Received December 26, 2018, accepted January 9, 2019, date of publication January 23, 2019, date of current version February 22, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2894513

Modeling the Delivery of Coded Packets in D2D Mobile Caching Networks

AMEER AHMED[1,2,3], HANGGUAN SHAN[1,2], (Member, IEEE), AND AIPING HUANG[1,2], (Senior Member, IEEE)

[1]College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China
[2]Zhejiang Provincial Key Laboratory of Information Processing and Communication Networks, Hangzhou 310027, China
[3]Faculty of Information and Technology, COMSATS University Islamabad at Sahiwal Campus, Sahiwal 57000, Pakistan

Corresponding author: Hangguan Shan (hshan@zju.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771427 and Grant U1709214, and in part by the Zhejiang Provincial Public Technology Research of China under Grant 2016C31063.

A conference version of this work is under review by Proc. IEEE ICC 2019 [1].

ABSTRACT Caching popular files on the mobile nodes have been seen as a promising solution to improve the network performance. In this paper, we analyze the delivery of coded packets of the requested file in a mobile caching network of nodes each with a few pre-cached packets of different files from a large file library. To model the packet delivery process, we develop a two-dimensional Markov chain framework where each request has a deadline requirement. For the delivery of packets, we consider the mobility aspect of caching network where the nodes request the files from other neighbors while moving among different cells and offload the requested files on the device-to-device link. To characterize the network performance, we derive closed-form expressions for file outage probability, the average delay in receiving file, and average throughput to provide some valuable insights on different parameter settings. The simulation results not only verify the accuracy of the analysis but also shed some light on the scenarios where the performance gap between the mobile and fixed caching networks is prominent.

INDEX TERMS Mobile caching network, asynchronous request, D2D communication, TDMA spatial reuse, coded caching.

I. INTRODUCTION

Due to technological advancements in smart mobile devices, a shift in mobile users has been witnessed from connection-centric communications (simple voice or text) to information or content-centric communications in recent years [2]. The use of multimedia services has exponentially increased the wireless data traffic and is expected to grow even more in next few years [3], [4]. As studied in [5]–[8], most of the wireless traffic these days is due to asynchronous requests where different users request the same content at different times from the central server. Undoubtedly, conventional methods like increasing spectral resources, network densification, and increasing spectral efficiency by adopting better modulation, coding, and multi-antenna techniques have been adopted in 4G and 5G wireless systems to enhance wireless link capacity; however, these methods are either not cost effective to implement or are not efficient enough to boost network capacity due to attaining the theoretical upper-bounds [9].

In recent years, different solutions have been proposed to reduce the number of transmissions of the same content from the central server by caching the popular contents at the nearby base station (BS)/helper nodes and at the terminal nodes. Therefore, caching can be one of the most promising solutions to boost the capacity of future wireless networks, which has been widely used in wired networks for content distribution [10], [11]. Because it not only helps to increase the network throughput but also reduces the load on the backhaul network and can contribute towards setting up a green communication network [12] by reducing the energy consumption and improving the energy efficiency of devices [13]–[15]. Moreover, caching also improves the performance of Internet of Things (IoT) devices for real-time data processing using smart prefetching of desired data before the arrival of IoT queries [16].

In caching networks, caching popular contents on wireless nodes takes place in two steps: placement phase and delivery phase [17]. Caching can be coded/uncoded [18], [19], at the nearby BS/helper nodes (femto-caching) [20]–[22] or at the terminal nodes [23]. Recently, device-to-device (D2D) communication has attracted the attention of researchers from both industry and academia and is expected to become a key enabler of the future wireless networks with possible advantages including data offloading at higher bit-rate, reducing battery usage, and robustness against network failures [24].

Node mobility is an important characteristic of wireless networks but little attention has been paid on the impact of mobility in caching networks; in contrast, node mobility can further improve the network capacity of a caching network and a thorough theoretical analysis is needed to understand the impact of node mobility in D2D mobile caching networks. A very first study on the impact of mobility was done in [25], where improvement in per-user throughput has been shown via packet relaying using multiuser diversity. The rationale behind conducting this study is to see the impact of node mobility on the performance of mobile caching networks. For this purpose, we also utilize the mobility aspect of wireless nodes to find the useful transmitters having the required contents by modeling each transmitting node as a relay, i.e., nodes can search the required contents in other cells while they move, provided sufficient packets of each file are cached by the nodes in the network.

