Device-to-Device Networking Meets Cellular via Network Coding

Yasaman Keshtkarjahromi\textsuperscript{a}, Student Member, IEEE, Hulya Seferoglu\textsuperscript{b}, Member, IEEE, Rashid Ansari, Fellow, IEEE, and Ashfaq Khokhar, Fellow, IEEE

Abstract—Utilizing device-to-device (D2D) connections among mobile devices is promising to meet the increasing throughput demand over cellular links. In particular, when mobile devices are in close proximity of each other and are interested in the same content, D2D connections such as Wi-Fi Direct can be opportunistically used to construct a cooperative (and jointly operating) cellular and D2D networking system. However, it is crucial to understand, quantify, and exploit the potential of network coding for cooperating mobile devices in the joint cellular and D2D setup. In this paper, we consider this problem, and: 1) develop a network coding framework, namely NCMI, for cooperative mobile devices in the joint cellular and D2D setup, where cellular and D2D link capacities are the same; and 2) characterize the performance of the proposed network coding framework, where we use packet completion time, which is the number of transmission slots to recover all packets, as a performance metric. We demonstrate the benefits of our network coding framework through simulations.

Index Terms—Network coding, cooperative mobile devices, device-to-device (D2D) networking.

I. INTRODUCTION

The increasing popularity of diverse applications in today’s mobile devices introduces higher demand for throughput, and puts a strain on network resources especially on cellular links. In fact, cellular traffic is growing exponentially, and it is expected to remain so for the foreseeable future [1], [2].

The default operation in today’s networks is to connect each mobile device to the Internet via its cellular or Wi-Fi connection, Fig. 1(a). On the other hand, utilizing device-to-device (D2D) connections simultaneously with the cellular connections is promising to meet the increasing throughput demand [3]–[10]. In particular, when mobile devices are in the close proximity of each other and are interested in the same content, D2D connections such as Wi-Fi Direct can be opportunistically used to construct a cooperative (and jointly operating) cellular and D2D networking system, Fig. 1(b).

In this paper, our goal is to understand the potential of the system when D2D networking meets cellular, and develop a network coding framework to exploit this potential. We consider a scenario that a group of mobile devices are within one another’s transmission range and thus can hear each other. These cooperative mobile devices, are interested in the same content, e.g., video, exploit both cellular and D2D connections. In this setup, a common content is broadcast over cellular links,\textsuperscript{1} Fig. 2(a). However, mobile devices may receive only a partial content due to packet losses over cellular links, Fig. 2(b). The remaining missing content can then be recovered by utilizing both cellular and D2D connections simultaneously in a cooperative manner. In this setup, thanks to utilizing different parts of the spectrum, cellular links and D2D connections (namely Wi-Fi Direct) operate concurrently. Thus, a mobile device can receive two packets simultaneously; one via cellular, and the other via D2D connections. The fundamental question in this context, and the focus of this paper, is to design and analyze efficient network coding algorithms that take into account (i) concurrent operation of cellular and D2D connections, and (ii) the cooperative nature of mobile devices.

The performance of network coding in cellular-only and D2D-only systems has been considered in previous work [12], [13]. This paper fills the gap of understanding the benefits of utilizing both cellular and D2D links to achieve performance gains.

\textsuperscript{1}Note that broadcasting over cellular links is part of LTE [13]–[15], and getting increasing interest in practice, so we consider broadcast scenario instead of unicast.
work, [12], [16]–[22], in the context of broadcasting a common content over cellular links, and repairing the missing content via retransmissions over cellular links, or by exploiting D2D connections. The following example demonstrates the potential of network coding in cellular- or D2D-only systems.

Example 1 (Cellular-Only Setup): Let us consider Fig. 2(a), where four packets, $p_1, p_2, p_3, p_4$, are broadcast from the base station. Assume that after the broadcast, $p_1$ is missing at mobile device $A$, $p_2$ is missing at $B$, and $p_3$ and $p_4$ are missing at $C$, Fig. 2(b). The missing packets can be recovered via re-transmissions (broadcasts) in a cellular-only setup (D2D connections are not used for recovery). Without network coding, four transmissions are required so that each mobile device receives all the packets. With network coding, two transmissions from the base station are sufficient: $p_1 + p_2 + p_3$ and $p_4$. After these two transmissions, all mobile devices have the complete set of packets. As seen, network coding reduces four transmissions to two, which shows the benefit of network coding in cellular-only setup.

D2D-Only Setup: Now let us consider packet recovery by exploiting only D2D connections (cellular connections are not used for recovery). Assume that $p_1$ is missing at mobile device $A$, $p_2$ is missing at $B$, and $p_3$ and $p_4$ are missing at $C$. Without network coding, four transmissions are required to recover all missing packets in all mobile devices. With network coding by exploiting D2D connections, two transmissions are sufficient: (i) mobile device $B$ broadcasts $p_1 + p_3$, and (ii) $A$ broadcasts $p_2 + p_4$. After these two transmissions, all mobile devices have all the packets. In this example, by taking advantage of network coding, the number of transmissions are reduced from four to two transmissions.

The above example demonstrates the benefit of network coding in cellular-only and D2D-only setups. Yet, mobile devices are not limited to operate in cellular-only and D2D-only setups. Indeed, mobile devices can exploit multiple interfaces simultaneously including cellular and D2D connections. The following example demonstrates the potential of network coding in the joint cellular and D2D setup.

Example 2 (Joint Cellular and D2D Setup): Let us consider Fig. 2(b) again, and assume that after the broadcast, $p_1$ is missing at $A$, $p_2$ is missing at $B$, and $p_3$ and $p_4$ are missing at $C$. In order to recover the missing packets, we exploit both cellular and D2D links; each mobile device can receive two simultaneous packets, one of which is transmitted through the cellular links from the source and the other is transmitted through a D2D link from one of the mobile devices. We assume that cellular and D2D links have the same capacity. For this example, the following transmissions are simultaneously made to recover the missing packets: (i) the base station broadcasts $p_1 + p_3$ via cellular links, and (ii) mobile device $A$ broadcasts $p_2 + p_4$ via D2D links. The number of transmission slots is thus reduced to one from two as compared to Example 1.

As seen, mobile devices that use their cellular and D2D connections simultaneously and cooperatively have a potential of improving throughput significantly. However, it is crucial to understand and quantify the potential of network coding for cooperating mobile devices in joint cellular and D2D setup. In this paper, we consider this problem, and (i) develop a network coding framework, called network coding for multiple interfaces (NCMI) with jointly operating cellular and D2D interfaces for cooperative mobile devices, where cellular and D2D link capacities are the same, and (ii) characterize the performance of the proposed network coding framework, where we use packet completion time, which is the number of transmission slots to recover all packets, as a performance metric. The following are the key contributions of this work:

- We propose a network coding algorithm; NCMI-Batch, where packets are network coded as a batch (of packets) to improve the throughput of cooperative mobile devices in a joint cellular and D2D setup. By taking into account the number of packets that each mobile device would like to receive for packet recovery, we develop an upper bound on the packet completion time of NCMI-Batch.
- For the same joint cellular and D2D setup, we develop a network coding algorithm; NCMI-Instant, where packets are network coded in a way that they can be decoded immediately after they are received. NCMI-Instant is crucial for applications with deadline constraints. Furthermore, we characterize the performance of NCMI-Instant, and develop an upper bound on its packet completion time.
- We develop a lower bound on the packet completion time when any network coding algorithm is employed in the joint cellular and D2D setup.
- We evaluate NCMI-Batch and NCMI-Instant for different numbers of devices and packets, and loss probabilities. The simulation results show that our algorithms significantly improve packet completion time as compared to baselines, and the upper bounds we developed for NCMI-Batch and NCMI-Instant are tight.

The structure of the rest of this paper is as follows. Section II presents preliminaries and our problem statement. Section III presents our proposed network coding algorithm, and provides our upper bound analysis. Section IV presents the lower bound on the performance of any network coding algorithm. Section V presents simulation results. Section VI presents related work. Section VII concludes the paper.

II. Preliminaries & Problem Statement

We consider a setup with $N$ cooperative mobile devices. Let $N$ denotes the set of devices in our system with $N = |N|$. These devices are within close proximity to each other, so they are within the transmission range of one another. The mobile devices in $N$ are interested in receiving packets $p_m$ from set $\mathcal{M}$, i.e., $p_m \in \mathcal{M}$ and $M = |\mathcal{M}|$.

The mobile devices in our system model are able to use cellular and D2D interfaces simultaneously to receive data. In particular, we consider a two-stage model for our joint cellular and D2D setup. In the first stage, all packets are broadcast to all devices via cellular links, while in the second stage, missing packets are recovered by utilizing cellular and D2D links jointly. This two-stage model fits well for error correction, because the cellular-only operation in the first stage saves energy (as keeping multiple interfaces open increases energy consumption), and the joint cellular and D2D operation in the second stage helps quickly recover missing packets while relieving the load on cellular links. In our setup, D2D links are only needed if some packets are lost over the cellular broadcast link. Thus, at time slots when there are no packet losses, which is common scenario for practical loss rates, D2D interfaces of devices would remain idle if we keep both cellular and D2D interfaces open in the first stage. However, idle interfaces, i.e., even if no packet is transmitted or received, still consume energy [23], [24]. Thus, our two-stage model
is a good approach for mobile devices operating on batteries. Next, we further explain the operation of our two-stage model.

In the first stage, all packets are broadcast to all devices via cellular links. Mobile devices may receive partial content due to packet losses over the cellular broadcast link. Thus, after the first stage, the set of packets that mobile device \( n \in \mathcal{N} \) has successfully received is \( \mathcal{H}_n \), and is referred to as \( \text{Has} \) set of device \( n \). The set of packets that is lost in the first stage at mobile device \( n \) is referred to as \( \text{Wants} \) set of device \( n \) and denoted by \( \mathcal{W}_n \). Furthermore, we define the set \( \mathcal{M} \) as \( \mathcal{M} = \bigcap_{n \in \mathcal{N}} \mathcal{W}_n \). Note that the packets in \( \mathcal{M} \) are not received by any devices during the first stage.

