simulation results when the transmission rate is 12 Mb/s.

A. Evaluating Pace via ns-3 Simulations

We consider both grid and random topologies with 20, 50, and 100 nodes. We have set the received signal strength threshold for correct decoding to be equal to -80 dBm for all nodes, while the data packet size is 1500 B. We employ the ns-3 Friis propagation model, and we consider the 802.11a mode of operation. The maximum distance between any two nodes is 300 m; we assume that there is no mobility. In addition, in our simulations we select random senders and destinations, which typically are separated by multiple hops; we apply the AODV protocol (implemented in ns-3) for routing (our approach does not depend on the routing protocol in use). PACE is applied locally at every router, and native packets may be encoded/decoded multiple times as they are forwarded along multiple hops along their route to the destination. We vary the number of data flows between 2, 5, and 10. We first evaluate PACE over fixed-rate topologies, and subsequently we consider multirate possibilities with rate adaptation.

In multihop scenarios, we assume that a routing algorithm is in place. Hence, a route for each flow is specified. For simplicity, consider an intermediate node where two flows converge. The relay indicates the transmission rate for the preceding nodes (flows) by consider the following node (on the route) as temporary destination. In other words, the local topology is considered for whether or not to apply network coding at that hop. At each hop, independent decisions are made.

1) PACE Offers Benefits Over Blind NC Application in Large-Scale Topologies: We first consider a grid topology with a fixed rate of 12 Mb/s. As shown in Figs. 12–14, PACE always offers a noticeable throughput gain, due to its ability to make correct decisions on whether or not to apply NC; the gains could be as high as 350% (in the 100-node case). The benefits increase as the scale of the network increases and the routes are longer; this is because the gains on each local hop add up, and the more the opportunities, the higher the gain.

From Figs. 13 and 14, we observe that higher bit rates lead to less modest performance gains with PACE. This is directly attributable to the degraded quality of links at the higher rates. As

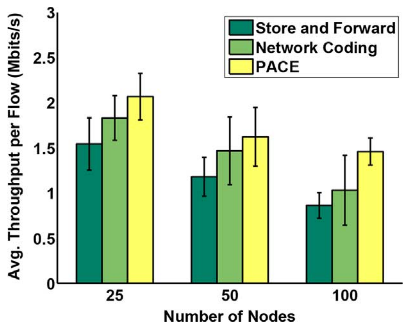


Fig. 13. Simulation results when the transmission rate is 36 Mb/s.

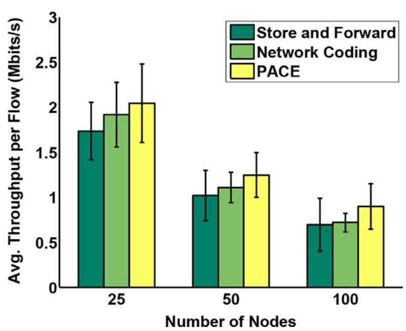


Fig. 14. Simulation results when the transmission rate is 54 Mb/s.

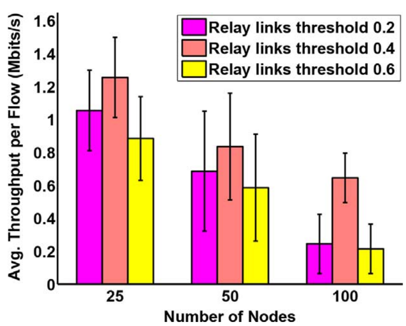


Fig. 15. Varying the relay incoming threshold between 0.2, 0.4, and 0.6.

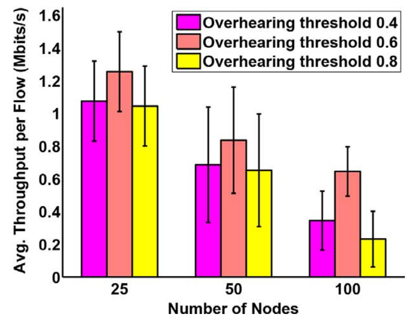


Fig. 16. Varying the overhearing threshold between 0.4, 0.6, and 0.8.

