

Article

CoFi: Coding-Assisted File Distribution over a Wireless LAN

Naji Ma ^{1,2,*} and Ming Diao ¹

¹ College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China; diaoming@heu.edu.cn

² Information and Network Center, Heilongjiang University of Science and Technology, Harbin 150022, China

* Correspondence: 5203datashare@163.com; Tel.: +86-451-8803-6153

Received: 7 November 2018; Accepted: 25 December 2018; Published: 10 January 2019



Abstract: The wireless channel is volatile in nature, due to various signal attenuation factors including path-loss, shadowing, and multipath fading. Existing media access control (MAC) protocols, such as the widely adopted 802.11 wireless fidelity (Wi-Fi) family, advocate masking harsh channel conditions with persistent retransmission and backoff, in order to provide a packet-level best-effort service. However, the asymmetry of the network environment of client nodes in space is not fully considered in the method, which leads to the decline of the transmission efficiency of the good ones. In this paper, we propose CoFi, a coding-assisted file distribution protocol for 802.11 based wireless local area networks (LANs). CoFi groups data into batches and transmits a random linear combination of packets within each batch, thereby reducing redundant packet and acknowledgement (ACK) retransmissions when the channel is lossy. In addition, CoFi adopts a MAC layer caching scheme that allows clients to store the overheard coded packets and use such cached packets to assist nearby peers. With this measure, it further improves the effective throughput and shortens the buffering delay when running applications such as bulk data transmission and video streaming. Our trace based simulation demonstrates that CoFi can maintain a similar level of packet delay to 802.11, but increases the throughput performance by a significant margin in a lossy wireless LAN. Furthermore, we perform a reverse-engineering on CoFi and 802.11 using a simple analytical framework, proving that they asymptotically approach different fairness measures, thus resulting in a disparate performance.

Keywords: CoFi distribution protocol; delay; harsh channel condition; hybrid coding scheme; IEEE802.11 standard; internet of things; media access control protocol; throughput; Wi-Fi

1. Introduction

The IEEE 802.11 (Wi-Fi) based wireless media access control (MAC) protocol has become a dominant technology that provides convenient access to the internet in a wireless LAN, in which a single access point (AP) can serve the file requests from multiple clients [1]. However, the effective performance of Wi-Fi, in terms of throughput, delay, and stability, is still incomparable to its wireline counterpart. This is mainly because of its sensitivity to ambient interference and the volatile wireless link condition [2]. In particular, the persistent-retransmission based loss protection schemes, and a backoff scheme that does not differentiate losses from congestion, have been identified as the main reasons for the inefficiency of 802.11 [3,4]. Alternative protocols have been proposed. In Reference [5] a new MAC algorithm, called multiple access with salvation army (MASA), which adopts a less sensitive carrier sensing to promote more spatial reuse of the channel, was proposed. The MASA alleviates the problem of a high collision probability by adaptively adjusting the communication distance via “packet salvaging” at the MAC layer. The performance of throughput and packet delivery rate was improved significantly. In Reference [6] authors presented the design, implementation, and evaluation

of a system that opportunistically caches overhead data to improve subsequent transfer throughput in wireless mesh networks. A new backoff scheme which performs well for dense networks resulting in a low collision probability was proposed by Karaca et al. [7]. The backoff scheme opportunistically gives higher priority to users with a high traffic load and better channel conditions, and thus reduced unnecessary contention. A simple window adaptation scheme for backoff in 802.11 MAC protocol was proposed by Shetty et al. [8]. The scheme used constant step size stochastic approximation to adjust collision probabilities to set values. Using an approximate analytic relationship between this probability and the backoff window ensured a high throughput and fairness. Protocols in all the above references replaced these two reasons with other schemes to provide a best-effort packet level service.

In this paper, we propose CoFi, a MAC level file distribution protocol that aims at improving the WLAN throughput and delay performance under harsh channel conditions. CoFi masks the wireless channel variations using coding based batch transmission. Instead of striving to improve per-packet reliability via retransmission, the AP in CoFi continuously transmits random linear combinations of original packets, which are grouped into batches. Once the receiver accumulates a sufficient number of linearly independent packets, it sends a single ACK to acknowledge the successful reception of the entire batch of data. The rationale of random linear network coding lies in removing packet ordering by mixing the information inside each batch [9–11]. With this technique, the scheduling problem within each batch can be avoided since the receiver does not have to be halted because of the failure of a single transmission and the subsequent retransmission attempts. As long as the receiver can collect a sufficient number of coded blocks, it will be able to recover the original file. This advantage provides a natural solution to the notorious fairness problem in 802.11, which degrades the throughput of all clients once one of them is experiencing a severe loss [12].

A further optimization that CoFi uses to improve performance is coding based MAC layer caching. The basic idea of MAC layer caching is to take advantage of the wireless broadcast nature [13]. Each client can overhear the coded packets that are intended for another client. When one of the (AP client) links suffers from a harsh channel condition, CoFi allows another client who has overheard packets intended for the weak client to serve as a relay. The relay replaces the role of the AP, and transmits random linearly combined packets to the weak client, until it can successfully decode. With the coding based batching and caching schemes, CoFi is able to significantly reduce the control overhead and redundant transmissions in traditional 802.11 like protocols, thus improving the throughput performance, while ensures full MAC layer reliability.

In designing the coding based batch transmission and MAC caching protocols, we have tried to improve not just the throughput and reliability, but also the per-packet delay. Note that the throughput-delay relation with network coding is not straightforward, since for each coded chunk of data, the receiver must wait until the entire batch can be decoded. In general, the delay performance of network coding depends on both the end-host computation time and the batch transmission time [14]. The end-host delay is usually negligible on a powerful mobile device, especially when using fast coding implementations [15]. Therefore, the per-packet delay is usually constrained by the batch transmission time, which depends on the channel condition, as well as the batch size. With extensive trace based simulation, we demonstrate that the per-packet delay of CoFi can be close to 802.11, especially in a lossy wireless network, while the effective throughput can be more than twice higher. Such a performance advantage renders it particularly useful for applications such as bulk file transfer and streaming, which can tolerate an initial buffering delay.

The surprisingly large performance gap between CoFi and 802.11 may seem counter-intuitive, given that network coding does not fundamentally improve the network capacity [16]. Towards a more rigorous justification, we perform reverse-engineering of both protocols using theoretical analysis. With a simple analytical model, we prove that CoFi and 802.11 essentially approach different fairness measures in an asymptotic manner, thus leading to disparate throughput efficiency.

The remainder of this paper is organized as follows. In Section 2, we review existing work on network coding and MAC caching protocols that improve the performance of wireless networks.

We continue to discuss the major design issues of CoFi in Section 3, and then describe the implementation CoFi in Section 4. Section 5 evaluates the performance of CoFi, with respect to throughput and delay, as well as the computation cost of network coding. Section 6 presents an in-depth analysis of the asymptotic properties of CoFi, aiming at a theoretical justification of its performance. Finally, Section 7 summarizes the paper and presents our future work.

2. Related Work

Since the pioneering work by Ho et al. [17], randomized network coding has shifted from the information theory domain to practical wireless networking systems. This trend has been marked by the COPE protocol [18], which employs the simplest form of network coding (i.e., XOR coding) for multi-session unicast in wireless mesh networks. COPE allows intermediate forwarders to opportunistically XOR code incoming packets heading towards different next-hops, based on a prior knowledge of the decodability at the intended downstream nodes. The encoding nodes broadcast the coded packets to all down streams, thereby reducing the number of transmissions compared with traditional routing. The MAC-independent opportunistic routing & encoding (MORE) protocol [19] goes one step further by combining random linear network coding with opportunistic multipath routing. Owing to the resilience of random linear network coding to packet losses, MORE achieves 70% higher throughput on average over a traditional best-path routing protocol. Due to the problem that the stopping and waiting “ACK” policy degrades throughput in MORE, Lin et al. addressed the problem [20] by allowing the coexistence of different data segments. In the proposed method, called code opportunistic routing (CodeOR), the source node transmits W (window size) concurrent segments. When the source node receives end-to-end feedback from the destination node, the node adds a new segment to the current window. Kim et al. [21] proposed a novel scattered random network coding (S-RNC) scheme, which takes the advantage of error position diversity to improve the throughput performance in multiple hops wireless networks. The RNC blocks can be classified into different groups according to different bit positions (i.e., error probability), and the lower ones are protected. The sender and relays scatter the bits of these protected coded blocks on “good” bit positions and the rest on “bad” bit positions. Amerimehr et al. [22] investigated the throughput gain of inter-flow network coding over a non-coding scheme on multicast sessions in multi-hop wireless networks. They also defined a new metric, network unbalance ratio, which identifies the amount of unbalance instability among nodes.