In the second stage, missing packets are recovered by utilizing cellular and D2D links jointly. In particular, a mobile device may receive two recovery packets; one from cellular link and another from D2D link, simultaneously. Exploiting joint cellular and D2D links has the potential of improving throughput [10]. In order to use the available resources more efficiently, we need to determine the best possible network coded packets to be transmitted over cellular and D2D links at each transmission slot. This is an open problem and the focus of this paper.

In particular, in this paper, we develop a network coding framework for multiple interfaces (NCMI) in a joint cellular and D2D setup to recover missing packets in the second stage. Namely, we develop NCMI-Batch based on batch network coding and NCMI-Instant based on instantly decodable network coding (IDNC) [25]. In NCMI-Batch, each transmitted packet to device \( n \in \mathcal{N} \) is a linear combination of the missing packets in that device and thus it carries information about all missing packets. Therefore, all missing packets can be decoded at device \( n \) once enough number of packets are received successfully by device \( n \). In NCMI-Instant, each transmitted packet to device \( n \in \mathcal{N} \) carries information about only one of the missing packets at that device. Therefore, one missing packet can be decoded each time a transmitted packet is received successfully by device \( n \).

The integral part of our work is to analyze the throughput performance of NCMI-Batch and NCMI-Instant. The amount of time required to recover the missing packets is an indicator of resource (such as energy, time, and bandwidth) consumption. Therefore, we consider the packet completion time as the performance metric, which is defined as follows:

Definition 1: Packet completion time \( T \) is the number of transmission slots in the 2nd stage that are required for all devices to receive and decode all packets in their \( \text{Wants} \) sets.

In our joint cellular and D2D setup, \( \eta_n, n \in \mathcal{N} \) denotes the loss probability over the cellular link towards device \( n \) and \( \epsilon_{k,l} \) denotes the loss probability over the D2D link when device \( k \) transmits a packet to device \( l \). We assume that \( \eta_n \) and \( \epsilon_{k,l} \) are i.i.d. according to a uniform distribution.

Assumptions: We assume, without loss of generality, that for each packet \( p_m \in \mathcal{M} \), there is at least one mobile device that wants packet \( p_m \). In other words, \( \forall p_m \in \mathcal{M}, \exists n \in \mathcal{N} \) such that \( p_m \in \mathcal{W}_n \). This assumption does not violate generality, because packets that are not wanted by any of the devices could be removed from \( \mathcal{M} \).

III. NCMI AND UPPER BOUNDS ON \( T \)

In this section, we develop network coding algorithms with multiple interfaces (NCMI) for the joint cellular and D2D setup, and provide upper bounds on their packet completion time. In particular, we develop two network coding algorithms; NCMI-Batch, which uses random linear network coding, and NCMI-Instant, which provides instant decodability guarantee. We first consider the case of no loss in the second stage, while there is loss in the first stage, and analyze NCMI-Batch and NCMI-Instant. The analysis with no loss provides us insight while designing network coding algorithms for the lossy scenario in the second stage.\(^2\)

A. No Loss in the Second Stage

In this section, we assume that cellular connections in the first stage are lossy, but both the cellular and D2D connections are lossless in the second stage. Thus, all the transmitted packets in the second stage are received correctly. We develop network coding algorithms NCMI-Batch and NCMI-Instant, and develop upper bounds on the packet completion time.

1) NCMI-Batch:

a) Algorithm description: As we mentioned earlier in Section II, our system model consists of two stages. In the first stage, all packets are broadcast to all devices via cellular links without network coding. In the second stage, both cellular and D2D links are utilized simultaneously and network coding is employed. In particular, both the source and one of the devices in the local area transmit network coded packets simultaneously in each transmission slot until there is no missing packet in the local area. In NCMI-Batch the transmitted packets are formed as a linear combination of the missing packets and thus carry information about all packets in the set \( \mathcal{M} \). Therefore, to minimize the packet completion time, the network coded packets to be transmitted from the source (through the cellular links) and in the local area among the mobile devices (through D2D links) are selected such that they contain as much information as possible about all missing packets. Next, we explain the details of how network coding is performed by the source and the local area devices in each transmission slot.

The source (i) determines the missing packets in all mobile devices, and (ii) transmits linear combinations of these packets (using random linear network coding over a sufficiently large field) through cellular links. These network coded packets are innovative and beneficial for any device \( n \) for which \( |\mathcal{H}_n| \leq M \), because these network coded packets carry information about all missing packets in the local area. After each transmission, if the received packet is innovative for device \( n \), it is inserted into the set \( \mathcal{H}_n \). The procedure continues until each device \( n \) receives \( |\mathcal{W}_n| \) innovative packets.

On the other hand, in the local area, the mobile device \( n_{\text{max}} \) with the largest \( \text{Has} \) set has the most information about the missing packets among all cooperating devices. Therefore, at each transmission slot, one of the devices is selected randomly as the controller; the controller selects \( n_{\text{max}} = \arg\max_{n \in \mathcal{N}} |\mathcal{H}_n| \) as the transmitter. If there are multiple such devices, one of them is selected randomly. The transmitter linearly combines all packets in its \( \text{Has} \) set, \( \mathcal{H}_{n_{\text{max}}} \), and broadcasts the network coded packet to all other mobile devices via D2D links. This network coded packet is beneficial to any device \( k \neq n_{\text{max}} \) (any device except for the transmitter)

\(^2\)We also note that computational complexity and signaling overhead of lossless NCMI are lower as compared to their lossy versions as we will demonstrate later in this section and in the Supplementary Material. Thus, if D2D links are lossless, we can directly use lossless NCMI to enjoy lower computational complexity and signaling overhead.
as long as $W_k \setminus \mathcal{M}_c$ is not an empty set. The reason is that device $n_{\text{max}}$ has the most number of packets from the set $\mathcal{M}$ among all devices. Therefore, $n_{\text{max}}$ is the only device that has innovative information about the missing packets at all devices $k \in (\mathcal{N} \setminus n_{\text{max}})$. After each transmission, if the received packet has innovative information for device $n$, the received packet is inserted into the $\text{Has}$ set of device $n$. Note that the network coded packets that include packets from $\mathcal{M}_c = \bigcap_{n \in \mathcal{N}} W_n$ can only be transmitted from the source, since these packets do not exist in any of the devices. Therefore, the devices stop transmitting network coded packets through D2D links if each device $n$ receives (i) $|W_n| - |\mathcal{M}_c|$, if (ii) $|W_n|$ innovative packets from D2D links, or (iii) $|W_n|$ innovative packets from both the cellular and D2D links.

Next, we characterize the time until all missing packets are recovered, and calculate the packet completion time, $T$.

b) Upper bound on $T$: In order to characterize the performance of proposed NCMI-Batch, we consider the worst case scenario and develop an upper bound on the packet completion time of NCMI-Batch. We note that NCMI-Batch is guaranteed to outperform the upper bound, as proved in the Supplementary Material and shown in the simulation results.

Theorem 1: Packet completion time; $T$ when NCMI-Batch is employed by cooperative mobile devices in a joint cellular and D2D setup is upper bounded by

\[
T \leq \max \left( |\mathcal{M}_c|, \frac{1}{3} \max_{n \in \mathcal{N}} |W_n| \right) + \min_{n \in \mathcal{N}} |W_n| \left( \frac{1}{2} \max_{n \in \mathcal{N}} |W_n| \right).
\]

Proof: The proof is provided in the Supplementary Material.

Example 3: Let us consider three mobile devices with the Wants sets: $W_A = \{p_1, p_2, p_3\}, W_B = \{p_1, p_4, p_5\}, W_C = \{p_1, p_6, p_7\}$. Using NCMI-Batch, in the first transmission slot, the source transmits a linear combination of the packets $p_1, p_2, ..., p_7$, which is innovative for all devices as it carries information about all missing packets. Meanwhile, in the local area, the device with the largest $\text{Has}$ set is selected. Since there is equality in this example ($|\mathcal{H}_A| = |\mathcal{H}_B| = |\mathcal{H}_C| = 4$), one device is selected randomly, let us say device $A$. Device $A$ transmits a linear combinations of $p_1, p_2, p_5, p_6, p_7$ via D2D links. Note that this network coded packet is beneficial to both devices $B$ and $C$, as it carries information about their missing packets. Therefore, at the end of the first transmission slot, device $A$ receives one innovative packet and thus the size of its $\text{Has}$ set is increased by one and devices $B$ and $C$ receive two innovative packets and thus the sizes of the $\text{Has}$ set for these devices are increased by two; i.e., $|\mathcal{H}_A| = 5, |\mathcal{H}_B| = |\mathcal{H}_C| = 6$. In the second transmission slot, the source transmits a linear combination of $p_3, p_4, p_5, p_7$, which is innovative for all devices $A, B, C$ and at the same time $B$ or $C$ (with the larger size of $\text{Has}$ set than $A$) transmits a linear combination of the packets in its $\text{Has}$ set, which is innovative for $A$. Therefore, at the end of the second transmission slot, device $A$ receives two innovative packets and thus the size of its $\text{Has}$ set is increased by two and devices $B$ and $C$ receive one innovative packet (they only need one innovative packet to be satisfied) and thus the sizes of the $\text{Has}$ sets for these devices are increased by one; i.e., $|\mathcal{H}_A| = 7, |\mathcal{H}_B| = |\mathcal{H}_C| = 7$. As seen, each device receives three innovative packets after two transmission slots and thus the packet completion time is equal to $T = 2$. On the other hand, from Theorem 1, the upper bound for the packet completion time is equal to $\max \left( \frac{1}{3}(3 + 3), \frac{4}{2} \right) = 2$. As seen, the inequality $T \leq 2$ provided in (1) is satisfied, in this example. It can also be seen that the upper bound is tight, in this example.

c) Computational complexity:

Lemma 2: The complexity of NCMI-Batch, when the channels are lossless in the second stage, is polynomial with the complexity of $O(M^2 + N)$.

Proof: The proof is provided in the Supplementary Material.