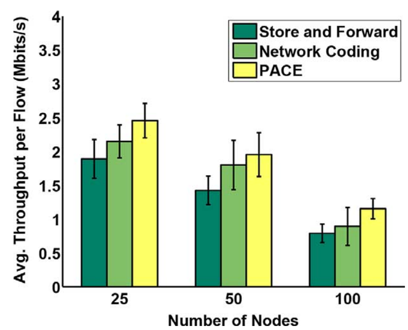


Fig. 17. PACE versus NC and store-and-forward for ETT message length of 256 B.

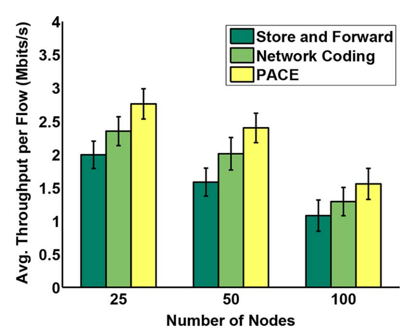


Fig. 18. PACE versus NC and store-and-forward for ETT message length of 512 B.

seen in the figures, NC itself offers more modest benefits compared to a traditional store-and-forward setting. Here, the increased likelihood of encountering poor overhearing link qualities, as well as poor incoming link qualities, often precludes the use of NC. However, PACE still makes the right decisions at each local router on whether or not to use NC.

2) Choice of Parameters for PACE: Next, we show that the choices we make with PACE (PDR on overhearing links should be higher than 0.6, and the PDR on the incoming links should be greater than 0.4) are indeed the best choices in larger-scale (random network) settings. Fig. 15 shows that if the threshold on the incoming link qualities is changed to 0.2 or 0.6, a wrong decision is made and could lead to up to a 3-fold degradation in throughput. Similarly, Fig. 16 shows that a wrong decision on the threshold for the overhearing link qualities could degrade the throughput by up to three times.

3) Sensitivity to the ETT Probe Size: As discussed, PACE relies on ETT probes, exchanged among neighbors to determine link qualities and average queueing times. In Figs. 17 and 18, we consider probe sizes of 256 and 512 B, respectively. The throughput gains with the 512-B probe packet yields more accurate assessments of link qualities and, thus, offers a higher throughput than if a probe size of 256 B is used.

4) PACE With Rate Adaptation: Next, we evaluate the ability of PACE to jointly choose the policy (NC or store-and-forward) and the transmission rate to be used with the chosen policy. Since this involves the "choice" of the right transmission rate, we compare PACE to NC and store-and-forward in conjunction with a popular rate adaptation algorithm, viz., the Adaptive Auto Rate Fall-back (AARF algorithm [38]). Fig. 19 shows that PACE achieves a throughput gain of approximately 15% as compared to the NC case in the grid topology considered.

5) Evaluating PACE Via Testbed Measurements: We have implemented PACE in Click [39] as an embedded software module in the NCRAWL platform [9]. We experiment on a wireless testbed deployed on the third floor of the Computer Science building at the University of California (UC), Riverside, CA, USA; the deployment is depicted in Fig. 20. The nodes are based on the Soekris net4826 hardware configuration and run a Debian Linux distribution with kernel v2.6 over NFS. Each node is equipped with a WN-CM9 wireless mini-PCI card, which carries the AR5213 Atheros chipset. Every card is connected to a 5-dBi-gain external omnidirectional antenna. We experiment with the 802.11a mode of operation in order to avoid interference from the collocated campus WLANs. All devices set their transmission powers to 16 dBm. We use fully saturated UDP traffic; the default data packet size is 1500 B.