Despite the wide interest on the saturation throughput of network coding based protocols, little attention has been paid to its delay performance. In addition, randomized network coding has mostly been applied to improving the performance of a single network flow. When multiple unicast sessions are running concurrently, it is unknown whether network coding can still provide benefits, given its aggressive transmission strategy. In Reference [23], the delay performance gains of network coding are quantified using theoretical analysis, assuming an ideal time division multiple access (TDMA) based cellular network. In designing the CoFi protocol, we focused on the performance of network coding in realistic carrier sense multiple access (CSMA) based wireless LANs. Another deficiency of existing randomized network coding protocol is that they are mostly applied to routing in static mesh networks. For example, MORE [19] requires the extensive measurement of all link conditions in a mesh network, and then runs a centralized algorithm to determine the number of packets sent over each link based on its average quality. In CoFi, we used signal strength as the metric for the selection of relays, which can be realized without any measurement overhead, and which is more adaptable to a mobile scenario.

The idea of opportunistically caching packets is not new. It has been applied in wireless mesh networks in order to reduce the routing overhead [5] and to save redundant transmissions [6]. However, these schemes strongly depend on the underlying scheduling algorithm that determines “which packets to cache” and “which cached packets to transmit”. In CoFi, we take advantage of the inherent

randomized nature of network coding, eliminate the complex scheduling problem, and thus further reduce the file completion time.

Network coding has been applied to the 802.16 worldwide interoperability for microwave access (WiMax) cellular networks as well [24,25], in order to improve its throughput and stability. The 802.11 based WLAN differs from Reference [24] in that all subscribers in the WLAN contend for the same channel. In this case, the delay experienced by each client depends not only on its channel condition, but also on the link quality experienced by others. Furthermore, in a multichannel system such as WiMax, the MAC caching scheme cannot be applied, since nearby links are usually assigned orthogonal sets of sub channels.

3. Coding Based Batching and Caching in CoFi

In this section, we introduce the major design issues in CoFi, i.e., coding based batch transmission and MAC layer caching. We begin with the basic idea of randomized network coding and how it can be applied to simplify both mechanisms.

3.1. Background: Randomized Network Coding

Existing coding based wireless unicast protocols have mostly adopted the following batch based scheme, which was first introduced by Chou et al. [26]. Before transmission, in order to facilitate the encoding of the original data file, it is grouped into batches (Figure 1), each containing n blocks of size k bytes. n and k are termed batch size and block size, respectively. A batch can be represented as a $n \times k$ matrix B , with rows being the n blocks, and columns the bytes (integers from 0 to 255) of each block. In each batch the coding operation is performed. The encoding operation produces a linear combination of the original blocks in this batch by $X = R \cdot B$ (Figure 2), where R is a $n \times n$ matrix composed of random coefficients in the Galois field $GF(2^8)$ [27]. The coding coefficients (rows in R), along with the coded blocks (rows in X), are packetized and transmitted to the receiver.

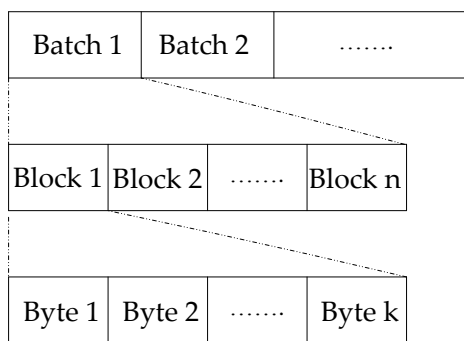


Figure 1. Batch files for random linear encoding.

$$R \cdot B = X$$

| | | |
|-------|---------|---------------|
| Row 1 | block 1 | coded block 1 |
| Row 2 | block 2 | coded block 2 |
| | | |
| Row n | block n | coded block n |

Figure 2. The encoding operation is equivalent to a matrix multiplication $X = R \cdot B$ (within the finite field $GF(2^8)$), where R and B are the random coefficient matrixes, and B is the original data blocks in one batch. The decoding operation is the matrix reversion: $B = R^{-1} \cdot X$.

The decoding operation at the receiver side is the matrix reversion $B = R^{-1} \cdot X$, where each row of X represents a coded block and each row of R represents the coding coefficients accomplished with it [27]. It is required that the matrix R is of full rank, i.e., the receiver must collect n independent coded blocks for this batch to successfully recover the original batch B . Afterwards, it returns a single ACK packet to the sender, which confirms the successful decoding of the entire batch.

Random linear network coding allows a re-encoding operation at the helper node. Specifically, it produces a new block by re-encoding existing blocks it has overheard in this batch. The re-encoding operation replaces the coding coefficients accomplished with the original coded packets with another set of random coefficients. For instance, if we consider the existing coded packets to be rows in the matrix Y , which from the viewpoint of the AP was obtained using $Y = R_y \cdot B$ (B is the original un-coded packets and R_y is the random coefficients), then the helper node may produce a new code block by re-encoding existing packets as $Y^{\sim} = R^{\sim} \cdot R_y \cdot B = R_y^{\sim} \cdot B$. As a result, the original coefficients R_y are replaced by R_y^{\sim} . The re-encoding operation circumvents the packet level scheduling problem in traditional MAC caching protocols [5,6], because by randomly mixing information from all existing blocks, a newly generated coded block is innovative to the receiver with a probability close to 1 [17].

Note that the packet overhead of network coding is negligible, since the size of the coding coefficient (n) is usually much smaller than the packet size (k). This issue has been well recognized in existing work [19].

3.2. Coding Based Batch Transmission

The coding based batch transmission scheme in CoFi resembles the above mentioned randomized network coding. In designing such a batch transmission scheme, a key factor is the parameter settings, including the batch size and the block size. A large batch size provides better protection against packet losses, but also induces higher computation overhead. On the other hand, a smaller block size results in lower computation overhead, but also implies a higher coefficient overhead and lower level of loss resilience. In this paper, we use a small batch size, so that CoFi can never be computation bounded. In a parallel project, we look into an adaptive approach that opportunistically adjusts the batch size and block size according to the variation in the channel condition and CPU load of the wireless node.

An important optimization that CoFi uses to reduce the computation and coefficient overhead is hybrid coding. Specifically, for each batch, a transmitter sends the first n packets without network coding, where n equals the batch size. Afterwards, it performs random linear encoding upon each outgoing packet. In this way, it still ensures each outgoing packet is fresh for the receiver with high probability. However, since the first n packets are sent without coding, the computation cost of network coding can be reduced. In addition, since the first n packets are sent per se, CoFi no longer needs to wait until a full batch of data is available at the transmitter buffer. Consequently, the per-packet delay of CoFi is reduced, and it can gracefully degrade to the non-coding based protocol which guarantees a per-packet quality of service (w.r.t. delay), when the channel condition is satisfactory. Note that such a hybrid coding scheme is not applicable to the existing coding based multipath multihop routing protocols (such as MORE [19]) since there is no direct link between the source and destination in those scenarios.

Note that CoFi has no exponential backoff after losses (this design choice is adopted by 802.11 MAC, which treats all losses as congestion signals, thus resulting in a low efficiency). However, CoFi still preserves the exponential backoff scheme after sensing a busy channel. This is necessary to avoid collisions when both the clients and the AP need to reserve the transmission time.

3.3. Coding Based MAC Layer Caching

A major component of CoFi is the coding based MAC layer caching scheme, as illustrated in Figure 3. The intuition behind MAC caching is to allow each client to opportunistically overhear packets intended for a co-located peer, and pass them later to it. This will further reduce the delay due to bursty losses, which is quite typical in the wireless environment.