2) NCMI-Instant: This section develops a heuristic; NCMI-Instant, to network code packets in a way that they can be decoded immediately after they are received by the mobile devices. NCMI-Instant is crucial for loss tolerant real-time applications with deadline constraints.

a) Algorithm description: NCMI-Instant determines the IDNC packets to be transmitted at each transmission slot through the cellular and D2D links with the goal of minimizing the packet completion time. To reach this goal, the optimal way is exhaustively creating all combinations of IDNC packets that can be transmitted from the source and the mobile devices in all transmission slots and selecting the sequence of packets that results in the minimum packet completion time. However, the complexity of exhaustive search is high and thus in this paper, we proposed a heuristic method NCMI-Instant with linear complexity with respect to the number of devices and quadratic complexity with respect to the number of packets. NCMI-Instant consists of three steps: (i) creating IDNC packets (Algorithm 1), (ii) grouping the created IDNC packets into the sets $\mathcal{M}_c, \mathcal{M}_t, \mathcal{M}_d$ (Algorithm 1), and (iii) determining the IDNC packets to be transmitted from the two interfaces based on the the created groups. In the following, we explain these steps.

Step 1 (Creating IDNC Packets): Algorithm 1 is a greedy algorithm that creates IDNC packets, sequentially, from the packets in the set of missing packets in all devices. The main idea behind this algorithm is to check uncoded packets sequentially and try to merge them into IDNC packets. We note that similar ideas have been considered in network coding literature, e.g., a greedy algorithm for creating network coding packets over wireless mesh networks is developed using a similar approach in [12]. According to Algorithm 1, the first IDNC packet is created with the first uncoded packet. Then for each remaining uncoded packets, the algorithm checks all of the previously created IDNC packets; if the uncoded packet can be merged with one of the IDNC packet (i.e., the merged IDNC packet is instantly decodable for all devices that want one of the uncoded packets in the merged IDNC packet), the IDNC packet would be updated by adding the uncoded packet. Otherwise, a new IDNC packet containing the uncoded packet is created. This algorithm creates IDNC packets with complexity of $O(M^2 + N)$. Note that the complexity of exhaustive search to create all possible IDNC packets is NP-hard. Next, we describe the details of Algorithm 1 on Step 1, creating IDNC packets.

In Algorithm 1, we define vectors with length $N$, whose elements are either \textit{NULL} or $p_m \in \mathcal{M}$. A network coded packet is associated to each vector and constructed as a linear combination of all elements of the vector that are not equal to \textit{NULL}. We describe how each vector is defined by Algorithm 1 in the following. First, the vector $\mathbf{v}_n$ with the length of $N$ is defined for each packet $p_m \in \mathcal{M}$.
Algorithm 1: Grouping the Packets in the Wants Sets

1: for any packet $p_m$ in $M$ do
2: Define vector $v_m$ with size $N$. Each element of the vector $v_m$ is initially set to NULL; i.e., $v_m[n] = NULL, \forall n \in N$.
3: for any device $n$ in $N$ do
4: if $p_m$ is wanted by device $n$ then
5: $v_m[n] = p_m$ and $p_m$ is removed from the Wants set $W_n$.
6: if there exists a vector $v_{m'}$, $m' < m$ that satisfies the following condition: for any $n$, for which $v_{m'}[n] = p_m$ then $v_{m'}[n] = NULL$ then
7: Replace $v_{m'}$ with $v_m + v_m$ and delete $v_m$. (Note that $p_m + NULL = p_m$ for any $m$)
8: Each element of $M_c$ is constructed by network coding all packets in a vector $v_m$ if all elements of $v_m$ are the same and not equal to NULL; i.e., $v_m[1] \neq NULL$ and $v_m[n] = v_m[1], \forall n \in N$.
9: Each element of $M_d$ is constructed by network coding all packets in a vector $v_m$ if $v_m$ contains at least one element equal to NULL; i.e., $\exists x \in N \mid v_m[x] = NULL$.
10: Construct $M_I$ using the remaining vectors. In other words, each element of $M_I$ is constructed by network coding all packets in a vector $v_m$ if $v_m$ does not contain any NULL element and contains at least two different elements; i.e., $v_m[n] \neq NULL, \forall n \in N$ and $\exists n, x \mid v_m[n] \neq v_m[x]$.

nth element of this vector, where $n$ is any device that wants that packet, is initially set to $p_m$. The value of this vector for the remaining elements, which correspond to the devices that have that packet, is initially set to NULL (lines 1-5). A network coded packet is instantly decodable for device $n$ if it contains information about one and only one of the packets in the Wants set of device $n$. Let us consider packet $p_m$ with its corresponding vector $v_m$ and the instantly decodable network coded packet corresponding to vector $v_m$. These two packets can be merged as a new instantly decodable network coded packet (line 7) if the value of vector $v_m$ for all devices that want packet $p_m$, i.e., any device $n$ for which $v_m[n]$ is $p_m$) is NULL (the condition of if statement at line 6).

Step 2 (Grouping the Created IDNC Packets): After creating the IDNC packets, we group them into the sets, $M_c$, $M_I$, and $M_d$, based on their differences from the viewpoint of cellular and D2D links. The packets in $M_c$ can only be transmitted via cellular links. The packets in $M_d$ can be transmitted via cellular or D2D links, but it may take more time slots if they are transmitted via D2D links. The packets in $M_d$ can be transmitted either via cellular or D2D links, and the number of time slots for transmitting a packet is the same in both cellular and D2D. The sets $M_c$, $M_I$, and $M_d$ are created as the following. $M_c$ is the set of packets that are not available in the local area and should only be re-broadcast from the base station (source) to all devices. These are the packets that are wanted by all devices. $M_d$ is the set of IDNC packets that can be transmitted by a single transmission from either the source or a mobile device. These are the packets that can be formed by combining a subset of the packets in the $H$ set of one of the devices (note that any combination of packets can be created in the source as it has all packets). $M_I$ is the set of remaining IDNC packets that can be transmitted by a single transmission from the source or by two transmissions (according to Lemma 3) in the local area. These are the packets that contain one and only one uncoded packet from the Wants set of each device and thus they are not available in one single device. Therefore, in order to send these packets through D2D links, they need to be divided into two IDNC packets and each part can be transmitted from the device that has all the corresponding uncoded packets.

Lemma 3: Each network coded packet in $M_I$ can be transmitted by exactly two transmission slots using D2D links by splitting the packet into two (network coded) packets.

Proof: The proof is provided in the Supplementary Material.

The properties of the sets $M_c$, $M_I$, and $M_d$ are summarized in Table I. Next, we describe the details of Algorithm 1 on Step 2: grouping the created IDNC packets.

The set $M_c$ consists of all packets that are wanted by all devices, i.e., by using any vector $v_m$ whose elements are the same and not equal to NULL (line 8). The set $M_d$ consists of the network coded packets that can be formed by combining a subset of the packets in the $H$ set of one of the devices, called $x$; i.e., by using the packets in the vector $v_m$ whose $x$th element is NULL (line 9). The set $M_I$ consists of the rest of network coded packets. In other words, each network coded packet in $M_I$ contains one and only one packet from the Wants set of each device; i.e., by using any vector $v_m$ whose elements are not equal to NULL and it contains at least two different elements (line 10). Next, we give an example on how Algorithm 1 works.

Example 4: Let us assume that there are three mobile devices with the Wants sets: $W_A = \{p_1, p_4\}$, $W_B = \{p_1, p_2, p_3\}$, $W_C = \{p_1, p_4, p_5\}$. Note that $M = \bigcup_{n \in N} W_n = \{p_1, p_2, p_3, p_4, p_5\}$ and $H_n = M \setminus W_n$ for $n \in \{A, B, C\}$; $H_A = \{p_2, p_3, p_5\}$, $H_B = \{p_4, p_5\}$, $H_C = \{p_2, p_3\}$. According to Algorithm 1, $v_1$ is equal to $[p_1, p_1, p_1]$, because it is wanted by all three devices, as shown in Table II. Then, for packet $p_2$, we define vector $v_2 = [NULL, p_2, NULL]$, because $p_2$ is wanted by device $B$ only. Similarly, for packet $p_3$, we define vector $v_3 = [NULL, p_3, NULL]$, $v_3$ cannot be merged with $v_1$ or $v_2$, because the second element of $v_3$ is $p_3$, while the second elements of $v_1$ and $v_2$ are not NULL. In the next step, we define vector $v_4 = [p_4, NULL, p_4]$. $v_4$ can be merged with $v_2$, because the first and second elements of $v_4$ is $p_4$ and the first and the second elements of $v_5$ is NULL. Therefore, $v_2$ is updated as $v_2 + v_4 = [p_4, p_2, p_4]$ and $v_4$ is deleted, as shown in Table II. Similarly, $v_5$ is defined as $[NULL, NULL, p_5]$, $v_5$ can be merged with $v_3$, because the third element of $v_5$ is $p_5$ and the third element of $v_3$ is NULL. Therefore, $v_3$ is updated as $v_3 + v_5 = [NULL, p_3, p_5]$ and $v_5$ is deleted, as shown in Table II.

All elements of $v_1$ are the same and not equal to NULL. Therefore, $M_c$ is constructed from $v_1$: $M_c = \{p_1\}$, as shown

| Set | Cellular Link | D2D Link |
|-----|--------------|----------|
| $M_c$ | 1 | N/A |
| $M_I$ | 1 | 2 |
| $M_d$ | 1 | 1 |
in Table II, \( v_3 \) has a NULL element. \( M_d \) is constructed from \( v_3 \); \( M_d = \{ p_3 + p_5 \} \). \( v_2 \) does not have any NULL element and its 1st and 2nd elements are different. \( M_l \) is constructed from vector \( v_2 \); \( M_l = \{ p_2 + p_6 \} \).

Step 3 (Determining the IDNC Packets to be Transmitted From the Two Interfaces): At each transmission slot, two packets are selected from the sets \( M_c \cup M_l \cup M_d ; \) one to be transmitted from the base station and another one to be transmitted from one of the mobile devices. The idea behind packet selection in lossless NCMI-Instant is that the created IDNC packets can be transmitted in the minimum packet completion time. Note that the decisions of which network coded packet is transmitted and which device is selected as the transmitter in the local area, are made by the controller (which can be selected randomly among mobile devices).