6) Implementation Details: The proposed enforcement logic is implemented as a thin sublayer within the MAC layer; at this layer, the PACE logic determines the best forwarding paradigm (network coding or store-and-forward), as well as the appropriate data rate to be used with the chosen paradigm. PACE is composed of three modules: i) Link PDR measurement module; ii) Policy-Aware Decision Engine; and iii) Feedback module. The *Link PDR measurement module* is responsible for gathering information about the quality of the links incident on the node (PDR). Specifically, we modify the standard probing mechanism to incorporate ETT-type probing [36]; this allows the module to estimate the PDR on each link at each possible transmission bit rate. Each node periodically reports the estimated PDR on each of its incident links to its neighbors. The relay node collects the reports and calculates the PDR for each link in its local topology (linear, X, or wheel as the case may be). The *Policy-Aware Decision Engine* is the heart of our approach. It decides whether or not to invoke network coding. In addition, it selects the transmission bit rates that maximize the throughput given the current channel conditions (PDRs). The logic used by this module is a direct application of what was described in Section IV. The *Feedback module* is responsible for informing the transmitters about the chosen rates.

7) Experiments With Fixed Rates: We have performed testbed measurements on "X"-type subtopologies on our testbed. In these experiments, we change the qualities of the links to create 15 different subtopologies. The testbed setup is the same as was described in Section III. The results are averaged over different link PER values. Fig. 21 shows the average throughput with various tested transmission rates. We observe that PACE outperforms the blind NC and store-and-forward application strategies throughout. Since the experiments are over a single local hop, the gains over NC are modest ($\approx 10\%$).

8) Experiments With Rate Adaptation: Next, we consider a multirate case where PACE also makes decisions on the rate to use. In Fig. 22, we show testbed results for PACE in five different scenarios. Each scenario is created essentially by randomly choosing a set of links in our topology (with different link qualities). For NC, we choose the best rate (as predicted by PACE). However, no store-and-forward option is involved. PACE appropriately chooses between NC and store-and-forward, depending on the scenario. PACE achieves the same throughput as NC in the worst-case scenario; this is a case where the best decision is NC almost all the time, and PACE makes that decision. However, on average, it achieves 10% gain. The gain is again modest in these cases since only a single hop is involved, and the best rate for NC is being used. However, it is important to note that in three out of the five cases, PACE offers a higher throughput. We have considered several other scenarios (with other sets of links), and we find that in about 40% of the considered cases, we observe gains with PACE (due to its properly choosing the store and forward option).

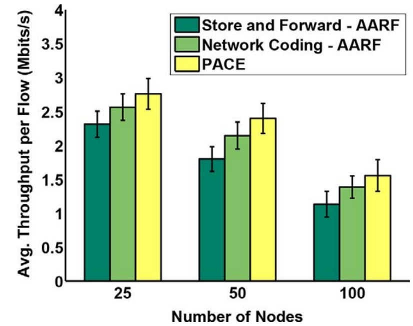


Fig. 19. PACE performs better than the AARF algorithm.

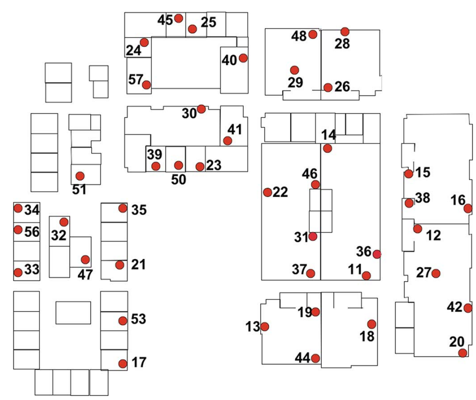


Fig. 20. Deployment of our testbed at UC Riverside. Nodes are represented by dots along with their IDs.