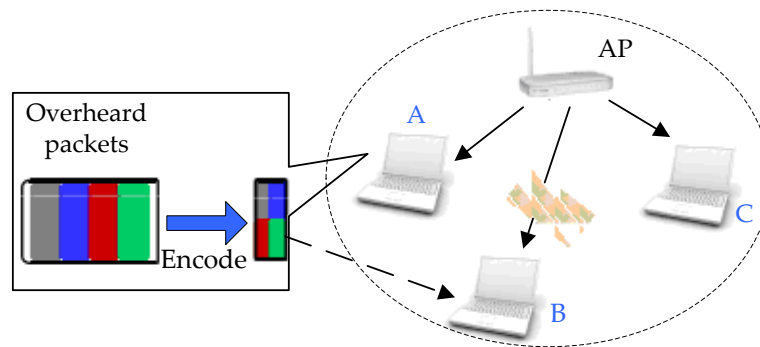


Figure 3. The caching scheme batch transmission and MAC in CoFi.

With random linear network coding, the “packet selection” problem, i.e., choosing which packets to send in the cache, is circumvented. However, two critical issues still need to be addressed in the coding based MAC caching scheme: who should relay and when should the relay help, which correspond to a relay selection problem and relay scheduling problem, respectively.

3.3.1. The Relay Selection Problem

When multiple potential relays are available for help, a relay selection algorithm must be applied, so as to avoid redundant transmissions. In the MORE multipath network coding protocol [19], a credit based algorithm is used that determines the number of coded packets each intermediate forwarder should transmit. The credit is computed at the source node in a centralized manner, based on the average quality of each link. The algorithm is applicable to mesh networks with static nodes and stable links, whose quality do not vary over a long period of time. In contrast, CoFi targets wireless LANs with possible mobile clients and varying link conditions. In such scenarios, measuring the average link quality (time-averaged packet reception rate) of all links online is infeasible due to its large traffic overhead.

More specifically, we advocate a signal strength based on a single-relay selection algorithm. For each client, at most one relay is allowed to help by transmitting cached packets when it is experiencing severe losses. When multiple relays are available, the best is selected according to its potential. The potential of a client R as a relay for client C is defined as:

$$P(R \rightarrow C) = \min\{S(AP \rightarrow R), S(R \rightarrow C)\} \quad (1)$$

where $S(AP \rightarrow R)$ denotes the average signal strength from the AP to node R. The potential of each client C, i.e., $P(C \rightarrow C)$ equals the average signal strength from AP to itself. The selected best relay R, which has the highest potential, will be allowed to offer help for client C if and only if the following two conditions are satisfied:

- $P(R \rightarrow C) - P(C \rightarrow C) > TS$, where TS is a threshold that is used to avoid oscillation of relay selections due to instantaneous variation of channel condition. Currently, we set TS to 10 dB, which is the typical capture threshold used in wireless LANs that indicates a signal significantly overwhelms another. Our experience with the USRP programmable wireless transceiver [28] indicates that the typical variation of signal strength for a static link also lies within 10 dB.
- The packet loss rate of link $(AP \rightarrow C)$ is larger than a threshold TL, which is the PHY level loss rate in a WLAN. This threshold should be set according to the typical QoS requirement of clients. In our experiments, TL is equal to the inverse of the maximum retry limit specified by the 802.11 MAC, i.e., when the PHY loss rate is larger than TL, the 802.11 protocol will suffer from low MAC level packet delivery ratio.

Note that both the above two thresholds are hardcoded to typical values, requiring no runtime intervention.

The advantage of the signal-strength based approach lies in its simplicity and efficiency. Our measurement of wireless links (Section 5) indicates that most links experience a stable signal strength with small variance unless the clients move. Therefore, a few samples are enough to determine the average quality of a link. To obtain the signal strength from R to C, for instance, C only needs to overhear a number of ACKs that are sent from R and are intended for the AP. In contrast, measuring the MAC level packet loss ratio typically requires a large number of samples over a relatively long period of time.

Existing work has mostly discarded signal strength as an indicator of the link quality [29,30]. However, such measurements adopted 802.11 MAC layer throughput as the link quality metric. The 802.11 throughput is not in close correlation with the signal strength because of its persistent retransmission, per-packet ACK, and exponential backoff scheme. In CoFi, a single-direction transmission without backoff dominates the traffic, and the link rate is directly related with the signal strength.

3.3.2. The Relay Scheduling Problem

The scheduling scheme in CoFi triggers the relay to transmit cached packets to the intended client in order to save the channel time wasted on the lossy link between the AP and that client. Once the best relay can decode an entire overheard batch, it will take over the role of the AP, serving the intended client by sending a random linear combination of packets in the batch. When the AP overhears a coded packet from the relay, it stops transmitting to the weak client since it knows the relay has a full batch of data that can rescue the target client.

Beside triggering the relay, the scheduling algorithm must ensure liveness, i.e., it can continue to the next batch after the client can decode the current one. Towards this end, we marked the relay or the client as “FULL” whenever they can decode the current batch. A marked node will broadcast an ACK whenever it receives another packet with the current batch sequence, which implies that the sender is still unaware of the decodability at this node. Such a persistent feedback scheme essentially allows the relay to forward the ACK to the AP, so that it can proceed to a new batch of data.

4. Implementation of CoFi

We have implemented the CoFi protocol based on the 802.11 MAC/PHY layer in ns-2 [31], a widely used packet level discrete event simulator. As mentioned above, we suppressed the original per-packet ACK in 802.11, and used per-batch ACK instead. The 802.11 style per-packet retransmission and the exponential backoff schemes are discarded in CoFi. This modified MAC layer interfaces the higher layers through the CoFi encoding/decoding queues (Figure 4).

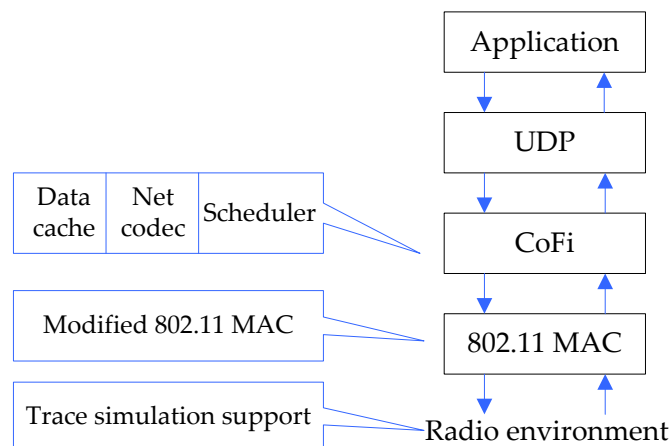


Figure 4. Implementation of CoFi in the ns-2 simulator.

4.1. Network Encoding and Decoding

Although we opted to use a simulation to evaluate the performance of CoFi, real network coding is still needed because it serves as a checker for packet dependence. The implementation is based on C++ and consists of three parts: The coding queue, the encoding/decoding operation, and the elementary matrix operations in the Galois finite field $GF(2^8)$.

A coding queue is simply a finite buffer, whose size equals the batch size. The AP maintains separate coding queues for each client, while each client maintains a coding queue for its own packets, and for the overheard packets intended for other clients. A newly coming packet is put into the coding queue only if it is linearly independent from all existing packets in that queue. Independence checking and decoding operations are performed concurrently, using Gauss–Jordan elimination [27]. The encoding operation is simply a matrix multiplication over the finite field $GF(2^8)$. Elementary arithmetic operations in $GF(2^8)$ are implemented using a widely adopted lookup-table method [32,33].

4.2. Packet Management

The format of a data packet is illustrated in Figure 5. Beside the legacy 802.11 header, a batch sequence is piggybacked, which is used for triggering the persistent ACK and the transmission of a new batch. In addition, before encoding the original data, an offset sequence must be placed into the data field. This is necessary because the decoding algorithm (Gauss–Jordan elimination) may break the original ordering of packets through matrix row swapping. By encoding the offset sequence together with data, we can recover the transport layer sequence of each packet using its batch sequence and MAC layer offset.