A. RELATED WORKS

In the literature, caching in D2D networks has been addressed by different solutions. In [26], the framework of deploying helper nodes or small base stations (SBSs) with large storage capacity is presented with D2D content sharing. In [27], a simplified system model is presented to study the delivery of files under centralized and random caching strategies using D2D link. Wang *et al.* [28] demonstrate the importance of caching in next generation 5G wireless networks, present an idea of edge based caching scheme and discuss possible opportunities and challenges of caching at core network and at terminal nodes. Cooperation among the BS, the relays, and the terminal nodes for the content dissemination in heterogenous networks (HetNets) is investigated in [29], where some nodes have the file caching ability. In order to minimize the transmission cost, edge caching at the BSs with D2D offloading is discussed in [30]. Ji *et al.* [31] present an overview of different cache strategies and show significant improvement in performance using D2D caching. Ji *et al.* [32] discuss spatial reuse gain and coded multicast gain in D2D caching networks. The work in [33] presents the idea of proactive caching of popular contents and shows improvement in user satisfaction by leveraging D2D communication and social networks for content dissemination. In [34], authors consider a wireless D2D network and characterize the optimal tradeoff between throughput and the number of nodes in outage, which depends on the transmission range of the nodes. However, in these works, the impact of node mobility was ignored and a static network scenario is considered.

There are some studies which consider the impact of node mobility in caching networks. Alfano [35] derive the throughput-delay tradeoff under different mobility models and transmission ranges. A general framework of mobility-aware caching methodology in content-centric wireless networks has been proposed Wang *et al.* [36] exploit nodes' trajectory information to boost the caching efficiency using cooperation between the two BSs and improve the network performance. Krishnan and Dhillon [37] consider a library of two files, find coverage probability of partially cached file and show the positive impact of user mobility on D2D caching networks. The work in [38] presents increase in data offloading through simulations based on captured node mobility pattern using inter-contact model [39]. However, all the aforementioned studies assume that one contact duration is long enough to receive the whole content and ignore the fact that contact duration may not be enough for the reception of the complete file.

Besides, some works consider contact duration (which follows exponential distribution) while investigating the effect of node mobility. For example, the work in [40] studies optimization problem to minimize the cost of offloading data. However, the impact of node mobility is discussed through simulations only. In [41], optimization problem is formulated to improve data offloading ratio and show comparison among different caching strategies. Shabani *et al.* [42] achieve coded caching and spatial reuse gain simultaneously using node mobility. However, in these works, analysis on the joint impact of node mobility and transmission errors in D2D caching networks is missing.

B. CONTRIBUTIONS

In this work, we propose an analytical framework to study caching and node mobility in a D2D caching network. Specifically, we consider the cluster based wireless network where the nodes move according to independent and identically distributed (or reshuffling) mobility model, request other nodes in the cluster for their required files and receive them under varying successful transmission probabilities. We divide the time into slots and only one link is active in the cluster during each slot. Moreover, by using Markov chain, we analyze the reception of packets of requested file which has a deadline requirement and derive expressions of multiple performance metrics. The main contributions of this work are summarized as follows:

- We present a D2D mobile caching network where we divide a large file into packets and only a few coded packets of each file can be received in a slot from the nearby nodes in the cluster without creating any congestion on the backhaul link.
- We develop a two-dimensional (2-D) Markov chain framework to analyze the impact of node mobility and transmission errors in receiving packets of files. Using the Markov analysis, we then derive closed-form expressions for performance metrics including outage probability, delay, and throughput.
- We provide analysis of results on different parameter settings and simulation results verify our theoretical modeling. Further, we identify the scenarios in which node mobility helps to improve network performance as compared with the static node scenarios.

The remainder of this paper is organized as follows. System model is defined in Section II. Packet reception modeling is discussed in Section III. Expressions for performance metrics are derived in Section IV. Theoretical results are verified using simulations in Section V. Finally, concluding remarks are presented in Section VI.