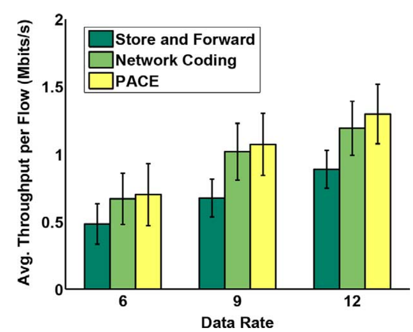


Fig. 21. Experimental results in the X-topology for different transmission rates.

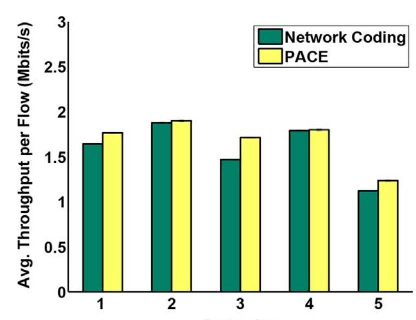


Fig. 22. PACE offers the highest benefits at all different rates in the X-topology.

9) Comparison Between PACE and COPE: Next, we evaluate PACE in realistic scenarios by comparing the average

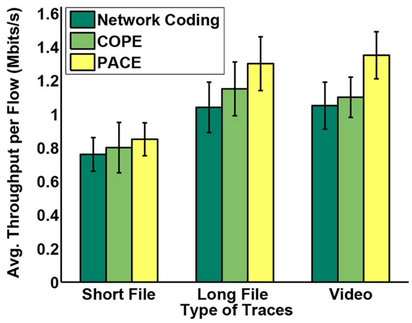


Fig. 23. Comparison between PACE, COPE, and NC for different types of traces.

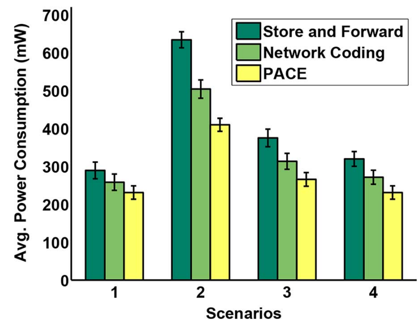


Fig. 24. Average power consumption for different channel conditions scenarios.

throughput to that of COPE or only network coding. We configured COPE to switch off network coding if the loss rate exceeds 20%, as in the default case [40]. The number of nodes are set to 100 and the results are averaged over 10 runs. Each run lasts for 5 min. One input flow consists of real video file traces obtained from [41]. Other short/long file traces were collected by downloading short and long files and capturing the traces using [42]. The file sizes are 200 kB and 50 MB, respectively. These were also used as candidate input traffic traces. Fig. 23 shows that PACE significantly performs better than COPE and NC for video and long file traces. On the other hand, for shorter files, the average throughput of PACE and COPE resemble that of using only NC. This is because with short files, the traffic loads are low/modest. Thus, the relays often do not find opportunities to encode. Thus, store-and-forward is used more often than not in all cases.

10) Evaluation of Power Consumption With PACE: PACE decreases the overall number of retransmissions compared to using either network coding or store and forward exclusively. Thus, PACE saves the consumed energy. In order to quantify the savings, we perform a set of experiments and record the total number of (re)transmission(s) in the following cases: 1)Store and Forward; 2) Network Coding; and 3) PACE. We compute the power consumed, on average, with our Soekris net4826 boxes with various transmission rates and payload sizes. Thereafter, we perform a direct mapping of the number of (re)transmissions to power consumption. We perform four sets of experiments during the day with the X-topology, each spanning an hour. In each experimental scenario, we reposition the nodes; we change the quality of the links in the topology at arbitrary instances in time (vary frequency) to characterize variations in conditions. The transmission bit rates are chosen as per the strategy. Fig. 24 shows the power consumption of PACE compared to that with store and forward and network coding. We find that in all sets of experiments, PACE outperforms both network coding and store-and-forward in terms of the power consumed. In scenario 1, the links were primarily of good quality, and we did not change the quality of the links often; thus, the gains over pure network coding are modest. In scenario 2, we varied the quality of the links with the highest frequency; PACE invokes network coding or store-and-forward as appropriate and thereby decreases the overall number of required transmissions. It therefore provides significant power savings 20% as compared to NC. The frequency with which we changed the qualities of the links in scenarios 3 and 4 were less than that in scenario 2 but more than that in scenario 1; thus, as one might expect, the gains are less than that in scenario2, but slightly higher than that in scenario 1. Note here that if the occurrence of poor-quality links becomes prevalent, all schemes require higher numbers of retransmissions and, thus, experience higher power consumptions.