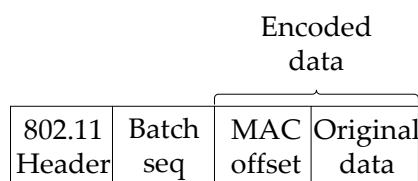


Figure 5. CoFi data packet format.

The ACK packet is similar to the 802.11 specification. Remarkably, the ACK sent from a relay will piggyback the address of the client under its help, because the AP only needs to know who has decoded, rather than who is helping the client.

4.3. Decentralized MAC Caching

The relay scheduling algorithm can be realized without any notification message among client, relay, and the AP. However, the current version of CoFi still relies on message exchanges to carry out the relay selection procedure. More specifically, when a client experiences low throughput due to weak signal strength, it broadcasts a short “HELP” message to neighbors in the WLAN. Each potential relay overhearing this message will reply with the signal strength from the AP to it. The AP collects all reply messages in a timeout period, performs the relay selection, and subsequently broadcast a message containing the identity of the selected relay.

A client needs to re-initiate the relay selection protocol whenever the average signal strength (a moving average) from the relay to it drops by more than TS. This may happen when either of them moves. When the AP → Relay signal strength changes by more than TS, the relay also has to notify the client under help so that a better relay can be selected. Such a relay selection algorithm is feasible in a typical dynamic WLAN since the movement speed of nodes is usually low.

5. Performance and Evaluation

In this section, we evaluate the throughput and delay performance of CoFi, in comparison with the 802.11 protocol. Before delving into the detailed evaluation, we first check the practicality of network coding in terms of computation cost.

5.1. The Computation Cost of Network Coding

Network coding is essentially a mechanism that trades computation time for network throughput. As long as the time spent on encoding or decoding one packet is shorter than that of transmitting one packet (in other words, the coding bandwidth is larger than the effective link bandwidth), the coding operations will not hurt the link throughput, because they can be pipelined with each other and performed in parallel.

To quantify the computation cost of network coding, we test the average bandwidth of $GF(2^8)$ random linear encoding, on a 3.5 GHz desktop PC with 16 GB memory. Figure 6 plots the encoding bandwidth as a function of batch size and block size. Since the encoding always operates over a dense matrix, it tends to dominate the decoding time. Therefore, we focus on encoding bandwidth only. When the batch size is small, the coding bandwidth can be larger than the effective bandwidth of the 802.11 links [34]. Such a result is consistent with existing literature [19]. However, the actual coding bandwidth in our experiment is lower than [19]. This is mainly because our implementation is based on less optimized code. In fact, existing and ongoing work has already exploited fast network encoding/decoding methods, such as hardware accelerated approach [15].

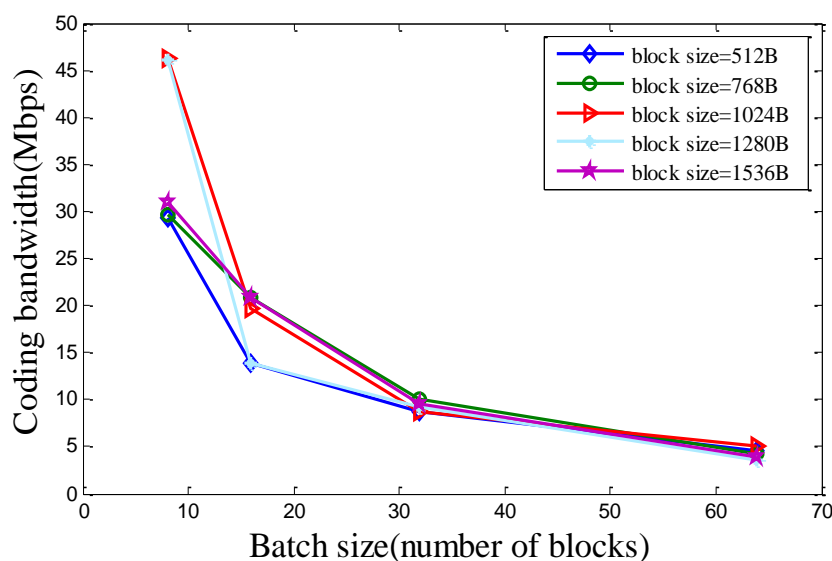


Figure 6. The computation bandwidth of random linear encoding.

Considering the high computing power of existing clients, such as handheld mobile devices and laptops, the computation load of decoding and encoding operations of the clients usually do not affect the bandwidth of MAC protocols. In our experimental evaluation of CoFi, we also assumed the cost of network coding could be ignored.

However, for low-end access points, such as sensors, computation resources may be limited, and the complexity of network coding may make a considerable difference. In our ongoing work, we implement and test network coding over a pair of programmable wireless nodes (the USRP testbed [28]), and dynamically adjust the batch size to satisfy the real-time QoS constraint of the link. We found that when the CPU is heavily loaded with other concurrent tasks, even with batch size 10, and packet size 1.5 KB, the computation cost of network coding is considerable. This work will be complementary to our CoFi protocol.

5.2. The Trace Based Simulation

The variation of wireless channels with space and time is attributed to both large-scale effects (path-loss and shadowing) and small-scale effects (multipath fading and Doppler fading), and only the former is accounted for in the ns-2 channel model. To obtain more realistic results, we adopted a trace-based approach to compensate for the inaccuracy of ns-2 simulation. Specifically, we obtained physical layer packet loss traces using a pair of universal software radio peripheral (USRP) programmable transceiver [28]. Compared with existing 802.11 traces, the advantage of USRP is that it has no MAC protocol, and therefore it can isolate the channel variation from the MAC layer protocol overhead. We modified the ns-2 physical layer module by intercepting all packet transmissions, and then fed the traces into the module, allowing it to determine whether a packet could be successfully received (Figure 7). To improve the granularity of traces, each USRP packet had only 32 bytes, which was shorter than the data packet (1.5 KB) and ACK packet (46 bytes) in our simulation. Whenever the airtime (duration of transmission) of a simulated packet overlapped an erroneous packet in the trace, it was declared as loss.

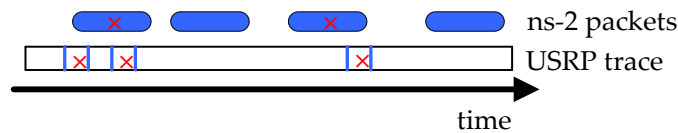


Figure 7. Trace simulation: Feeding USRP packet loss traces into the ns-2 physical module (checked ns-2 packets are declared as losses).

We collected the packet loss traces for all links in the 3-node (AP, A, B) topology in Figure 3, and plot the results in Figure 8. We observed that most links tended to have a stabilized loss rate. However, some links may have experienced a large loss variation and asymmetric loss characteristics. For example, the (A→B) and (B→A) links demonstrate a very different loss rate and variation. Such properties were not modeled in the ns-2 simulator, either.

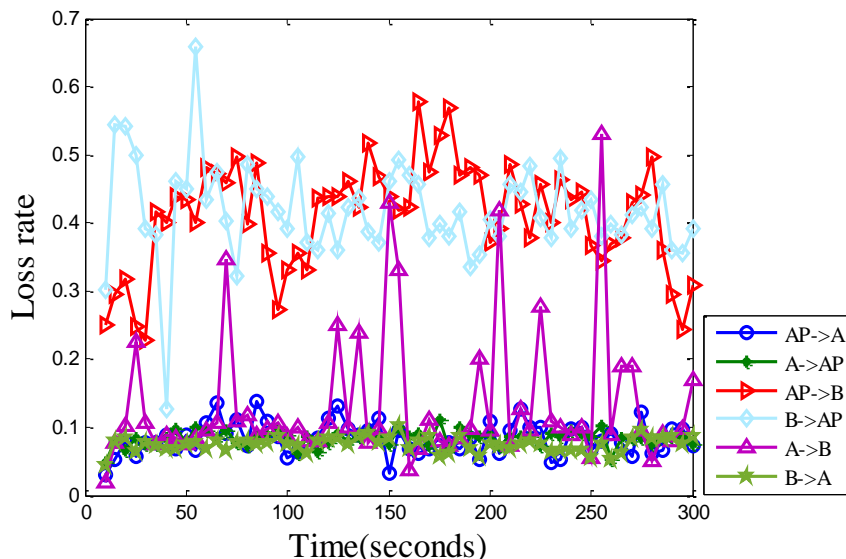


Figure 8. The packet loss rate traces for the 3-node (AP, A, B) topology in Figure 3. Each point represents the average packet loss rate during the past 5 s.