II. SYSTEM MODEL

A. NETWORK DESCRIPTION

We consider a network consisting of N nodes belonging to $\mathcal{N} \triangleq \{n_1, n_2, \ldots, n_N\}$ deployed over a square region. The network is time slotted and the duration of each time slot is Δt. The network is divided into $M \times M$ clusters, each of random number of nodes, as shown in Fig. 1, where each square represents a cluster $S_c \triangleq \{\text{nodes in the cluster}\} \subseteq \mathcal{N}$. Each cluster acts like a cell and each cell is activated after K orthogonal time slots using TDMA spatial reuse, i.e., $K = \left(\left\lceil \sqrt{2}(1+\delta) \right\rceil + 1\right)^2$, with $p_{cluster}^{act} = 1/K$ being the cluster scheduling probability, where $\delta > 0$ is an interference control parameter [43]. Nodes can only communicate within the same cluster and only one transmitter in the cluster is active in a given time slot. Transmission range ($r = \sqrt{2}/M$) of each node is kept small for localized D2D communication with constant transmission power.

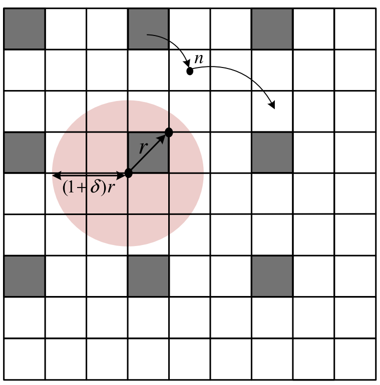

FIGURE 1. The network with cluster activation using TDMA reuse parameter $K = 9$ where each cluster is activated after every 9 time slots. In a given slot, simultaneously active clusters are represented by grey squares with no concurrent transmission inside the circular disk area.

B. INTERFERENCE CONTROLLED COMMUNICATION MODEL

To account for the interference among different nodes during simultaneous transmissions in the network, we assume that all nodes can communicate within a uniform transmission range r. Following the widely adopted protocol interference model [44], an interference free communication from node v to node u at time slot t is possible if their Euclidean distance $d_{u,v}(t)$ is not larger than the fixed transmission range r, i.e., $d_{u,v}(t) \leq r$, inside the cluster, and there is no other active transmitter w that simultaneously transmits with node v within the distance $(1+\delta)r$, i.e, $d_{v,w}(t) \geq (1+\delta)r$.

C. MOBILITY MODEL

In this work, similar to [45]–[47] we focus on the widely adopted bi-dimensional independent and identically distributed (i.i.d) mobility model (also called reshuffling model) to describe the random mobility of nodes. With i.i.d mobility model the number of nodes in the cluster changes rapidly because their positions in the network are reshuffled after each time slot. At the beginning of a time slot, each node takes independent decision to uniformly select a destination cluster among all the M^2 clusters with equal probability and stays there for the duration of whole time slot.

D. FILE LIBRARY AND CONTENT REPLICATION

We consider a large file library $\mathcal{F}$ of F files. Assume that each file i is formed by a sequence of L packets denoted by $\mathbf{P}_i$. Each node has a limited cache capacity of $H(\geq L)$ packets for voluntarily storing different packets of different files from $\mathcal{F}$. To transfer these files from $\mathcal{F}$ to individual nodes, a two-step procedure is followed.

1) CACHE PLACEMENT WITH LINEAR NETWORK CODING (LNC)

In the first step, packets of each file are coded together by LNC [48]–[50] over a finite field $\mathbb{F}_q$. Network encoded packets $\mathbf{N_{CP}} = \left[\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_{\bar{J}_i}\right]^T$ of file i are generated by precoding $\mathbf{P}_i$ of dimension $1 \times L$ with the encoding coefficient matrix $\mathbf{C_M}$ of dimension $L \times \bar{J}_i$ over a finite field $\mathbb{F}_q^{L \times \bar{J}_i}$, where $\hat{P}_1, \hat{P}_2, \ldots, \hat{P}_{\bar{J}_i}$ are the $\bar{J}_i(\gg L)$ encoded packets of file i and $[.]^T$ is the transpose of the row vector of encoded packets. Each node caches different network coded packets of each file. The number of coded packets of file i cached by each node, denoted by J_i, is decided by $J_i = \max\{\lfloor H \times p_c(i) \rfloor, 1\}$, where $p_c(i)$ is the probability of caching file i [27], satisfying

$$p_c(i) = \frac{i^{-\gamma_c}}{\sum_{j=1}^{F} j^{-\gamma_c}}, \quad i = 1, 2, \ldots, F \tag{1}$$

where γ_c is Zipf distribution exponent of file popularity. The set of $J_i(\ll \bar{J}_i)$ packets stored by node n is defined as $S_{n,i} \triangleq \{\hat{P}_{i,1}, \hat{P}_{i,2}, \ldots, \hat{P}_{i,J_i}\}$. To cache packets of every file, we consider that $H >> F$. Otherwise, some files will not be cached by the nodes and can only be requested from the BS.