VI. Discussion

In this section, we discuss certain design aspects of PACE.

Focus on Single-Router Topologies: In this paper, we primarily focus on local wireless NC settings involving a single router; packets encoded by a router are decoded at the next hop. Such scenarios are prominent in WLAN deployments [10], where clients associated with the same access point (AP) exchange data; examples of application include WiFi-based home networking and applications such as online gaming and video streaming [5], [25]. In such cases, the AP essentially plays the role of the relay encoding packets exchanged among its clients. We envision that practical wireless NC will mostly be applied in such topologies, given the ease of the decision-making process and the simplicity in gathering topological information at the relay site. However, we also show that this does not preclude the use of PACE in large-scale ad hoc and mesh topologies with multihop routes. In particular, in multihop settings, a packet may be encoded and decoded multiple times as it traverses multiple relays (depending on the topology) along its route to the final destination. As long as an encoded packet does not traverse more than one hop, our work is directly applicable in such settings as well: PACE will make local NC decisions at the individual routers. We plan to extend PACE for more complex topologies, such as the butterfly [1], in our future work.

Applicability of Our Framework With Other NC Architectures: We have implemented the PACE logic as part of the NCRAWL software platform [9]. We use NCRAWL given its lightweight implementation and its ability to efficiently support multirate experiments. Clearly, certain NC-related choices of other platforms, such as COPE [1], are compatible with our scheme. For example, the decision on which packets are to be encoded together and when generally depends on the relay configuration; it could also depend on other factors, such as the traffic profiles and network policies. Such decisions are orthogonal to our study, and we simply adopt the same assumptions for such procedures from prior NC frameworks such as COPE; for example, we use a fixed preconfigured value for a timer upon the expiration of which, temporarily stored packets are examined, dequeued, and coded together to form encoded packets at the relay. Since PACE is independent of these procedures, it is directly applicable with other NC architectures and solutions [1], [10], [7].

Multiple Relays (M)*:* Solving the problem for M multiple relays (simultaneously setting a set of sources to a set of destinations) is challenging for the following reasons: 1) If each of the M relays transmits information without coordination, it could lead to wasteful transmissions (more like a broadcast). 2) If the source has to choose among the M relays (to choose a single relay), a new design that accounts for factors such as the load on each relay, will be necessary. 3) If, instead, the relays were to coordinate of themselves, the coordination across relays to maximize coding opportunities requires communications between relays and, thus, incurs overhead. The overhead versus tradeoff benefits need to be studied. In all of the above cases, we believe that a different and more complicated system design is needed. Specifically, if one is not careful, there may be delays incurred in sending encoded packets, due to channel occupancy for control information exchange as mentioned above.

VII. Conclusion

In this paper, we argue that when NC is applied in a careless manner, it may cause significant throughput degradation in multirate environments. In many cases, a traditional store-and-forward approach may be preferable to NC. Via extensive experiments and an analysis, we characterize the regimes where NC offers throughput benefits and those where it does not. This study allows us to formulate a set of guidelines regarding when NC should be applied. Based on these guidelines and our analytical model, we design PACE, a policy-aware coding enforcement logic, which allows a router to switch between NC and store-and-forward modes depending on the link qualities. We evaluate PACE both on a simple prototype testbed and via extensive simulations. Our evaluations show that PACE could potentially offer network-wide throughput improvements of up to 350% as compared to a fixed rate NC policy that is blindly applied.