Packet Selection by the Source: The packets in \( M_c \) should only be transmitted from the source. On the other hand, a packet transmission from the set \( M_l \) targets all devices. Therefore, the source selects a packet from \( M_c \) to transmit, if this set is not empty. Among the sets \( M_l \) and \( M_d \), the packets in \( M_l \) targets all devices, while the packets in \( M_d \) targets a subset of devices. On the other hand, packets in \( M_l \) require less number of transmissions if transmitted from the source than the mobile devices. Therefore, the source selects a packet from \( M_l \) to transmit if \( M_c \) is empty. At last, if both \( M_c \) and \( M_l \) are empty, a packet in \( M_d \) is selected to be transmitted from the source.

Packet Selection by the Mobile Devices: Any packet from the set \( M_d \) can be transmitted from the local area by a single transmission, however only a partial of a packet from the set \( M_l \) can be transmitted by a single transmission from the local area, so the order of transmitting the packets in the local area is (i) \( M_d \) and (ii) \( M_l \).

b) Upper bound on \( T \): Now, we analyze the packet completion time performance of NCMI-Instant by developing an upper bound on the packet completion time.

Theorem 4: When NCMI-Instant is employed, packet completion time by cooperative mobile devices in a joint cellular and D2D setup is upper bounded by:

\[
T \leq \min\left(\max\left(\frac{M}{2}, |M_c|\right), \max\left(\frac{1}{3} (2 \min_{n \in N} |W_n| + |M_d|), \frac{1}{2} \min_{n \in N} |W_n| + |M_d|\right)\right). \tag{2}
\]

Proof: The proof is provided in the Supplementary Material.

Example 5: Let us assume that there are three mobile devices with the Wants sets; \( W_A = \{ p_1, p_2, p_3, p_4, p_7, p_9 \} \), \( W_B = \{ p_1, p_2, p_5, p_7, p_{10} \} \), \( W_C = \{ p_1, p_3, p_6, p_8 \} \). Note that \( M = \bigcup_{n \in N} W_n = \{ p_1, \ldots, p_{10} \} \) and \( M_n = M \setminus W_n \) for \( n \in \{ A, B, C \} \); \( H_A = \{ p_3, p_5, p_6, p_8, p_{10} \} \), \( H_B = \{ p_3, p_4, p_6, p_8, p_9 \} \), \( H_C = \{ p_2, p_4, p_5, p_7, p_9, p_{10} \} \). According to Algorithm 1, \( M_c = \{ p_1 \} \), \( M_d = \{ p_3 + p_10 \} \), and \( M_l = \{ p_2 + p_3, p_4 + p_5, p_8 + p_7 + p_8 \} \) in this example. Using NCMI-Instant, in the first transmission slot packet \( p_1 \) (in set \( M_c \)) is transmitted from the source and packet \( p_9 + p_{10} \) (in set \( M_d \)) is transmitted from device C. In the second transmission slot, packet \( p_2 + p_8 \) (in set \( M_l \)) is transmitted from the source and packet \( p_7 \) (the first split packet of the third network coded packet in set \( M_l \) available at device C) is transmitted from device C. In the third transmission slot, packet \( p_4 + p_5 + p_6 \) (in set \( M_l \)) is transmitted from the source and packet \( p_8 \) (the second split packet of the third network coded packet in set \( M_l \) available at device A and B) is transmitted from device A or B. Therefore, in total three transmission slots are required by NCMI-Instant; \( T = 3 \). On the other hand, from Theorem 4, the upper bound for the packet completion time is equal to \( \lceil \min(5, \max(1, 3, 2.5)) \rceil = 3 \).

As seen, the inequality \( T \leq 3 \) provided in (2) is satisfied, in this example. It can also be seen that the upper bound is tight, in this example.

\[ \boxdot \]

c) Computational complexity:

Lemma 5: The complexity of NCMI-Instant when the channels are lossless in the second stage is polynomial with \( O(M^2 + N) \).

Proof: The proof is provided in the Supplementary Material.

B. Lossy Links in the Second Stage

In the previous section, we developed the network coding algorithms NCMI-Batch and NCMI-Instant for the case that packets in the second stage are received successfully in the receiver devices without any loss (Fig. 3(a)). In this section, we consider a more realistic scenario, where the channels are lossy in the second stage (Fig. 3(b)).

We assume that each packet transmitted from the source to each device \( n \in N \) is lost with probability of \( \eta_n \). Similarly, each packet transmitted from the transmitter device \( k \) to the receiver device \( l \) is lost with probability of \( \epsilon_{k,l} \). We also assume that probability of all channel losses, \( \eta_n, \forall n \in N \) and D2D channel losses, \( \epsilon_{k,l}, \forall k, l \in N \), are i.i.d. and uniformly distributed.

The packet completion time for the lossy links in the second stage, depends on the packets that are received successfully in addition to the transmitted packets at each transmission slot. Therefore, for each transmitted packet, we define targeted receivers, \( \mathcal{N}_r \), as the set of devices for which the transmitted packet is innovative. We also define successful receivers as the set of devices for which the transmitted packet is innovative and is received successfully. We consider the average number of successful receivers for each transmitted packet, which is calculated as follows.

Average number of successful receivers, for a packet transmitted from the source to the set of targeted receivers \( \mathcal{N}_r \),
is equal to the average number of packets that are received successfully at the devices in the set $\mathcal{N}_r$ and calculated as $\sum_{n \in \mathcal{N}_r} (1 - \eta_n)$. Similarly, average number of successful receivers for a packet transmitted from the transmitter device $t$ to the set of targeted receivers $\mathcal{N}_t$ is equal to $\sum_{n \in \mathcal{N}_t} (1 - \epsilon_{t,n})$.

Note that, in the case of no loss, the average number of successful receivers with the set of targeted receivers, $\mathcal{N}_r$, is equal to the size of this set; $|\mathcal{N}_r|$.

I) NCMI-Batch:

a) Algorithm description: As described in Section III-A1, for the case of lossless channels in the second stage, network coding decisions by NCMI-Batch are made such that the transmitted packets contain as much information as possible about missing packets in all devices. For the case of lossy channels, the transmitted packets are selected such that the successfully delivered packets contain as much information as possible about the missing packets in all devices. In other words, the transmitted packets are selected such that the average number of successful receivers are maximized. Packet selection algorithm by NCMI-Batch at each transmission slot for the lossy channels is provided in Algorithm 2. According to this algorithm, two packets are selected; one to be transmitted from the source and the other one to be transmitted in the local area. The details of packet selection in Algorithm 2 are described next.

Algorithm 2 NCMI-Batch for Lossy Channels

1: $\mathcal{M}$: the set of the missing packets in all devices.

Packet Selection by the Source:

2: A network coded packet is generated as a linear combination of all packets in $\mathcal{M}$.

3: The source broadcasts this packet to all mobile devices.

Packet Selection by the Mobile Devices:

4: for any device $n$ in $\mathcal{N}$ do

5: $\mathcal{N}_r = \{k \mid (\exists p \in \mathcal{H}_n \mid p \not\perp H_k)\}$. i.e., packet $p$ (which could be network coded or not) cannot be expressed as a linear combination of the packets in $\mathcal{H}_k$.

6: $Ave_n = \sum_{k \in \mathcal{N}_r} (1 - \epsilon_{n,k})$.

7: $x = \arg\max_{n \in \mathcal{N}_r} Ave_n$.

8: A network coded packet is generated as a linear combination of all packets in $\mathcal{H}_x$.

9: $x$ broadcasts this packet to all other mobile devices.

Packet selection by the source (lines 2-3): The average number of successful receivers, for a transmitted packet from the source with the set of targeted receivers, $\mathcal{N}_r$, is equal to $\sum_{n \in \mathcal{N}_r} (1 - \eta_n)$. On the other hand, any device $n$ with $|\mathcal{W}_n| > 0$ or equivalently $|\mathcal{H}_n| < M$, is interested in receiving an innovative packet and can be a member of $\mathcal{N}_r$ for the transmitted packet from the source. Therefore, in order to maximize the average number of successful receivers, we need to enlarge the set $\mathcal{N}_r$ to all devices that need an innovative packet; i.e., $\mathcal{N}_r = \{n \mid (|\mathcal{H}_n| < M)\}$. According to Algorithm 2, in NCMI-Batch, the source (i) determines the missing packets in all mobile devices, and (ii) transmits linear combinations of these packets (using random linear network coding over a sufficiently large field) as the network coded packets through cellular links. This network coded packet carry information about all missing packets in the local area, and thus is innovative and beneficial for any device $n$ for which $|\mathcal{H}_n| < M$; i.e., $\mathcal{N}_r = \{n \mid (|\mathcal{H}_n| < M)\}$.

Therefore, the packet selected to be transmitted from the source in NCMI-Batch maximizes the average number of successful receivers. After each transmission, if the received packet is innovative for device $n$, it is inserted into $\mathcal{H}_n$ set. The procedure continues until each device $n$ receives $|\mathcal{W}_n|$ innovative packets.

Packet selection by the mobile devices (lines 4-9): In NCMI-Batch, the controller (which can be randomly selected among mobile devices) selects one of the devices as the transmitter (according to line 7). Then, the transmitter transmits a linear combination of the packets in its $\mathcal{H}_n$ set to all other devices through D2D links. The set of targeted receivers, $\mathcal{N}_r$, for a packet transmitted from device $n$, is the set of devices, for which a random linear combination of the packets in $\mathcal{H}_n$ is innovative. In other words, if there is at least one packet from $\mathcal{H}_n$ that is linearly independent of all packets in $\mathcal{H}_k$, then $k$ is a member of $\mathcal{N}_r$; $\mathcal{N}_r = \{k \mid (\exists p \in \mathcal{H}_n \mid p \not\perp H_k)\}$. Accordingly, the average number of successful receivers for a packet transmitted from $n$ is equal to $\sum_{k \in \mathcal{N}_r} (1 - \epsilon_{n,k})$. In NCMI-Batch, the mobile device $x$ with the largest average number of successful receivers is selected as the transmitter among all devices at each transmission slot; $x = \arg\max_{n \in \mathcal{N}_r} \sum_{k \in \mathcal{N}_r} (1 - \epsilon_{n,k})$. If there are multiple of such devices, one of them is selected randomly. The transmitter linearly combines all packets in its $\mathcal{H}_n$ set, $\mathcal{H}_x$, and broadcasts the network coded packet to all other mobile devices. This network coded packet is beneficial to all devices in the set of targeted receivers (by the transmitter) that receive the packet successfully.