2) DATA DELIVERY IN MOBILE D2D COMMUNICATIONS

For any partially cached file i, each node requests other nodes in the cluster and gets the remaining packets from its neighbors. This is referred to as delivery of missing contents to a requesting node (say receiver n). We assume that in the beginning of every slot, all the other nodes in the same cluster are aware of the request made by the node. Moreover, each request has a deadline and follows a popularity distribution. If $p_r(i)$ is the probability of requesting file i, then popularity of file i, considering Zipf's law for content popularity distribution [31], [51], can be found as

$$p_r(i) = \frac{i^{-\gamma_r}}{\sum_{j=1}^{F} j^{-\gamma_r}}, \quad i = 1, 2, \ldots, F \tag{2}$$

where γ_r is Zipf's law exponent with bound $0 < \gamma_r < 1$. A larger γ_r exponent corresponds to a higher content reuse, i.e., the first few popular files correspond to the majority of requests.

As the time goes, the node gets more and more encoded packets of the requested file from the nearby nodes. If at time t, node n gets $j_i(t) \in [0, L - J_i]$ packets from other nodes, let $s_{n,i} \triangleq \{P'_{i,1}, P'_{i,2}, \ldots, P'_{i,j_i(t)}\}$, denote the set of these received packets. Here, $\hat{P}_{i,j} \neq P'_{i,j'}$, $\forall j, j'$, and $L = J_i + j_i(t) + L_R(t)$, with $L_R(t)$ representing the number of remaining packets of file i required by the node.[1] If the total number of received encoded packets from the BS and from other nodes are at least equal to L at the receiver, i.e., $J_i + j_i \geq L$, then we can recover the whole file by decoding the encoded packets. If node n does not receive enough packets to decode file i until the deadline then a deadlock occurs. This deadlock is called an outage.

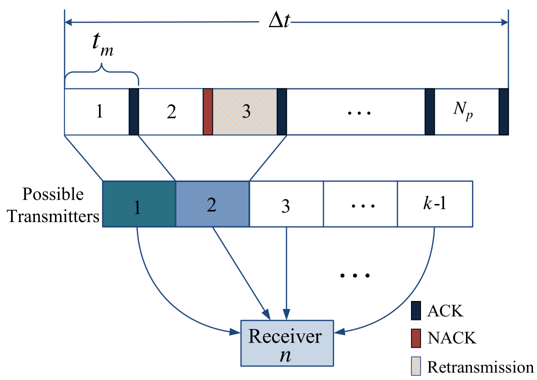

FIGURE 2. Division of time slot into N_p mini-slots for packet transmission from $k - 1$ possible transmitters to receiver n.

E. USER SCHEDULING

Since any node can generate a request for the required missing packets of any file, any node can be scheduled randomly to receive the packets from the other nodes in the cluster with probability $1/k$, assuming k nodes in the cluster. We divide the whole time slot Δt into N_p number of mini-slots, each of duration t_m, i.e., $N_p = \Delta t / t_m$, where t_m represents the time to transmit one packet once, as shown in Fig. 2. Each mini-slot t_m is further divided into two parts: packet transmission and its feedback. If the transmitter receives an acknowledgement (ACK), showing the successful transmission of packet in a mini-slot t_m, then it can transmit the new packet if it has more packets to transmit. However, if it receives a negative acknowledgement (NACK), then it retransmits the unsuccessful packet and it will keep retransmitting the failed packet until it receives the ACK of that packet or until time slot Δt expires. For any transmission, we consider that it succeeds with the same probability p_s.

In the beginning, all the nodes have different packets of a file i and the number of new packets in the cluster needed by node n of file i can be $n_{new} \in \{0, 1, 2, \ldots, (k-1) \times J_i\}$, where k is number of nodes in the cluster and $(k-1) \times J_i$ are the largest possible number of packets of file i in the cluster. To avoid the transmission of duplicate (useless) packets, a transmitting node compares its cached encoded packets with the packets received by the receiving node and only different packets are transmitted.

[1] For simplicity, we will write $j_i = j_i(t)$ and $L_R = L_R(t)$ in the remaining text.