5.3. CoFi Performance in an Elementary Topology

Using the above trace-based simulation, we evaluated the performance of CoFi within the 3-node (AP, A, B) topology in Figure 3. Unless otherwise noted, all MAC/PHY layer parameter settings follow

Table 1. Since the USRP link bandwidth is 1 Mbps, we set the maximum link bandwidth to 1 Mbps in ns-2 simulation, so that the simulation evolved in a similar time scale with the traces.

Table 1. Parameters setting for the performance test.

| Parameter | Value |
|----------------------------------|---|
| Application | UDP with stable data source (data is always available for transmission) |
| CoFi block size | 1 KB |
| CoFi batch size | 8 |
| Data rate | 1 Mbps |
| Basic rate (for control packets) | 1 Mbps |
| 802.11 Max Retry Limit | 7 |
| RX Thresh | 1.52×10^{-11} |
| CS Thresh | 3.08×10^{-11} |
| Transmission power | 0.2818 W |
| Path loss exponent | 4.0 |
| Shadowing deviation | 7.0 |

Our first performance metric is throughput, which is the amount of decodable data received by each client within the unit time. As demonstrated in Figure 9, the clients running CoFi experience much higher throughput than those using 802.11. CoFi provides more than three times improvement of time-averaged throughput for each client and guarantees a full link level reliability for each client. In contrast, the 802.11 MAC discards a packet if it cannot be acknowledged after a number of retransmission attempts (equal to Max Retry Limit). In addition, 802.11 suffers from link asymmetry. If the lossy link is on the forward direction, it affects 802.11 more than CoFi, because of the persistent retransmission and exponential backoff mechanisms. On the other hand, since CoFi uses batch transmission which requires fewer ACKs, it suffers less from ACK losses on the reverse direction.

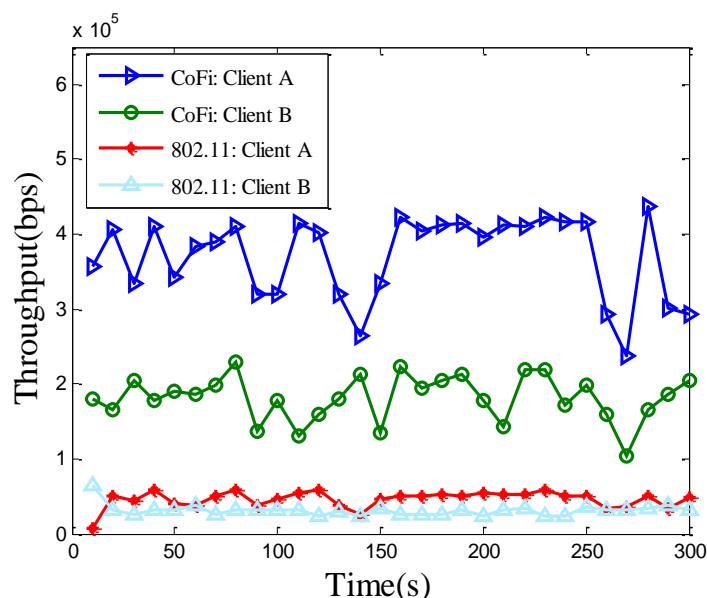


Figure 9. The throughput (averaged over 10-s periods) of each client running 802.11 protocol and CoFi protocol in the elementary topology.

Figure 10 evaluates the time-averaged per-batch delay of CoFi and 802.11. For CoFi, the per-batch delay was measured without considering the packet reliability, i.e., it equaled each time duration when the receiver collected n different packets (n is the batch size). For 802.11, client A needed to strictly receive a packet one by one according to the order in which the packets were sent. At the same time, the

transmitter needed to get full ACKs (equaling the number of packets). This indicated the probability that the data transmission was interfered by the changes of the channel environment being increased. Since the per-batch delay was inversely proportional to the throughput, the large performance gap between these two protocols was consistent with the throughput comparison in Figure 9.

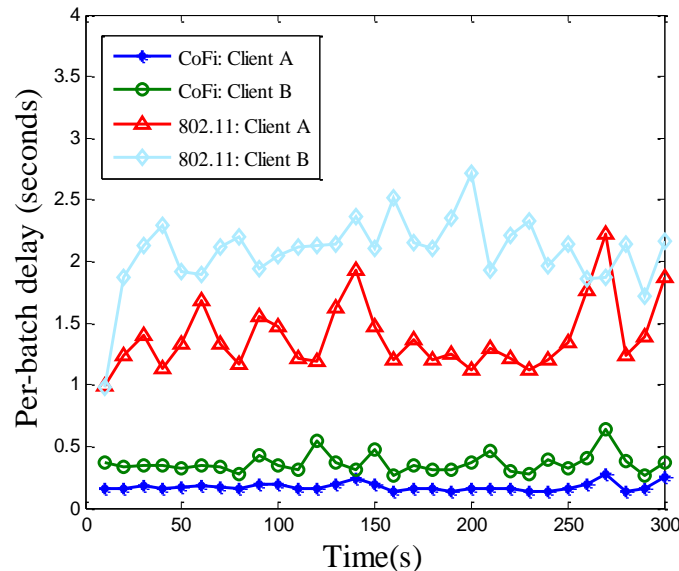


Figure 10. The per-batch delay (averaged over 10-s periods) of each client running 802.11 protocol and CoFi protocol in the elementary topology.

We proceed to evaluate the per-packet delay of both protocols. In CoFi an encoded packet is not visible at the receiver side until the entire batch it belongs to is decodable. Therefore, the per-packet delay of CoFi equals its per-batch delay if a packet reception fails during the hybrid coding stage. From Figure 11, we see that the per-packet delay of client A (which experiences low loss rate) when using CoFi is similar to the case when running 802.11. This is partly because of its hybrid coding scheme, which allows it to gracefully degrade to a non-coding protocol. In addition, since CoFi discards the persistent retransmission and exponential backoff scheme, the client B that is under a harsh channel condition does not affect the throughput of client A who experiences much less packet losses. For client B, the per-packet delay of CoFi is higher than that of 802.11, since the successful reception of an entire batch takes longer than receiving a single packet in 802.11. We noted that using a smaller batch size does not necessarily result in shorter delay, especially for lossy links. This is because a smaller batch size also entailed more interaction overhead, such as the ACK feedback and the switch between the AP and relay. In the extreme case when batch size equals 1, CoFi was essentially degraded into a persistent retransmission protocol, and the benefit of MAC caching also diminished.

Another problem of interest is: To what extent does the MAC caching scheme help? Figures 12 and 13 compare the two versions of CoFi with and without MAC caching, in terms of throughput and per-packet delay.

Since the MAC caching scheme ensures that the throughput of high-quality links are not affected when they offer help to other links, client A achieves the same level of throughput and delay, no matter if MAC caching is used. However, the throughput of client B is nearly doubled with MAC caching. The reason is that MAC caching replaces the low quality link ($AP \rightarrow B$) with the high-quality link ($A \rightarrow B$) when relay A can offer help. The packets received by client A are recoded and sent to Client B to improve the transmission efficiency. From Figure 13, we also observe that MAC caching is especially helpful during the period when the ($AP \rightarrow B$) link experiences harsh channel condition, where the bursty losses can be smoothed out. Therefore, caching is particularly useful for jitter-sensitive applications, such as video streaming over the wireless LAN.

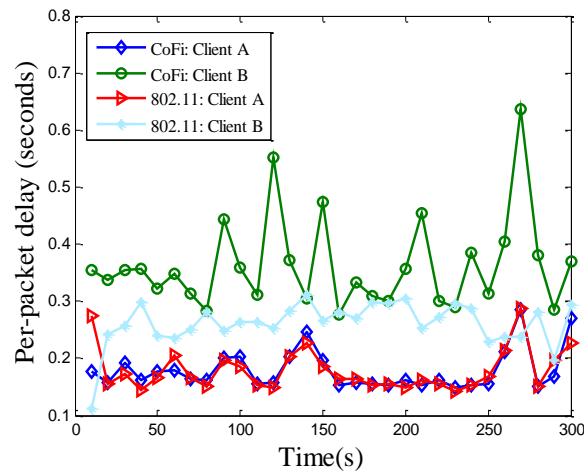


Figure 11. The per-packet delay (averaged over 10-s periods) of each client running 802.11 protocol and CoFi protocol in the elementary topology.