After each transmission, if the successfully received packet has innovative information for device $k$, it is inserted into the $\mathcal{H}_k$ set of device $k$. Similar to the case of lossless channels, the cooperating devices stop transmitting network coded packets if each device $n$ successfully receives $|\mathcal{W}_n|$ innovative packets from both the source and cooperating devices. Note that selecting the device with the largest average number of successful receivers for the case of no loss, $\epsilon_{n,k} = 0, \forall n, k \in \mathcal{N}$, is equivalent to selecting the device with the largest number of targeted receivers and thus the device with the largest $\mathcal{H}_n$ set. This is aligned with our strategy, presented in Section III-A1, for packet selection in NCMI-Batch when the channels are lossless in the second stage. Next, we give an example on NCMI-Batch for lossy channels.

Example 6: Let us consider three mobile devices with the Wants sets; $\mathcal{W}_A = \{p_1, p_2, p_3\}$, $\mathcal{W}_B = \{p_1, p_4, p_5\}$, $\mathcal{W}_C = \{p_1, p_6, p_7\}$, the Has sets $\mathcal{H}_A = \{p_4, p_5, p_6, p_7\}$, $\mathcal{H}_B = \{p_2, p_3, p_6, p_7\}$, $\mathcal{H}_C = \{p_2, p_3, p_4, p_5\}$ and probabilities of channel losses; $\eta_A = 0.35, \eta_B = 0.4, \eta_C = 0.45, \epsilon_{A,B} = 0.1, \epsilon_{A,C} = 0.3, \epsilon_{B,C} = 0.2, \epsilon_{C,A} = 0.3, \epsilon_{C,B} = 0.2$. By using NCMI-Batch, $p_1, \ldots, p_7$ are combined as a network coded packet in the source and transmitted to all devices in the first slot. This transmitted packet can be beneficial to all devices ($\mathcal{N}_r = \{A, B, C\}$) as it carries information about all missing packets. Note that the average number of successful receivers for this transmitted packet is equal to $(1 - 0.35) + (1 - 0.4) + (1 - 0.45) = 1.8$. Meanwhile, in the local area, the device with the largest average number of receivers is selected as the transmitter. The set of targeted receivers for a random linear network coded packet transmitted from $A$ is $\{B, C\}$, as $p_4 \in \mathcal{H}_A$ (or $p_5 \in \mathcal{H}_A$) is linearly independent of all the packets in $\mathcal{H}_A$ and $\mathcal{H}_B$; $\mathcal{H}_B = \{p_2, p_3, p_6, p_7\}$ for device $B$.

\[ \text{\cite{NCMI-Batch}} \]
independent of $\mathcal{H}_B = \{p_2, p_3, p_6, p_7\}$ and $p_6 \in \mathcal{H}_A$ (or $p_7 \in \mathcal{H}_A$) and thus the average number of successful receivers is equal to $(1-\epsilon_{A,B}) + (1-\epsilon_{A,C}) = 1.6$. Similarly, the average number of successful receivers for a random linear network coded packet transmitted from $B$ and $C$ is equal to 1.7 and 1.5, respectively. Therefore, device $B$ with the maximum average number of successful receivers is selected as the transmitter and transmits a random linear combination of $p_2, p_3, p_6, p_7$. Note that this network coded packet can be received successfully at devices $A$ and $C$ with probabilities of 0.9 and 0.8, respectively. Thus, in the first slot, the source transmits a random linear combination of $p_1, \ldots, p_7$ and device $B$ transmits the same linear combination of $p_2, p_3, p_6, p_7$, simultaneously. The procedure is repeated at every slot until each device receives 3 innovative packets. □

Next, we characterize how long it takes until all missing packets are recovered, i.e., the packet completion time $T$.

b) Upper bound on $T$: In order to characterize the performance of NCMI-Batch, we provide an upper bound on the packet completion time obtained from NCMI-Batch in the following theorem.

**Theorem 6:** The average packet completion time $T$ when NCMI-Batch is employed by cooperative mobile devices in a joint cellular and D2D setup when the channel links are lossy is upper bounded by

$$T \leq \left[ \max \left( \frac{|M_c|}{1 - \prod_{n \in N} \eta_n}, T_j \right) \right],$$

where,

$$T_j = \begin{cases} \max(T_x, T_r), & \text{if } |W_{x_e}| - |W_{x_c}| \leq \frac{|W_{x_e}|}{1 - \eta_r - \epsilon_{x,r} + \eta_x}, \\ \frac{|W_{x_e}|}{2 - \eta_r - \epsilon_{x,r}}, & \text{otherwise}, \end{cases}$$

$$T_x = \frac{|W_{x_e}|(1 - \epsilon_{x,r}) + |W_{x_c}|(1 + 2\eta_x - 2\eta_r + \eta_x - \epsilon_{x,r})}{(1 - \eta_r - \epsilon_{x,r} + \eta_x)(3 - 2\eta_x - \epsilon_{x,r})},$$

$$T_r = \frac{|W_{r_e}|(1 - 2\eta_r - \epsilon_{x,r} + \eta_x) + |W_{x_e}|(1 - \epsilon_{x,r})}{(1 - \eta_r - \epsilon_{x,r} + \eta_x)(3 - 2\eta_r - \epsilon_{x,r})},$$

$$x = \arg \max_{n \in N} |\mathcal{H}_n| = \arg \min_{n \in N} |W_{n_e}|,$n$$

$$r = \arg \max_{n \in (N \setminus x)} |W_{n_e}| - \eta_n - \epsilon_{x,n}.$$  

**Proof:** The proof is provided in the Supplementary Material.

c) Computational complexity:

**Lemma 7:** The complexity of NCMI-Batch when the channels are lossless in the second stage is polynomial with $O(M^2 + N)$.  

**Proof:** The proof is provided in the Supplementary Material.

2) NCMI-Instant: In section III-A2, we described and analyzed NCMI-Instant for the case that the channel links are lossless. In this section, we consider a more generalized case where the channel links are lossy and present NCMI-Instant for this generalized case.

a) Algorithm description: NCMI-Instant algorithm, described in Section III-A2a, groups the packets that are wanted by the devices into sets $M_c, M_l, and M_d$ when there is no loss in the second stage. Then, packets from these sets are network coded and transmitted from the cellular and D2D links. Yet, when the links are lossy in the second stage, this approach should be revised for the following reasons.

First, each network coded packet is transmitted successfully when there is no loss. This makes creating fixed $M_c, M_l, and M_d$ sets possible and reasonable before transmitting the packets in the second stage. However, when the channels are lossy in the second stage, the transmitted network coded packet at each transmission slot may be received successfully by some of the targeted receivers (known as successful receivers) and may be lost by the others. Thus, fixed $M_c, M_l, and M_d$ sets are not appropriate in this scenario, and they should be updated after each transmission depending on successful packet transmissions.

Also, when the channel links are lossy, different network coded packets even if they are in the same set (such as in $M_l$) have different priorities for transmission. In particular, the packets that can be received by possibly larger number of devices successfully should be prioritized as it would deliver more information in one transmission, which would eventually reduce the packet completion time. The details of packet selection in Algorithm 3 are described below.

**Packet selection by the source (lines 2-7):** The average number of successful receivers for any packet in the set $M_c \cup M_l$ that is transmitted from the source is equal to $\sum_{n \in N}(1 - \eta_n)$, because this packet targets all devices. On the other hand, the average number of successful receivers for any packet $p$ in the set $M_d$ with its corresponding vector $v_p$ is equal to $\sum_{n \in \{v_p[n] \neq NULL\}}(1 - \eta_n)$, because this packet is innovative for any device $n \in N$ for which $v_p[n]$ is not NULL and thus the set of targeted receivers is $N_r := \{n \in N | (v_p[n] \neq NULL)\}$. Since $N_r$ is a subset of $N$, $\sum_{n \in N_r}(1 - \eta_n)$, the average number of successful receivers for any packet in $M_c$ or $M_l$, is greater than $\sum_{n \in N_r}(1 - \eta_n)$, the average number of successful receivers for any packet in $M_d$ and thus, a packet from the set $M_c$ or $M_l$ is prioritized to be transmitted from the source than a packet from $M_d$. Between the sets $M_c$ and $M_l$, a packet from the set $M_c$ is prioritized to be selected (lines 2-3), because the packets in $M_c$ are not available to be transmitted through the other interface. Therefore, the order of transmitting the packets from the source is (i) $M_c$, (ii) $M_l$, and (iii) $M_d$. There is no priority among the packets of the same set for the sets $M_c$ and $M_l$, because all packets have the same average number of successful receivers (lines 4-5). For the set $M_d$, the packet with the maximum average number of successful receivers is prioritized to be transmitted (lines 6-7).

**Packet selection by the mobile devices (lines 8-10):** We first consider packet $p$ from set $M_l$ with its vector $v_p$. Each device $x \in N_r$ can transmit a partial of this packet that is available in its $Has$ set. $v_p[x]$ is the uncoded packet in the network coded packet $p$ that is wanted by device $x$. Therefore,
Algorithm 3 NCMI-Instant for Lossy Channels

1: Group the packets in the Wants sets of all devices into the sets $M_c$, $M_l$, and $M_d$ using Algorithm 1 and keep all network coded packets, $p \in (M_c \cup M_l \cup M_d)$, along with their corresponding vectors, $v_p$, in these sets.

Packet Selection by the Source:

2: if $M_c$ is not empty then
3: The first element of $M_c$ is selected to be transmitted from the source.
4: else if $M_l$ is not empty then
5: The first element of $M_l$ is selected to be transmitted from the source.
6: else if $M_d$ is not empty then
7: The packet with the maximum average number of successful receivers among all packets in the set $M_d$, which is equal to $\arg \max_{p \in M_d} \sum n(v_p[n] \neq NULL)(1 - \eta_n)$, is selected to be transmitted from the source.