References

[1] S. Katti *et al.*, "XORs in the air: Practical wireless network coding," *IEEE/ACM Trans. Netw.*, vol. 16, no. 3, pp. 497–510, Jun. 2008.

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

[3] T. S. Kim *et al.*, "A framework for joint network coding and transmission rate control in wireless networks," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[4] R. Kumar, S. Tati, F. De Mello, S. V. Krishnamurthy, and T. La Porta, "Network coding aware rate selection in multi-rate IEEE 802.11," in *Proc. IEEE ICNP*, 2010, pp. 92–102.

[5] H. Seferoglu and A. Markopoulou, "Distributed rate control for video streaming over wireless networks with intersession network coding," in *Proc. Packet Video*, 2009, pp. 1–10.

[6] J. Le *et al.*, "How many packets can we encode?—An analysis of practical wireless network coding," in *Proc. IEEE INFOCOM*, 2008, pp. 1040–1048.

[7] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: Design, implementation, and performance evaluation," in *Proc. ACM SIGMETRICS*, 2008, pp. 85–96.

[8] S. Katti, D. Katabi, H. Balakrishnan, and M. Medard, "Symbol-level network coding for wireless mesh networks," in *Proc. ACM SIGCOMM*, 2008, pp. 401–412.

[9] I. Broustis, G. Paschos, D. Syrivelis, L. Georgiadis, and L. Tassiulas, "NCRAWL: Network coding for rate adaptive wireless links," Tech. Rep., arXiv:1104.0645v1, 2011.

[10] E. Rozner *et al.*, "ER: Efficient retransmission scheme for wireless LANs," in *Proc. ACM CONEXT*, 2007, Art. no. 8.

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

[12] S. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371–381, Feb. 2003.

[13] C. Gkantsidis and P. Rodriguez, "Network coding for large scale content distribution," in *Proc. IEEE INFOCOM*, 2005, pp. 2235–2245.

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

[15] T. Ho, M. Medard, M. Effros, and D. Karger, "On randomized network coding," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2003, pp. 11–20.

[16] T. Ho and R. Koetter, "Online incremental network coding for multiple unicasts," DIMACS Working Group on Network Coding, 2005.

[17] Y. Wu, P. A. Chou, and S. Y. Kung, "Information exchange in wireless networks with network coding and physical-layer broadcast," Microsoft-TR-2004-78, 2004.

[18] Z. Li and B. Li, "Network coding: The case for multiple unicast sessions," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2004, pp. 1–9.

[19] A. Khreishah, I. M. Khalil, and J. Wu, "Polynomial time and provably efficient network coding scheme for lossy wireless networks," in *Proc. IEEE MASS*, 2011, pp. 391–400.

[20] S. Sharma, Y. Shi, J. Liu, Y. T. Hou, and S. Kompella, "Is network coding always good for cooperative communications?," in *Proc. IEEE INFOCOM*, 2010, pp. 1–9.

[21] C.-C. Wang, "On the capacity of wireless 1-hop intersession network codinga broadcast packet erasure channel approach," *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 957–988, Feb. 2012.

[22] C. H. Liu and F. Xue, "Network coding for two-way relaying: Rate region, sum rate and opportunistic scheduling," in *Proc. IEEE ICC*, 2008, pp. 1044–1049.

[23] L. F. M. Vieira, A. Misra, and M. Gerla, "Performance of network coding in multi-rate wireless environments for multicast applications," in *Proc. IEEE GLOBECOM*, 2007, pp. 1–6.

[24] B. Scheuermann, W. Hu, and J. Crowcroft, "Near-optimal co-ordinated coding in wireless multihop networks," in *Proc. ACM CONEXT*, 2007, Art. no. 9.