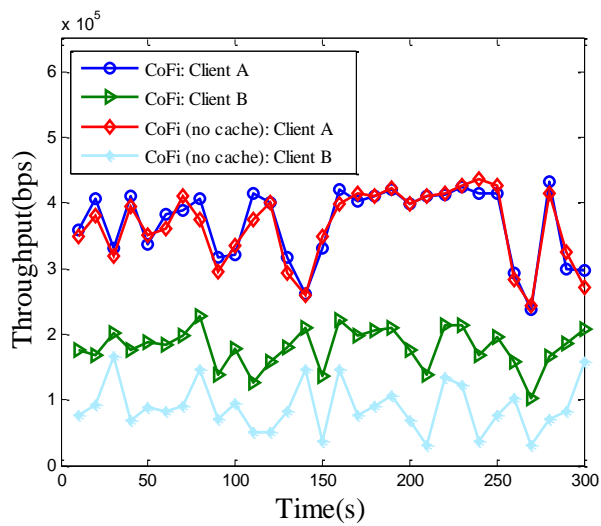


Figure 12. The throughput of each client running CoFi with/without MAC caching in the elementary topology.

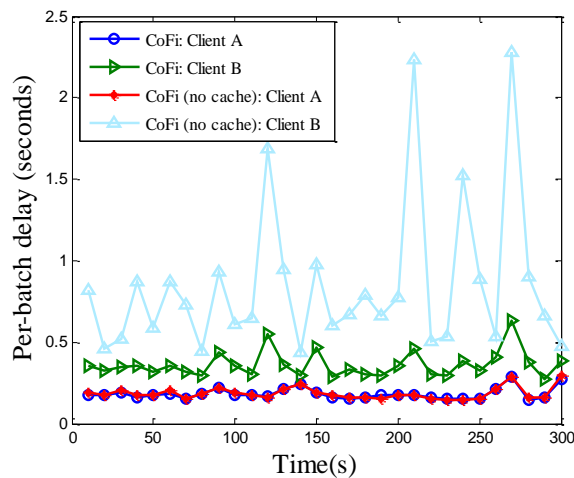


Figure 13. The per-packet delay of each client running CoFi with/without MAC caching in the elementary topology.

5.4. CoFi in a Dynamic Topology Setting

To further validate the design of CoFi, we tested it in a simulated dynamic topology with one AP and five mobile clients. Each client moved according to the random waypoint model with speed 2 m/s and sojourn time uniformly distributed within 10 and 30 s. The terrain was a 50 m-radius circular region, and the average reception probability at the edge of the region was 0.5. The maximum data rate of each link was set to 2 Mb/s, i.e., the basic data rate of 802.11.

Figures 14 and 15 plot the total network throughput and per-packet delay performance, respectively. Owing to its batch transmission, CoFi with or without MAC caching mechanisms improved the throughput of 802.11 by a significant margin, and the amplitude elevated was slightly superior to the value in the elementary topology. Furthermore, with the use of MAC caching scheme, the throughput was opportunistically improved, according to whether a relay was available nearby. It can be expected that in a dense network with many clients, CoFi will demonstrate even higher throughput gains due to more available relays. Another observation was that the per-packet delay of CoFi could be lower than the 802.11 protocol (Figure 15). This is because when more mobile clients are subscribed to the same AP, the probability that one of them experiences low channel quality becomes higher. Clients with low channel quality could encumber the ones with high channel quality. This will lead to transmission time increment in the 802.11 protocol. In addition, comparing with the elementary topology, we also observed that the time of per-packet delay increased and the volatility of value is quite large. This indicated that the dynamic topology needs higher level protocol and it is much more complex to be implemented. The results show that the volatility in running CoFi was much smaller than that in the 802.11 protocol, which means the present method is more suitable for application in a practical scenario. Therefore, the case where the low-quality link brings down the performance of the entire network becomes the common case.

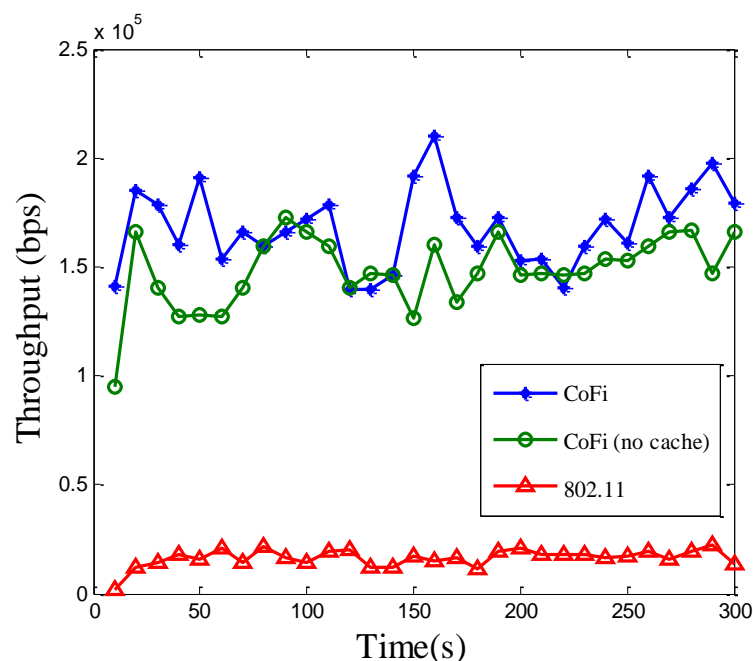


Figure 14. The total throughput of the network running 802.11 protocol and CoFi protocol with/without MAC caching in a dynamic topology.

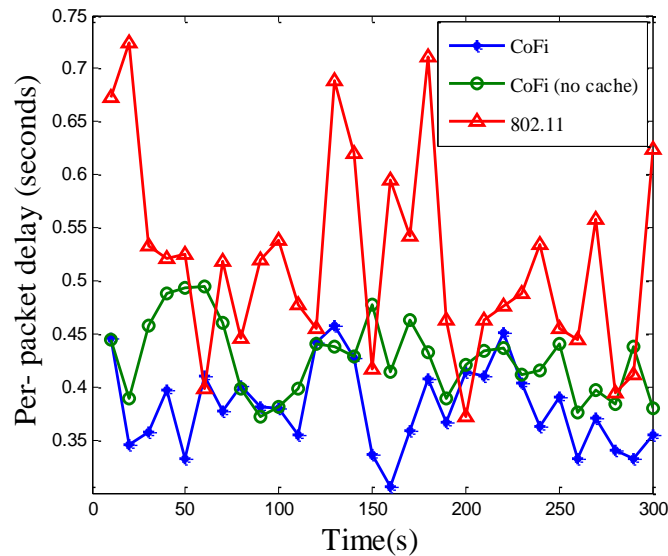


Figure 15. The per-packet delay of the network running 802.11 protocol and CoFi protocol with/without MAC caching in a dynamic topology.

5.5. Potential Application

The above evaluation demonstrated that CoFi could provide a much higher network throughput than the widely adopted 802.11 MAC protocol, especially when one or more clients in the WLAN are experiencing harsh channel conditions. However, CoFi does not guarantee low per-packet delay for all cases. Therefore, it is best applicable to bulk file transmissions and streaming protocols, which adopts buffers to tolerate an initial setup delay. In addition, when the custom VLSI of RLNC is implemented and the computing power of CPU improved, it is possible to apply random linear coding on handheld mobile devices and standalone wireless nodes [35]. There can be a wide variety of form for clients, such as the sensor nodes in IoTs, which greatly expands its range of application.