Packet Selection by the Mobile Devices:

8: Consider packet $p$ with its corresponding vector $v_p$ in $M_l$; any device in $\mathcal{N} \in \mathcal{N}^C$ can transmit partial of packet $p$ which is constructed by network coding all packets in the vector $v_p \setminus v_p[x]$. The average number of successful receivers for this packet is equal to $\sum n(v_p[n] \neq v_p[x])(1 - \epsilon_{x,n})$. Determine all possible transmitted packets with their corresponding transmitters and calculate their average number of successful receivers.

9: Consider packet $p$ with its corresponding vector $v_p$ in $M_d$; any device that has all uncoded packets in $v_p$ can transmit $p$. The average number of successful receivers for packet $p \in M_d$ to be transmitted from device $x \in \mathcal{N}$ is equal to $\sum n(v_p[n] \neq NULL)(1 - \epsilon_{x,n})$. Determine all possible transmitted packets with their corresponding transmitters and calculate their average number of successful receivers.

10: From all packets determined in 8 and 9, select the packet with the maximum average number of successful receivers to be transmitted from its corresponding transmitter through D2D links.

$v_p[x]$ is the only uncoded packet in $v_p$ that is not available in the Has set of $x$. With that being said, $x$ can transmit the partial of packet $p$ which is constructed by network coding all packets in the vector $v_p \setminus v_p[x]$ with the targeted receivers $\mathcal{N}_t = \{i \mid v_p[i] \neq v_p[x]\}$. Therefore, the average number of successful receivers for the partial packet transmitted from $x$ is equal to $\sum n(v_p[n] \neq v_p[x])(1 - \epsilon_{x,n})$. We determine all possible transmitted packets from the set $M_l$ along with their corresponding transmitters and calculate their average number of successful receivers (line 9). At last, we select the packet with the maximum average number of successful receivers among all packets from the sets $M_l$ and $M_d$ to be transmitted from its corresponding transmitter through D2D links (line 10).

b) Upper bound on $T$:

In order to characterize the performance of NCMI-Instant, we develop the following upper bound for the packet completion time obtained from NCMI-Instant for the lossy channels in the second stage.

Theorem 8: The average of packet completion time; $T$ when NCMI-Instant is employed by cooperative mobile devices in a joint cellular and D2D setup when the channel links are lossy is upper bounded by

$$T \leq \max \left( \frac{T_{s,c} + (T_{l,d} + T_{l,t})}{T_{l,d} + T_{l,t} + T_{s,d} + T_{s,t}} \right),$$

where $M_c$, $M_l$, and $M_d$ are the sets that are constructed by Algorithm 1 for the first transmission slot and $T_{s,c}$, $T_{s,d}$, and $T_{s,t}$ are the average packet completion times for the source to transmit the packets in the sets $M_c$, $M_l$, and $M_d$, respectively. $T_{l,d}$ and $T_{l,t}$ are the average packet completion times for the cooperating devices in the local area to transmit the packets in the sets $M_l$ and $M_d$, as calculated in the following:

$$T_{s,c} = \frac{|M_c|}{1 - \max_{n \in \mathcal{N}} \eta_n},$$

$$T_{s,t} = \frac{|M_l|}{1 - \max_{n \in \mathcal{N}} \eta_n},$$

$$T_{s,d} = \sum_{p \in M_d} \frac{1}{1 - \max_{n \in \mathcal{N}} \eta_n},$$

where $v_p$ is the vector associated with the network coded packet $p$.

$$T_{l,d} = \sum_{p \in M_l} \left( \frac{1 - \max_{n \in \mathcal{N}} \epsilon_{x,n}}{n(v_p[n] \neq v_p[x])} + \frac{1 - \max_{n \in \mathcal{N}} \epsilon_{x,n}}{n(v_p[n] = v_p[x])} \right),$$

where $x$ is the device to be selected to transmit the partial of packet $p \in M_l$. According to Algorithm 3, $x$ is selected such that the average number of successful receivers is maximized; $x = \arg \max_{x \in \mathcal{N}} \sum n(v_p[n] \neq v_p[i])(1 - \epsilon_{i,n})$. $x'$ is the device to be selected to transmit the residual of packet $p$, which includes the uncoded packet $v_p[x]$; $x'$ is selected such that the average number of successful receivers is maximized; $x' = \arg \max_{x \in \mathcal{N}} \sum n(v_p[n] = v_p[x])(1 - \epsilon_{i,n})$.

$$T_{l,t} = \sum_{p \in M_d} \frac{1}{1 - \max_{n \in \mathcal{N}} \epsilon_{x,n}},$$

where $x$ is the device to be selected to transmit packet $p \in M_d$. According to Algorithm 3, $x$ is selected such that the average number of successful receivers is maximized; $x = \arg \max_{x \in \mathcal{N}} \sum n(v_p[n] \neq NULL)(1 - \epsilon_{i,n})$. Proof: The proof is provided in the Supplementary Material.
c) Computational complexity: 

**Lemma 9**: The complexity of NCMI-Instant, when the channels are lossy in the second stage, is polynomial with the complexity of $O(M^3 + N^3)$.

**Proof**: The proof is provided in the Supplementary Material.

IV. LOWER BOUND ON $T$

In this section, we develop a lower bound on the packet completion time when any network coding algorithm is employed by cooperative mobile devices in a joint cellular and D2D setup.\(^4\) The effectiveness of a network coding algorithm is evaluated by comparing the packet completion time obtained from the algorithm with the lower bound; the closer the packet completion time obtained from a network coding algorithm is to the lower bound, the more effective is the algorithm.

**Theorem 10**: The packet completion time when network coding is employed by cooperative mobile devices in a joint cellular and D2D setup is lower bounded by:

$$T \geq \max\left(\frac{|M_c|}{1 - \prod_{n \in N} \eta_n}, \max_{n \in N} \left(\min_{x \in (N \setminus n)} 2 - \eta_n - \epsilon_{x,n}\right)\right). \quad (15)$$

**Proof**: The proof is provided in the Supplementary Material.

**Corollary 11**: The packet completion time when network coding is employed by cooperative mobile devices in a joint cellular and D2D setup when the channel links are lossless in the second stage, is lower bounded by:

$$T \geq \max\left(\frac{|M_c|}{2 \max_{n \in N} |W_n|}\right). \quad (16)$$

**Proof**: By substituting $\eta_n = \epsilon_{k,l} = 0, \forall n, k, l \in N$ in (15), the lower bound provided in (16) is obtained.

As shown in the simulation results, our proposed NCMI-Batch and NCMI-Instant methods perform closer to the lower bound as compared to the baselines.

V. SIMULATION RESULTS

We considered a topology shown in Fig. 1(b) for different number of devices, packets, and loss probabilities. Then we implemented our proposed methods, NCMI-Batch and NCMI-Instant for this topology and compared their performances with the lower bound and their upper bounds as well as the baselines of, (i) NoNC, which stands for No Network Coding scheme and (ii) NCSI, which stands for Network Coding for Single-Interface systems. Each simulated point, is the average of results over 500 iterations. In our simulation results, bounds are plotted using dashed lines, while the real simulation results are plotted using the solid lines. We first consider the case where the channels are lossless in the second stage. Then, we present the simulation results for lossy channels in the second stage. Finally, we investigate the effect of subpacketization on both lossless and lossy NCMI.

\(^4\)Note that the provided lower bound is not guaranteed to be achievable. However, it is guaranteed that there is no other scheme to perform better than the lower bound and thus it is a good metric to evaluate the performance of proposed NCMI methods.
Fig. 5. Packet completion time vs. (a) number of devices, and (b) loss probabilities of cellular links in stage one, for NCMI-Instant and NCMI-Batch as compared to the lower bound and their upper bounds as well as baselines, when the channels are lossless in stage two.

Fig. 6. Scalability of NCMI-Instant and NCMI-Batch with the network size as compared to the lower bound and their upper bounds as well as baselines, when the channels are lossless in stage two.

Packet Completion Time vs. Number of Devices: Fig. 5(a) shows the packet completion time for different number of devices and $M = 50$ packets. As shown in the figure, NCMI-Batch performs very close to the lower bound and upper bounds, and better than the single-interface systems and No-NC. NCMI-Instant performs better than single-interface systems and No-NC and worse than NCMI-Batch. The performance of NCMI-Instant gets closer to No-NC method for the larger number of devices. The reason is that the network coding opportunities of NCMI-Instant decrease in this lossless scenario when the number of devices increases, so NCMI-Instant gets closer to NoNC.

1) Packet Completion Time vs. Loss Probability: Fig. 5(b) presents the packet completion time for different loss probabilities in stage one for $M = 20$ packets and $N = 5$ devices. In this setup, the loss probabilities of cellular links are the same for all mobile devices. As seen, the performances of NCMI schemes and the lower bound get closer to the no network coding scheme by increasing the channel losses of cellular links. In other words, when the channel losses are large, the no network coding scheme (with lower complexity than network coding schemes) has the performance close to the optimal scheme, which is defined by the lower bound.

Scalability With the Network Size: Fig. 6 shows the packet completion time for different network sizes (number of devices) with $M = 500$ packets. As seen, NCMI-Batch scales well with increasing the network size, but NCMI-Instant performs close to No-NC scheme for larger networks and fixed number of packets. In order for NCMI-Instant to perform close to its best performance, the number of packets should also be increased by increasing the size of the network. The reason for this observation is that network coding opportunities of NCMI-Instant decrease with increasing number of devices in lossless scenario (lossless in stage two). This holds for a general setup as explained next.

Let us assume that we would like to create an IDNC packet \( p_m + p_k \). The condition to create this IDNC packet is that all devices that want \( p_m \) should have packet \( p_k \), and all devices that want \( p_k \) should have \( p_m \). The probability of this condition is \( \prod_{n \in N_{\epsilon}} (1 - \eta_n) \), where \( N_{\epsilon} \) is the set of devices that want packet \( p_m \) or \( p_k \) and \( \eta_n \) is the loss probability at stage one. As seen, the probability of creating an IDNC packet, \( i.e., \prod_{n \in N_{\epsilon}} (1 - \eta_n) \) decreases with increasing \( |N_{\epsilon}| \).