[25] H. Seferoglu and A. Markopoulou, "Opportunistic network coding for video streaming over wireless," in *Proc. Packet Video*, 2007, pp. 191–200.

[26] D. S. Lun *et al.*, "Minimum-cost multicast over coded packet networks," *IEEE Trans. Inform. Theory*, vol. 52, no. 6, pp. 931–938, Jun. 2006.

[27] H. Seferoglu, A. Markopoulou, and U. Kozat, "Network coding-aware rate control and scheduling in wireless networks," in *Proc. ICME*, 2009, pp. 1496–1499.

[28] K. Srinivasan *et al.*, "The $\mathcal{K}$ factor: Inferring protocol performance using inter-link reception correlation," in *Proc. ACM MobiCom*, 2010, pp. 317–328.

[29] S. Chachulski *et al.*, "Trading structure for randomness in wireless opportunistic routing," in *Proc. ACM SIGCOMM*, 2007, pp. 169–180.

[30] H. Seferoglu, A. Markopoulou, and K. K. Ramakrishnan, "I^2NC: Intra-and inter-session network coding for unicast flows in wireless networks," in *Proc. IEEE INFOCOM*, 2011, pp. 1035–1043.

[31] S. Rayanchu, S. Sen, J. Wu, S. Banerjee, and S. Sengupta, "Loss-aware network coding for unicast wireless sessions: Design, implementation, and performance evaluation," *Perform. Eval. Rev.*, vol. 36, no. 1, pp. 85–96, 2008.

[32] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw.*, 2008, pp. 315–326.

[33] *ANSI/IEEE 802.11-Standard*, 802.11, 1999.

[34] University of California, Riverside, CA, USA, "UCR wireless testbed," 2011 [Online]. Available: http://networks.cs.ucr.edu/testbed

[35] M. Heusse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance anomaly of 802.11b," in *Proc. IEEE INFOCOM*, 2003, pp. 836–843.

[36] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proc. ACM MOBICOM*, 2004, pp. 114–128.

[37] I. Broustis, K. Pelechrinis, D. Syrivelis, S. V. Krishnamurthy, and L. Tassiulas, "Quantifying the overhead due to routing probes in multi-rate WMNs," in *Proc. IEEE WCNC*, 2010, pp. 1–6.

[38] Y. Xi *et al.*, "Adaptive multirate auto rate fallback protocol for IEEE 802.11 WLANs," in *Proc. IEEE MILCOM*, 2006, pp. 1–7.

[39] "Click modular router," 2014 [Online]. Available: http://read.cs.ucla.edu/click/

[40] S. Katti *et al.*, "XORs in the air: Practical wireless network coding," in *Proc. ACM SIGCOMM*, 2006, pp. 243–254.

[41] "Video traces for network performance evaluation," 2014 [Online]. Available: http://trace.eas.asu.edu/tracemain.html

[42] "Wireshark," [Online]. Available: http://www.wireshark.org

Ahmed Osama Fathy Atya (S'10) received the B.Sc. degree in computer engineering from Cairo University, Giza, Egypt, and the M.Sc. degree in wireless technology from Nile University, Cairo, Egypt, and is currently pursuing the Ph.D. degree in computer science at the University of California, Riverside, CA, USA.

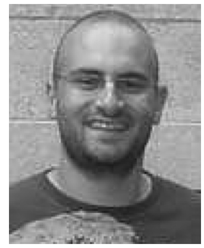
His areas of interest are networks with specific emphasis on wireless ad hoc networks, information security, wireless networks, and networks security.

Ioannis Broustis (S'05–M'08) received the Diploma in electronics and computer engineering from the Technical University of Crete, Chania, Greece, in 2003, and the M.Sc. and Ph.D. degrees in computer science and engineering from the University of California, Riverside, CA, USA, in 2005 and 2007, respectively.