III. FILE RECEPTION MODELING FRAMEWORK

As shown in Fig. 3, the packet reception process of a node's requested file may consist of multiple idle and packet receiving stages, both of which can span over multiple variable time slots. In Fig. 3, t_0 is the time when a requesting node initiates a request, t_e represents the time when the node finishes receiving all the packets of the requested file, and $t_0 + \Delta T$ is the deadline which represents the node in an outage if it has not received all the packets of the requested file until $t_0 + \Delta T$. To simplify analysis, we assume the same value of maximum waiting time ΔT for any file requested by any node. In this section, we first analyze the reception of packets with the help of a 2-D Markov chain and identify all transient and absorbing states. Then, we define transition probabilities among different states to derive the analytical expressions for the performance metrics of interest.

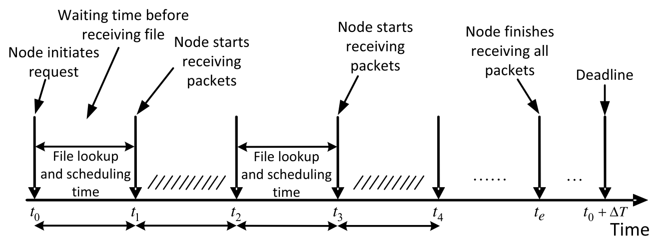

FIGURE 3. The node idle and packet receiving times.

A. TWO-DIMENSIONAL MARKOV MODEL

We construct a 2-D Markov chain to describe the reception of packets by the requesting node (say n) as shown in Fig. 4. In this Markov chain, a state is denoted by a pair of integers (l_i, τ), where l_i $(= J_i + j_i)$ represents the total number of packets of file i node n receives at the end of the τ-th time slot after the initiation of the request. For instance, state $(J_i, 0)$ represents the starting state at $\tau = 0$ when the node requests for the missing packets with J_i packet already received from BS and $j_i(t_0) = 0$ packet from other nodes. Here, to keep consistency with Fig. 3, we assume that the node initiates the request at t_0. The value of τ is incremented by 1, every time no matter whether the node receives packet(s) or not. It is assumed that all the transitions can utilize 1 time slot only.

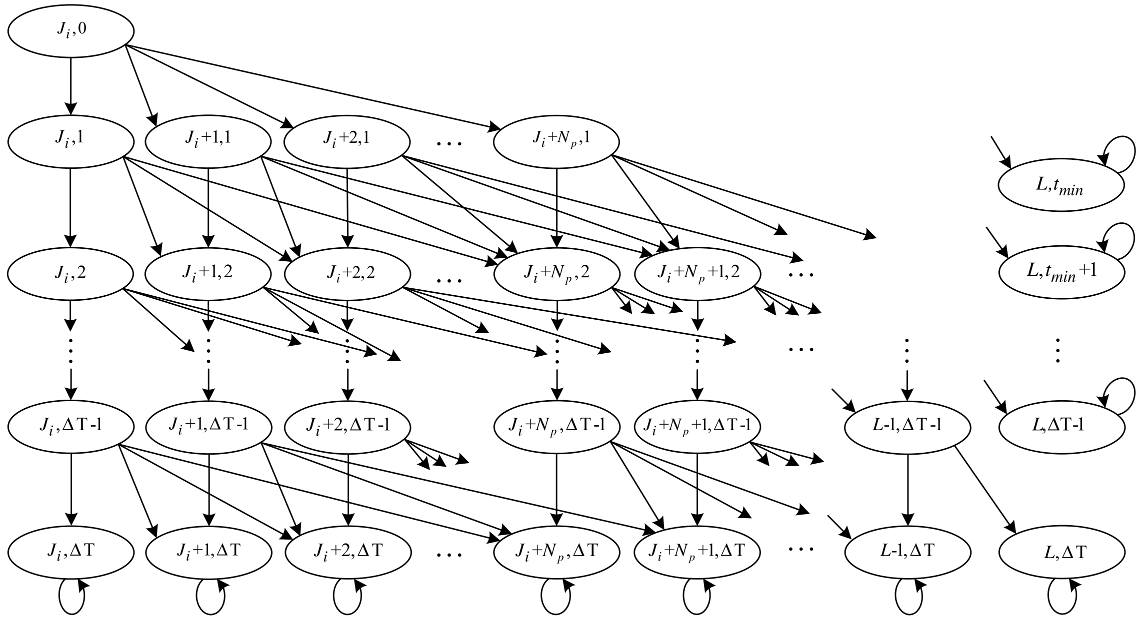

FIGURE 4. 2-D Markov chain for D2D transmissions of packets of a requested file from nodes in a mobile caching network.