B. Lossy Channels in Stage Two

In this section, we consider a setup in which the packet transmissions in both stages are lossy. In stage one, all packets are transmitted from the source; each transmitted packet is lost with the loss probability of \( \eta_n \) at each device \( n \). In stage two, two packets are broadcast simultaneously at each transmission slot; one packet is broadcast from the source and is lost with the loss probability of \( \epsilon_n \) at each device \( n \) and the other packet is broadcast from the transmitter device \( t \) to all other devices. (vi) NCMI-Batch, via D2D Links, which uses instantly decodable network coding via the single interface of D2D links. At each transmission slot, the device with the maximum size of \( Has \) set is selected as the transmitter and broadcast a random combination of all packets in its \( Has \) set to all other devices. (vii) NCMI-Instant, via Cellular Links, which uses instantly decodable network coding via the single interface of Cellular links. At each transmission slot, a random combination of all packets is broadcast to all devices that want this combination. (viii) NCMI-Instant, via Multiple D2D Links, which uses instantly decodable network coding via the single interface of D2D links. At each transmission slot, each interface selects a missing packet with the maximum average number of receivers to be broadcast to all devices. (iv) NCMI-Instant, via D2D Links, which uses instantly decodable network coding via the single interface of D2D links. At each transmission slot, each interface selects a missing packet with the maximum average number of receivers to be broadcast to all devices. (v) NCMI-Instant, via Cellular Links, which uses instantly decodable network coding via the single interface of Cellular links. At each transmission slot, each interface selects a missing packet with the maximum average number of receivers to be broadcast to all devices. (vii) NCMI-Instant, via Cellular Links, which uses instantly decodable network coding via the single interface of Cellular links. At each transmission slot, the union of the missing packets in all devices is grouped into \( M_{\epsilon}, M_t, \) and \( M_d \) sets according to Algorithm 1. Then, if \( M_{\epsilon} \) is not empty, the packets in this set are requested to be transmitted from the source until they are received successfully by at least one of the devices. If \( M_{\epsilon} \) is empty, the order of transmitting the packets is from the sets \( M_d \) and \( M_t \); at each transmission slot, the packet with the maximum average number of successful receivers is selected among all packets of the same set. (vi) NCMI-Instant, via D2D Links, which uses instantly decodable network coding via the single interface of D2D links. At each transmission slot, a random combination of all packets is broadcast to all devices that want this combination. (vii) NCMI-Instant, via Cellular Links, which uses instantly decodable network coding via the single interface of Cellular links. At each transmission slot, the union of the missing packets in all devices is grouped into \( M_{\epsilon}, M_t, \) and \( M_d \) sets according to Algorithm 1. The order of transmitting the packets is from the sets \( M_{\epsilon}, M_t, \) and \( M_d \).
the packet completion time significantly as compared to the single-interface systems and No-NC. Note that NCMI-Instant significantly improves packet completion time over NoNC when the number of devices increases. This is different than what we observed in Fig. 5(a) (where NCMI-Instant gets closer to NoNC with increasing number of devices). The reason is that network coding opportunities of NCMI-Instant still exist in this scenario due to losses in stage two. Also, network coding schemes using the single interface of D2D links outperform the network coding schemes using single interface of cellular links, despite the fact that more various network coded packet can be transmitted from the cellular link than the D2D link. The reason is that the packets to satisfy each device n using D2D links can be transmitted from a variety of channel links with the loss probabilities of \( \epsilon_{t,n}, t \in (N \setminus n) \) from which the most reliable link is selected. While the packets to satisfy each device n using the cellular links can only be transmitted from the cellular link with the fixed loss probability of \( \eta_n \) and thus if \( \eta_n \) is high, there is no other more reliable link to be selected. Finally, the performance of NCMI-Instant and NCMI-Batch are close to the lower bound and their upper bounds.

C. Effect of Subpacketization on Lossless and Lossy NCMI

We investigated the impact of subpacketization for a file with the size of \( M = 100 \) packets and \( N = 5 \) cooperating devices, and showed the results in Fig. 8. In this figure, a file with fixed size of \( M = 100 \) packets is transmitted for all simulated points, and the packet completion time is measured for this file. In particular, Fig. 8 presents the packet completion time of the file with \( M = 100 \) packets when this file is divided into different sized subfiles. For example, in Fig. 8 (b), the packet completion time of NCMI is close to 30 when subfile size is 10. This means that (i) the file of \( M = 100 \) packets is divided into 10 subfiles, each with 10 packets, (ii) each subfile is coded using NCMI, and (iii) the total delay of 10 subfiles (i.e., the whole file with \( M = 100 \) packets) is reported. The same argument holds for all of the simulated points in Fig. 8.

As seen in the figure, the packet completion time decreases with increasing the size of subfile. This means that it would be more efficient to apply proposed methods on the whole file instead of dividing it into subfiles and apply the proposed methods on each subfile. As seen, our NCMI algorithms still significantly improve the packet completion time as compared to the baselines.

VI. RELATED WORK

Network Coding for Single-Interface Systems

The performance of network coding has been evaluated for single-interface systems in literature. The problem of minimizing the number of broadcast transmissions required to satisfy all devices is considered in [17], and the bounds for packet completion time are developed. A deterministic linear network coding algorithm that minimizes the number of broadcast transmissions is considered in [22]. Minimization of the completion delay while broadcasting instantly decodable network coding packets has been considered in [25]. The problem of recovering the missing content using cooperative data exchange utilizing local area connections is considered in [18] and [19], and the lower and upper bounds on the minimum number of transmissions are developed. Deterministic algorithms for the cooperative data exchange problem with polynomial time are designed in [26] and [20]. In our previous work, [27], we considered data exchange problem for multimedia applications and proposed content-aware network coding schemes that improves content quality and reduces the packet completion time using D2D links only. As compared to [27], we consider cooperative mobile devices in the joint cellular and D2D setup, and develop a network coding scheme for this setup in this paper.

Multiple-Interface Systems

The performance of WiFi-only, cellular-only, and multiple-interface (WiFi plus cellular) systems are studied and compared in [3]. A flexible software architecture is developed in [4] by adaptively selecting available interfaces at mobile devices with the goal of improving Quality of Experience (QoE) while minimizing the energy overhead. The heterogeneity of cellular and Wi-Fi interfaces are effectively utilized to deliver data to mobile devices in [5]. Cooperative content delivery to mobile devices is developed in [6] by taking into account the link quality of both cellular and WiFi interfaces, where a device with the best link quality is preferred in the cooperative system. The scenario of delivering same
content to a group of users by using unicast cellular and D2D links is considered in [7]. A comprehensive survey on exploiting multiple available wireless interfaces to deliver content to cooperative mobile devices is provided in [8]. This line of work demonstrates the practicality of simultaneous operation of multiple interfaces including cellular and short-range D2D links. As compared to this work, we consider how efficient network coding algorithms can be developed with provable performance guarantees for cooperative mobile devices in the joint broadcast cellular and broadcast D2D setup.

**Network Coding for Multiple-Interface Systems**

Network coding has been employed in the previous work for devices with multiple interfaces. Wireless video broadcasting with P2P error recovery is proposed by Li and Chan [28]. An efficient scheduling approach with network coding for wireless local repair is introduced by Saleh et al. [29]. Another body of work [30]–[32] proposes systems where there are a base station broadcasting packets and a group of smartphone users helping each other to correct errors. Compared to prior work [28]–[32], where each phone downloads all the data, and D2D links are used for error recovery, our scheme jointly utilizes cellular and D2D links and analyzes the performance of network coding in such a setup.

Simultaneous operation of multiple interfaces and employing network coding for this setup has also been considered in the previous work; [9], [10], [33]. As compared to [9] and [10], we consider broadcast cellular links simultaneously with D2D links for error recovery purposes, while [9] and [10] use unicast cellular links simultaneously with D2D links for throughput improvement purposes. Multimedia streaming to a single user is considered in [25], where multiple interfaces are used at the single user. As compared to this work, we use cooperation among mobile devices that benefit D2D links in conjunction with cellular links. Also, in this paper, we consider how efficient network coding algorithms can be developed with provable performance guarantees for cooperative mobile devices in the joint cellular and D2D setup, instead of using existing network coding algorithms. In [34], the conference version of this work, we developed network coding schemes for cooperative mobile devices in the joint cellular and D2D setup, where we assumed the data transmissions are lossless. As compared to [34], in this paper, we consider lossy channels and propose network coding schemes by taking into account the probabilities of channel losses as well.

**VII. Conclusion**

In this paper, we considered a scenario where a group of mobile devices is interested in the same content, but each device has a partial content due to packet losses over cellular links. In this setup, mobile devices cooperate and exploit the cellular and D2D links jointly to recover the missing content. We developed network coding schemes; NCMI-Batch and NCMI-Instant for this setup, and analyzed their packet completion time. Simulation results confirm that NCMI-Batch and NCMI-Instant significantly reduce the packet completion time.