He is currently with AT&T Labs—Research, Florham Park, NJ, USA. From 2010 until 2012, he was with Alcatel-Lucent, Murray Hill, NJ, USA. During 2008, he was a Researcher with the Center for Research and Technology Hellas, Thessaloniki, Greece, as well as an Adjunct Assistant Professor with the University of Thessaly, Volos, Greece. During 2006, he was with Intel Research, Cambridge, U.K., while during 2005 he was with Nokia R&D, Boston, MA, USA. His research interests include cross-layer network protocol design, network security, development, and testbed experimentation for wireless networks.

Shailendra Singh (S'09–M'14) received the B.Tech. degree in computer science and engineering from the Gautam Buddh Technical University (GBTU), Lucknow, India, in 2007, and the M.S. degree in computer science from the State University of New York, Binghamton, NY, USA, in 2009, and is currently pursuing the Ph.D. degree in computer science at the University of California, Riverside, CA, USA.

His research interests include wireless networks and mobile computing with an emphasis on design and implementation of flexible and adaptive wireless communication systems.



Dimitris Syrivelis was born in 1979 in Volos, Greece. He received the diploma in electrical engineering from the Technical University of Crete, Chania, Greece, in 2003, and the Ph.D. degree from the University of Thessaly, Volos, Greece.

He is currently a Senior Researcher with the CERTH-ITI Institute, Thessaloniki, Greece. He has developed systems and algorithms for wireless network coding, clean slate approaches for the Internet like information-centric networking, and software defined networks. He has several publications in premier IEEE, ACM, and Usenix conferences and several open-source contributions on a variety of tools that support systems research. His research interests span the area of system architecture design and implementation with the main focus being novel networking systems.

Srikanth V. Krishnamurthy (S'94–M'99–SM'07–F'12) received the Ph.D. degree in electrical and computer engineering from the University of California, San Diego, La Jolla, CA, USA, in 1997.

From 1998 to 2000, he was a Research Staff Scientist with the Information Sciences Laboratory, HRL Laboratories, LLC, Malibu, CA, USA. Currently, he is a Professor of computer science with the University of California, Riverside, CA, USA. His research interests are primarily in wireless networks and security.

Dr. Krishnamurthy was the Editor-in-Chief of the *ACM Mobile Computing and Communications Review* from 2007 to 2009. He is the recipient of the NSF CAREER Award from ANI in 2003.

Tom La Porta (S'85–M'87–SM'99–F'02) received the B.S.E.E. and M.S.E.E. degrees from The Cooper Union, New York, NY, USA, in 1986 and 1987, respectively, and the Ph.D. degree in electrical engineering from Columbia University, New York, NY, USA, in 1992.

He is the William E. Leonhard Chair Professor with the Computer Science and Engineering Department, Pennsylvania State University, University Park, PA, USA, as well as the Director of the Networking and Security Research Center. Prior to joining Penn State in 2002, he was with Bell Laboratories, Holmdel, NJ, USA, since 1986. He was the Director of the Mobile Networking Research Department, Bell Laboratories, Lucent Technologies, where he led various projects in wireless and mobile networking. He was an adjunct member of faculty with Columbia University for 7 years, where he taught courses on mobile networking and protocol design. He has published numerous papers and holds 35 patents. His research interests include mobility management, signaling and control for wireless networks, security for wireless systems, mobile data systems, and protocol design.

Prof. La Porta is a Bell Labs Fellow. He was the founding Editor-in-Chief of the IEEE TRANSACTIONS ON MOBILE COMPUTING, and currently serves on its Steering Committee (he was Chair from 2009 to 2010). He served as Editor-in-Chief of the *IEEE Personal Communications Magazine*. He was the Director of Magazines for the IEEE Communications Society and was on its Board of Governors for 3 years. He received the Bell Labs Distinguished Technical Staff Award in 1996 and an Eta Kappa Nu Outstanding Young Electrical Engineer Award in 1996. He also won Thomas Alva Edison Patent Awards in 2005 and 2009.