The node initiates the request of file i at state $(J_i, 0)$ and from any state in the Markov chain receives $\alpha \in \{0, 1, \ldots, N_p\}$ number of packets from other nodes in the cluster and transits to a new state. Here, N_p represents the maximum number of packets received in one slot if there are no failed transmissions. The node may finish receiving packets at any of the states (L, t_{abs}), where $t_{abs} \in \{t_{min}, t_{min}+1, \ldots, \Delta T\}$ is the time required to receive all L packets. Here, $t_{min} = \left\lceil (L-J_i)/N_p \right\rceil$ is the minimum time which is required to receive all the packets. If we denote by s_{succ} the number of states with successful reception of L packets then we have $s_{succ} = \Delta T - t_{min} + 1$. If $L_R \leq N_p$, then from any transient state $(J_i+\beta, \tau_1)$, the node can transit to any of the s_{succ} states by receiving $\alpha = L_R$ packets where $L - J_i - N_p \leq \beta \leq L - J_i - 1$ and $\tau_1 \in \{t_{min}-1, t_{min}, \ldots, \Delta T - 1\}$. For simplicity, only one such transition between states $(L-1, \Delta T - 1)$ and $(L, \Delta T)$ is shown in Fig. 4.

The node transits to a deadlock state if after a time ΔT it does not receive the required number of packets. All the lower states in the chain as shown by $(J_i + q, \Delta T)$, where $0 \leq q \leq L - J_i - 1$, represent the deadlock states as the time for receiving the required packets expires. At ΔT, if the number of received packets is less than L then the node requesting the packets can be in any of the deadlock states. There are $s_d = L - J_i$ number of such deadlock states in the chain. Both s_{succ} and s_d are the absorbing states as after entering into any of these states process remains there and the state is treated as never changed. If we denote by s_{abs} the total number of absorbing states in the chain, then $s_{abs} = s_d + s_{succ}$.

From Fig. 4, we can observe that, until slot $\tau = t_{min} - 1$, the number of states in each slot is a summation of the number of states in the previous slot and N_p number of new states; and from slot $\tau = t_{min}$ to slot $\tau = \Delta T$ the number of states in each row remains the same, i.e., $(L - J_i + 1)$ states. Thus, the remaining number of states in the chain after slot $t_{min} - 1$ is $s_{rem} = (\Delta T - t_{min} + 1)(L - J_i + 1)$. If we denote by s_{tot} the total number of states in Fig. 4, then we have

$$s_{tot} = t_{min} + N_p \cdot \frac{t_{min}(t_{min}-1)}{2} + s_{rem}. \quad (3)$$

If we exclude the s_{abs} absorbing states from s_{tot} states in the Markov chain, then the remaining states are $s_{tr} = s_{tot} - s_{abs}$ transient states.

B. TRANSITION PROBABILITY

From Fig. 4, it can be observed that there are four types of transitions in the Markov chain. We now define these different types of state transitions and derive their transition probabilities.

Transition Type 1: In this transition type, node receives $\alpha = 0$ packet at the end of a time slot and transits to new state $(J_i + j_i, \tau + 1)$ from state $(J_i + j_i, \tau)$ with transition probability $p(J_i + j_i, \tau + 1 | J_i + j_i, \tau)$, where $j_i \in \{0, 1, 2, \ldots, L - J_i - 1\}$. There can be two possibilities if the node does not receive a packet after a time slot:

(1) The node is not in an active cluster;

(2) The node is in an active cluster but either it is alone in the cluster or it is not scheduled to receive packets or all the transmissions are failed or there are no nodes with new packets in the cluster.

Final expression for the transition probability $p(J_i + j_i, \tau+1 | J_i+j_i, \tau)$ is given in (4). The derivation of the probability is given in **Appendix A**.