**References**

[1] Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018, Cisco, San Francisco, CA, USA, 2018.
[2] Ericsson Mobility Report, Ericsson, Stockholm, Sweden, Aug. 2014.
[3] S. Deng, R. Netvatzali, A. Sivaraman, and H. Balakrishnan, “WiFi, LTE, or Both?: Measuring multi-homed wireless Internet performance,” in *Proc. ACM IMC*, Vancouver, BC, Canada, Nov. 2014, pp. 181–194.
[4] A. Nikravesh, Y. Guo, F. Qian, Z. M. Mao, and S. Sen, “An in-depth understanding of multipath TCP on mobile devices: Measurement and system design,” in *Proc. ACM MobiCom*, New York, NY, USA, Oct. 2016, pp. 189–201.
[5] C.-L. Tsao and R. Shokoumik, “On effectively exploiting multiple wireless interfaces in mobile hosts,” in *Proc. CoNEXT*, Rome, Italy, Dec. 2009, pp. 337–348.
[6] B. Barua, Z. Khan, Z. Han, A. A. Abouzeid, and M. Latva-Aho, “Incentivizing selected devices to perform cooperative content delivery: A carrier aggregation-based approach,” *IEEE Trans. Wireless Commun.*, vol. 15, no. 7, pp. 5030–5045, Jul. 2016.
[7] M. Stiemerling and S. Kiesel, “A system for peer-to-peer video streaming in resource constrained mobile environments,” in *Proc. ACM Workshop User-Provided Netw., Challenges Opportunities (U-NET)*, Rome, Italy, Dec. 2009, pp. 25–30.
[8] Z. Khan, A. V. Vasilakos, B. Barua, S. Shahabuddin, and H. Ahmadi, “Cooperative content delivery exploiting multiple wireless interfaces: Methods, new technological developments, open research issues and a case study,” *Wireless Netw.*, vol. 22, no. 6, pp. 1961–1983, Aug. 2016.
[9] L. Keller et al., “MicroCast: Cooperative video streaming on smartphones,” in *Proc. ACM MobiSys*, Lake District, U.K., Jun. 2012, pp. 57–70.
[10] H. Seferoglu, L. Keller, B. Cici, A. Le, and A. Markopoulou, “Cooperative video streaming on smartphones,” in *Proc. Allerton*, Sep. 2011, pp. 220–227.
[11] R. Ahlsvede, 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. Katti et al., “XORs in the air: Practical wireless network coding,” *IEEE/ACM Trans. Netw.*, vol. 16, no. 3., pp. 497–510, Jun. 2008.
[13] 3GPP Release 8. Accessed: 2008. [Online]. Available: http://www.3gpp.org/technologies/keywods-actnomcs/98-t-le
[14] LTE Broadcast White Paper. Accessed: 2013. [Online]. Available: http://www.ericsson.com/portaritl/medlalnustry/lte-broadcast
[15] LTE Broadcast. Accessed: 2014. [Online]. Available: https://www.qualcomm.com/products/lte/broadcast
[16] C.-C. Wang, “On the capacity of 1-to-K broadcast packet erasure channels with channel output feedback,” *IEEE Trans. Inf. Theory*, vol. 58, no. 2, pp. 931–956, Feb. 2012.
[17] S. Y. El Rouayheb, M. A. R. Chaudhry, and A. Sprintson, “On the minimum number of transmissions in single-hop wireless coding networks,” in *Proc. IEEE ITW*, Tahoe City, CA, USA, Sep. 2007, pp. 120–125.
[18] S. El Rouayheb, A. Sprintson, and P. Sadeghi, “On coding for cooperative data exchange,” in *Proc. IEEE ITW*, Cairo, Egypt, Jan. 2010, pp. 1–5.
[19] A. Sprintson, P. Sadeghi, G. Booker, and S. El Rouayheb, “A randomized algorithm and performance bounds for coded cooperative data exchange,” in *Proc. IEEE ISIT*, Austin, TX, USA, Jun. 2010, pp. 1888–1902.
[20] A. Sprintson, P. Sadeghi, G. Booker, and S. Y. El Rouayheb, “Deterministic algorithm for coded cooperative data exchange,” in *Proc. QShine*, Houston, TX, USA, Nov. 2010, pp. 282–289.
[21] S. E. Tajbakhsh, P. Sadeghi, and N. Aboufotarab, “Instantly decodable network codes for cooperative index coding problem over general topologies,” in *Proc. AusCTW*, Sydney, NSW, Australia, Feb. 2014, pp. 84–89.
[22] M. Yu, P. Sadeghi, and N. Aboufotarab, “On deterministic linear network coded broadcast and its relation to matroid theory,” in *Proc. IEEE ITW*, Hobart, TAS, Australia, Nov. 2014, pp. 536–540.
[23] G. P. Perrucci, F. H. P. Fitzek, and J. Widmer, “Survey on energy consumption entities on the smartphone platform,” in *Proc. IEEE 73rd Veh. Technol. Conf. (VTC Spring)*, Yokohama, Japan, May 2011, pp. 1–6.
[24] Y. Geng, W. Hu, Y. Yang, W. Gao, and G. Cao, “Energy-efficient computation offloading in cellular networks,” in *Proc. IEEE ICNP*, San Francisco, CA, USA, Nov. 2015, pp. 145–155.
[25] S. Sorour and S. Valae, “On minimizing broadcast completion delay for instantly decodable network coding,” in *Proc. ICC*, May 2010, pp. 1–5.
[26] N. Milosevic, S. Pawar, S. El Rouayheb, M. Gastpar, and K. Ramchandran, “Deterministic algorithm for the cooperative data exchange problem,” in *Proc. IEEE ISIT*, Saint Petersburg, Russia, Jul/Aug. 2011, pp. 410–414.
Yasaman Keshtkarjahromi received the B.S. degree in electrical and computer engineering from Tehran University, in 2010. She was a Summer Intern at Huawei Research and Development and Alcatel-Lucent Bell Labs in 2016 and 2015, respectively. She is currently pursuing the Ph.D. degree with the Electrical and Computer Engineering Department, University of Illinois at Chicago.

Hulya Seferoglu (S’04–M’11) received the B.S. degree in electrical engineering from Istanbul University, Turkey, in 2003, the M.S. degree in electrical engineering and computer science from Sabanci University, Turkey, in 2007, and the Ph.D. degree in electrical and computer engineering from the University of California at Irvine, Irvine, CA, USA, in 2010. She was a Post-Doctoral Associate with the Laboratory of Information and Decision Systems, Massachusetts Institute of Technology, from 2011 to 2013. She was a Summer Intern at AT&T Labs Research, DoCoMo USA Labs, and Microsoft Research Cambridge in 2010, 2008, and 2007, respectively. She is currently an Assistant Professor with the Electrical and Computer Engineering Department, University of Illinois at Chicago. She serves as an Executive Editor for the Transactions on Emerging Telecommunications Technologies from 2016 to 2019 and an Associate Editor for the IEEE/ACM Transactions on Networking from 2017 to 2019.

Her research interests are in the area of networking: design, analysis, and optimization of network protocols and algorithms. She is particularly interested in network optimization, network coding, and multimedia streaming.

Ashfaq Khokhar (F’99) received the B.S. degree in electrical engineering from the University of Engineering and Technology at Lahore, Pakistan, in 1985, the M.S. degree in computer engineering from Syracuse University in 1989, and the Ph.D. degree in computer engineering from the University of Southern California in 1993. He joined the Illinois Institute of Technology, University of Illinois at Chicago (UIC), where he was a Professor and the Director of Graduate Studies in the Department of Electrical and Computer Engineering. He has been a Professor and the Chair of the Department of Electrical and Computer Engineering since 2013. He also holds adjunct appointments in the Department of Computer Science and the Department of Biomedical and Health Information Sciences, UIC. He is also the Chief Technology Officer at Video Analytics, Inc., Woodridge, IL, USA, a startup company he founded that develops scalable and computationally efficient solutions for multimedia applications, including smart and intelligent video surveillance technologies. He has contributed to five edited volumes and coauthored nine book chapters, 55 publications in archival journals, and 158 refereed conference papers. His research is currently supported by multiple awards funded by the National Science Foundation and the National Institutes of Health. He has received numerous scholarly recognitions and awards, including the NSF CAREER Award in 1998 for his work on multi-threaded algorithms for multimedia applications. He has served as an Associate Editor on the Editorial Board of numerous journals. He was an elected IEEE fellow for his work on multimedia computing and databases in 2009.

Rashid Ansari (F’99) received the Ph.D. degree in electrical engineering and computer science from Princeton University, Princeton, NJ, USA, and the B.Tech. and M.Tech. degrees in electrical engineering from IIT, Kanpur, India. He served as a Research Scientist at Bell Communications Research and on the Faculty of Electrical Engineering, University of Pennsylvania. He is currently a Professor and the Head of the Department of Electrical and Computer Engineering, University of Illinois at Chicago. His research interests are in the general areas of signal, image, and video processing and communication, with recent focus on image and video analysis, multimedia communication, and medical imaging. He has served as a member of the Digital Signal Processing Technical Committee, IEEE Circuits and Systems Society, and a member of the Image, Video, and Multidimensional Signal Processing Technical Committee, IEEE Signal Processing Society. He was a member of the Editorial Board of the Journal of Visual Communication and Image Representation. He was an elected IEEE fellow for contributions to multirate digital signal processing and digital filter banks in 1999. He served on the organizing and executive committees of the SPIE Visual Communication and Image Processing conferences. He has served several times on the technical program committees of the IEEE International Conference on Image Processing and the IEEE/SPIE Visual Communications and Image Processing (VCIP) conferences, and was the Conference Chair of the IEEE/SPIE VCIP Conference in 1996. He has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS.

Ashfaq Khokhar (F’99) received the B.S. degree in electrical engineering from the University of Engineering and Technology at Lahore, Pakistan, in 1985, the M.S. degree in computer engineering from Syracuse University in 1989, and the Ph.D. degree in computer engineering from the University of Southern California in 1993. He joined the Illinois Institute of Technology, University of Illinois at Chicago (UIC), where he was a Professor and the Director of Graduate Studies in the Department of Electrical and Computer Engineering. He has been a Professor and the Chair of the Department of Electrical and Computer Engineering since 2013. He also holds adjunct appointments in the Department of Computer Science and the Department of Biomedical and Health Information Sciences, UIC. He is also the Chief Technology Officer at Video Analytics, Inc., Woodridge, IL, USA, a startup company he founded that develops scalable and computationally efficient solutions for multimedia applications, including smart and intelligent video surveillance technologies. He has contributed to five edited volumes and coauthored nine book chapters, 55 publications in archival journals, and 158 refereed conference papers. His research is currently supported by multiple awards funded by the National Science Foundation and the National Institutes of Health. He has received numerous scholarly recognitions and awards, including the NSF CAREER Award in 1998 for his work on multi-threaded algorithms for multimedia applications. He has served as an Associate Editor on the Editorial Board of numerous journals. He was an elected IEEE fellow for his work on multimedia computing and databases in 2009.