6. Reverse Engineering of CoFi and 802.11

In this section, we developed a simple model for CoFi and the 802.11 MAC, in order to further understand the root of their performance disparity. We first modeled the asymptotic throughput of CoFi and 802.11, and then proved that they essentially solve optimization problems with the objective of maximizing throughput, but with different fairness constraint.

6.1. Asymptotic Throughput of CoFi and 802.11

We consider a wireless LAN topology with one AP and K clients, denoted as C_1, C_2, \dots, C_K . The average reception rate of each link ($AP \rightarrow C_i$) equals p_i . The maximum link bandwidth supported by the underlying PHY is R . We assume each client requests a large file from the AP and the AP serves all clients in a round-robin manner.

For the 802.11 MAC, we assume the AP adopts persistent retransmission without backoff, thus analyzing an asymptotic upper bound of its throughput. For each client C_i , the average number of transmissions needed to deliver a unit packet is $x_i = 1/p_i$. Since the airtime of each packet is constant, the interval that AP serves C_i is:

$$f_i = \frac{1}{\sum_{j=1}^K x_j} \quad (2)$$

which is independent of the client's identity. Within this interval, the total units of packets that are successfully delivered is K . Therefore, the total network throughput is:

$$T = \frac{K}{\sum_{j=1}^K x_j} \quad (3)$$

Since each client only receives one packet intended for it during each interval, its effective throughput is:

$$T_i = R \cdot \frac{T}{K} = \frac{R}{\sum_{j=1}^K x_j} \quad (4)$$

which depends on the link quality of all other clients.

For CoFi without caching, in each round a client is served only once. Therefore, its effective throughput only depends its own link quality. Specifically, the throughput of a client C_i is:

$$T'_i = (R/K) \cdot p_i \quad (5)$$

6.2. Performance from a Fairness Point of View

In the literature of network resource allocation [36], two notions of fairness are widely adopted: Max-min fairness and proportional fairness. Consider the MAC as a bandwidth (or channel time) resource allocation protocol, then the former fairness measure essentially strives to allocate more channel time to the weak link [37,38], while the latter allocates the link bandwidth according to the quality of each link. More specifically, with the maxmin fairness measure, a MAC protocol with persistent retransmission solves the following optimization problem:

$$\max t \quad (6)$$

$$\text{subject to : } t \leq T_i, \forall 1 \leq i \leq K \quad (7)$$

$$\sum_{j=1}^K \frac{T_j}{p_j} \leq R \quad (8)$$

where t is the effective throughput of the client with the weakest link condition. It can be easily proved that $t \leq R \cdot (T/K) = R/\sum_{j=1}^K x_j$, which is achievable when $\forall i, T_i = R \cdot (T/K) = R/\sum_{j=1}^K x_j$. This is exactly the asymptotic throughput provisioned by the 802.11 protocol. Therefore, we have the following observation:

Theorem 1. *802.11 MAC essentially solves an optimization problem that achieves maxmin throughput fairness.*

It is well known that maxmin fairness strives to allocate the same level of resources to all clients, thus resulting in low efficiency. This reveals the fundamental reason behind the low performance of 802.11 under unsatisfactory channel conditions.

In contrast, with the proportional fairness measure, a MAC protocol without persistent retransmission solves the following optimization problem:

$$\max \sum_{j=1}^K \ln(F_j) \quad (9)$$

$$\text{subject to : } F_i \leq T'_i = \frac{R}{K} \cdot p_i \quad (10)$$

where F_j is the effective throughput that we can really get. Similar to the 802.11 case, we can solve this problem by upper-bounding the objective. The solution $(R/K) \cdot p_i$ is exactly the asymptotic throughput of CoFi without MAC caching. Therefore, we have:

Theorem 2. *CoFi without MAC caching essentially solves an optimization problem that achieves proportional throughput, fairness.*

In other words, high-quality links will enjoy a high throughput, while low-quality links are not starved. Such a proportional throughput measure may result in an imbalanced throughput allocation when the link qualities are different. The CoFi protocol with MAC caching offers a way to balance the throughput by replacing the low quality (AP→client) link with a high quality (relay→client) link. Therefore, CoFi essentially strikes a balance between the maxmin and proportional fairness measures.

Through the above analysis, it can be seen that 802.11 and CoFi are essentially distributed approximate solutions to certain optimization problems, but the optimization objectives are different. The network source allocation problems are also equivalent to a discrete combinatorial optimization problem under certain constraints, which may be considered to solve by stochastic optimization algorithms [39–41].

7. Conclusions

In this paper, we introduced CoFi, a MAC protocol that improves Wi-Fi performance under harsh channel conditions. To mask packet losses while ensuring reliable delivery, CoFi groups packets into batches, and continuously transmits random linear combination of all data in each batch, until they can be decoded by the receiver. In addition, CoFi allows clients to cache overheard packets, and schedules the best relay to help the client with weak link quality to the access point. Using trace based simulation, we find that CoFi can significantly improve throughput in a lossy network, and gracefully degrade to a non-coding protocol when the channel condition is good. The per-packet delay of CoFi is comparable to 802.11 in a lossy network, and even better in a dynamic topology with mobile clients. We also developed a simple analytical model that justifies the performance of CoFi from a fairness point of view. Our ongoing work tries to improve CoFi from the following aspects:

- Implementation of CoFi on real wireless nodes. We plan to implement CoFi on the USRP testbed, which will provide more accurate experimentation results. We are also considering making CoFi more compatible with existing 802.11 features, such as rate adaptation and service differentiating.
- CoFi for elastic applications. In this paper, we assumed stable application data when evaluating CoFi and 802.11. Elastic traffic, such as VBR, has more stringent requirements on delay performance. To adapt to such applications, we have to allow a dynamic batch size, so that each packet can be delivered on time. While adjusting the batch size, we also need to consider the end-host delay of network coding, especially when the computation load of coding operations is non-negligible.

Author Contributions: N.M. contributed to derive the results and write the paper. M.D. proposed the idea and supervised the work. He also revised the article.

Funding: The work described in this paper was partially supported by the National Natural Science Foundation of China under Grant (No. 51474100) and the Natural Science Foundation of Heilongjiang Province under Grant (No. QC2016093).

Acknowledgments: The authors would like to express their appreciation of associate professor X.Y. Zhang and Z.K. Wang for their valuable suggestions. The authors would also like to thank the anonymous reviewers for their constructive comments.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yao, Y.; Sheng, B.; Mi, N.F. A new packet scheduling algorithm for access points in crowded WLANs. *Ad. Hoc. Netw.* **2015**, *36*, 100–110. [[CrossRef](#)]
2. Mahanti, A.; Carlsson, N.; Williamson, C.; Arlitt, M. Ambient Interference Effects in Wi-Fi Networks. In Proceedings of the 9th IFIP TC 6 International Conference on Networking, Chennai, India, 11–15 May 2010.

3. Khan, M.Y.; Veitch, D. Peeling the 802.11 Onion: Separating Congestion from Physical PER. In Proceedings of the 3th ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization, San Francisco, CA, USA, 19 September 2008.
4. Masri, A.E.; Sardouk, A.; Khoukhi, L.; Hafid, A.; Gaiti, D. Neighborhood-Aware and Overhead-Free Congestion Control for IEEE 802.11 Wireless Mesh Networks. *IEEE Trans. Wirel. Commun.* **2014**, *13*, 5878–5892. [[CrossRef](#)]
5. Yu, C.; Kang, G.S.; Song, L. Link-layer Salvaging for Making Routing Progress in Mobile Ad Hoc Networks. In Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc), Urbana-Champaign, IL, USA, 25–27 May 2005.
6. Dogar, F.; Phanishayee, A.; Pucha, H.; Ruwase, O.; Andersen, D. Ditto: A System for Opportunistic Caching in Multi-hop Wireless Mesh Networks. In Proceedings of the 14th ACM International Conference on Mobile Computing and Networking, San Francisco, CA, USA, 14–19 September 2008.
7. Karaca, M.; Zhang, Z.; Landfeldt, B. An Opportunistic Backoff Scheme for Dense IEEE 802.11 WLANs. In Proceedings of the Globecom Workshops (GC Wkshps), San Diego, CA, USA, 6–10 December 2015; pp. 10–26.
8. Shetty, P.; Borkar, V.; Kasbekar, G. An Adaptive Window Scheme for Backoff in 802.11 MAC Protocol. In Proceedings of the 9th EAI International Conference on Performance Evaluation Methodologies and Tools, Berlin, Germany, 14–16 December 2015.
9. Swapna, B.T.; Eryilmaz, A.; Shroff, N.B. Throughput-Delay Analysis of Random Linear Network Coding for Wireless Broadcasting. *IEEE Trans. Inf. Theory* **2013**, *59*, 6328–6341. [[CrossRef](#)]
10. Pandi, S.; Gabriel, F.; Cabrera, J.A. PACE: Redundancy Engineering in RLNC for Low-Latency Communication. *IEEE Access* **2017**, *5*, 20477–20493. [[CrossRef](#)]
11. Wunderlich, S.; Gabriel, F.; Pandi, S. Caterpillar RLNC (CRLNC): A Practical Finite Sliding Window RLNC Approach. *IEEE Access* **2017**, 20183–20197. [[CrossRef](#)]
12. Joshi, T.; Mukherjee, A.; Yoo, Y.; Agrawal, D. Airtime Fairness for IEEE 802.11 Multirate Networks. *IEEE Trans. Mob. Comput.* **2008**, *7*, 513–527. [[CrossRef](#)]
13. Basu, P.; Chau, C.K.; Bejan, A.L.; Johnson, M.P. Efficient Multicast in Hybrid Wireless Networks. In Proceedings of the 34th Anniversary of the Premier International Conference for Military Communications (MILCOM), Tampa, FL, USA, 26–28 October 2015.
14. Chatzigeorgiou, I.; Tassi, A. Decoding Delay Performance of Random Linear Network Coding for Broadcast. *IEEE Trans. Veh. Technol.* **2017**, *66*, 7050–7060. [[CrossRef](#)]
15. Shojania, H.; Li, B. Parallelized Progressive Network Coding with Hardware Acceleration. In Proceedings of the 15th IEEE International Workshop on Quality of Service, Evanston, IL, USA, 21–22 June 2007.
16. Li, Z.; Li, B.; Jiang, D.; Lau, L.C. On Achieving Optimal Throughput with Network Coding. In Proceedings of the 24th IEEE Annual Joint Conference on Computer and Communications Societies (INFOCOM), Miami, FL, USA, 13–17 March 2005.
17. Ho, T.; Medard, M.; Shi, J.; Effros, M.; Karger, D. On Randomized Network Coding. In Proceedings of the 41st Annual Allerton Conference on Communication Control and Computing, Monticello, VA, USA, 26–29 October 2003.
18. Katti, S.; Rahul, H.; Hu, W.; Katabi, D.; Medard, M.; Crowcroft, J. XORs in the Air: Practical Wireless Network Coding. *IEEE/ACM Trans. Netw.* **2008**, *16*, 497–510. [[CrossRef](#)]
19. Chachulski, S.; Jennings, M.; Katti, S.; Katabi, D. Trading Structure for Randomness in Wireless Opportunistic Routing. In Proceedings of the ACM Conference on Applications Technologies Architectures and Protocols for Computer Communications, Kyoto, Japan, 27–31 August 2007.
20. Lin, Y.; Li, B.; Liang, B. Codeor: Opportunistic routing in wireless mesh networks with segmented network coding. In Proceedings of the 2008 IEEE International Conference on Network Protocols, Orlando, FL, USA, 19–22 October 2008.
21. Kim, R.Y.; Jin, J.; Li, B. Scattered Random Network Coding for Efficient Transmission in Multihop Wireless Networks. *IEEE Trans. Veh. Technol.* **2011**, *6*, 2383–2389. [[CrossRef](#)]
22. Amerimehr, M.H.; Ashtiani, F.; Valaee, S. Maximum Stable Throughput of Network-Coded Multiple Broadcast Sessions for Wireless Tandem Random Access Networks. *IEEE Trans. Mob. Comput.* **2014**, *13*, 1256–1267. [[CrossRef](#)]

23. Eryilmaz, A.; Ozdaglar, A.; Medard, M. On Delay Performance Gains from Network Coding. In Proceedings of the 40th IEEE Annual Conference on Information Sciences and Systems, Princeton, NJ, USA, 22–24 March 2006.
24. Jin, J.; Li, B.; Kong, T. Is Random Network Coding Helpful in WiMAX? In Proceedings of the 27th IEEE Conference on Computer Communications (INFOCOM), Phoenix, AZ, USA, 13–18 April 2008.
25. Lin, S.J.; Fu, L.Q. Throughput Capacity of IEEE 802.11 Many-to/From-One Bidirectional Networks with Physical-Layer Network Coding. *IEEE Trans. Wirel. Commun.* **2016**, *15*, 217–231. [[CrossRef](#)]
26. Chou, P.; Wu, Y.; Jain, K. Practical Network Coding. In Proceedings of the 41st Annual Allerton Conference on Communication Control and Computing, Monticello, VA, USA, 26–29 October 2003.
27. Zhang, X.; Li, B. Optimized Multipath Network Coding in Lossy Wireless Networks. In Proceedings of the 28th International Conference on Distributed Computing Systems, Beijing, China, 17–20 June 2008.
28. Ettus Research. The Universal Software Radio Peripheral. Available online: <http://www.ettus.com/> (accessed on 22 July 2017).
29. Reis, C.; Mahajan, R.; Rodrig, M.; Wetherall, D.; Zahorjan, J. Measurement-based Models of Delivery and Interference in Static Wireless Networks. In Proceedings of the ACM Conference on Application Technologies Architectures and Protocols for Computer Communications, Pisa, Italy, 11–15 September 2006.
30. Camp, J.; Robinson, J.; Steger, C.; Knightly, E. Measurement Driven Deployment of a Two-tier Urban Mesh Access Network. In Proceedings of the 4th International Conference on Mobile System Application and Services, Uppsala, Sweden, 19–22 June 2006.
31. Project, T.V. The Network Simulator-ns-2. Available online: <http://www.isi.edu/nsnam/ns> (accessed on 5 November 2017).
32. Wagner, N.R. The Laws of Cryptography with Java Code. Available online: <http://www.cs.utsa.edu/~wagner/laws/FFM.html> (accessed on 6 November 2017).
33. Sundararajan, J.K.; Shah, D.; Medard, M.; Jakubczak, S.; Mitzenmacher, M.; Barros, J. Network Coding Meets TCP: Theory and Implementation. *Proc. IEEE* **2011**, *99*, 490–512. [[CrossRef](#)]
34. IEEE 802.11. Available online: <http://en.wikipedia.org/wild/802.11> (accessed on 30 October 2016).
35. Angelopoulos, G.; Paidimarri, A.; Medard, M.; Anantha, P.C. A Random Linear Network Coding Accelerator in a 2.4GHz Transmitter for IoT Applications. *IEEE Trans. Circuits Syst.* **2017**, 2582–2590. [[CrossRef](#)]
36. Kelly, F.; Maulloo, A.; Tan, D. Rate Control for Communication Networks: Shadow Prices, Proportional Fairness and Stability. *J. Oper. Res. Soc.* **1998**, *49*, 237–252. [[CrossRef](#)]
37. Shah, S.H.; Nahrstedt, K. Price-based Channel Time Allocation in Wireless LANs. In Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, Tokyo, Japan, 23–24 March 2004; pp. 1–7.
38. Douglas, J.; Leith, Q.; Cao, V.; Subramanian, G. Max-Min Fairness in 802.11 Mesh Network. *IEEE/ACM Trans. Netw.* **2012**, *20*, 756–769.
39. Schumer, M.A.; Steiglitz, K. Adaptive step size random search. *IEEE Trans. Autom. Control* **1968**, *13*, 270–276. [[CrossRef](#)]
40. Kirkpatrick, S.; Gelatt, C.D., Jr.; Vecchi, M.P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)] [[PubMed](#)]
41. Martino, L.; Elvira, V.; Luengo, D.; Corander, J.; Louzada, F. Orthogonal Parallel MCMC Methods for Sampling and Optimization. *Digit. Signal Process.* **2016**, *58*, 64–84. [[CrossRef](#)]