$$\begin{aligned} &p(J_i+j_i, \tau+1|J_i+j_i, \tau) \\ &= \frac{K-1}{K} + \frac{1}{K} \\ &\times \Bigg\{ \left(1-\frac{1}{M^2}\right)^{N-1} + \sum_{k=2}^{N} C_{N-1}^{k-1} \left(\frac{1}{M^2}\right)^{k-1} \\ &\times \left(1-\frac{1}{M^2}\right)^{N-k} \times \Bigg[\frac{k-1}{k} + \frac{1}{k} \\ &\times \sum_{n_{new}=0}^{(k-1)J_i} p(n_r=0|N_{status}) p(n_{new}|j_i, k) \Bigg] \Bigg\} \end{aligned} \tag{4}$$

where N_{status} is the joint event representing the j_i packets received by the node until the previous slot, the status of the cluster as active, the number of nodes k in the cluster, the status of the node as scheduled, and the number of new packets n_{new} in the cluster, in the current slot; n_r denotes the number of received packets by the node in the current time slot; $p(\bar{n}_r = \alpha|N_{status})$ is the probability of the node receiving α packets in the current time slot and $p(n_{new}|j_i, k)$ is the probability of finding n_{new} new packets in the cluster of k nodes with j_i packets already received from the other nodes.

Transition Type 2: In this transition type, the node receives $\alpha > 0$ packet(s) and transits to a new state with transition probability $p(J_i+j_i+\alpha, \tau+1|J_i+j_i, \tau)$, where $j_i \in \{0, 1, 2, \ldots, L-J_i-1\}$.

Proof of $p(J_i+j_i+\alpha, \tau+1|J_i+j_i, \tau)$ for $\alpha > 0$ is given in **Appendix B**. Final expression for this transition probability is given by the expression as

$$\begin{aligned} &p(J_i+j_i+\alpha, \tau+1|J_i+j_i, \tau) \\ &= \frac{1}{K} \times \Bigg\{ \sum_{k=2}^{N} C_{N-1}^{k-1} \left(\frac{1}{M^2}\right)^{k-1} \left(1-\frac{1}{M^2}\right)^{N-k} \\ &\times \Bigg[\frac{1}{k} \times \sum_{n_{new}=0}^{(k-1)J_i} p(n_r=\alpha|N_{status}) p(n_{new}|j_i, k) \Bigg] \Bigg\} \end{aligned} \tag{5}$$

where $p(n_r = \alpha|N_{status})$ is the probability of the node receiving $\alpha > 0$ packets in the current time slot.

Transition Type 3: In this transition type, node once entered into a particular state remains there forever with transition probability $p(J_i+j_i, \Delta T|J_i+j_i, \Delta T) = 1$, where $j_i \in \{0, 1, 2, \ldots, L-J_i-1\}$. States in this transition type represent the deadlock states.

Transition Type 4: This transition type represents the reception of the required L packets. The node transits from state (L, t_{abs}) to state (L, t_{abs}) in each time slot with transition probability $p(L, t_{abs}|L, t_{abs}) = 1$.

To find the probabilities of transition types 1 and 2, we still need to determine the probability of receiving new packets $p(n_r = \alpha|N_{status})$ and the probability of number of new packets in the cluster $p(n_{new}|j_i, k)$ in the current time slot. Receiving $\alpha(\in \{0, 1, \ldots, N_p\})$ packets depends on the probability of successful transmissions (p_s). If there are all failed transmissions or there are no new packets in the cluster then we receive $n_r = 0$ packets; otherwise, we might receive $n_r > 0$ packets. We find $p(n_r = \alpha|N_{status})$ with respect to p_s as follows:

(1) If $p_s = 1$ then

$$p(n_r = \alpha|N_{status}) = \begin{cases} 1, & \text{if } L_R \geq N_p, \quad \alpha = \min\{n_{new}, N_p\} \\ 0, & \text{if } L_R \geq N_p, \quad \alpha \neq \min\{n_{new}, N_p\} \\ 1, & \text{if } L_R < N_p, \quad \alpha = \min\{n_{new}, L_R\} \\ 0, & \text{if } L_R < N_p, \quad \alpha \neq \min\{n_{new}, L_R\} \\ 1, & \text{if } n_{new} = 0, \quad \alpha = 0 \\ 0, & \text{if } n_{new} = 0, \quad \alpha > 0. \end{cases} \tag{6}$$

We derive (6) as follows:

- When $L_R \geq N_p$ then we can receive minimum of n_{new} or N_p packets, i.e., $p(n_r = \alpha|N_{status}) = 1$ for $\alpha = \min\{n_{new}, N_p\}$, otherwise, $p(n_r = \alpha|N_{status}) = 0$.
- When $L_R < N_p$ then we can receive minimum of n_{new} or L_R packets, i.e., $p(n_r = \alpha|N_{status}) = 1$ for $\alpha = \min\{n_{new}, L_R\}$, otherwise, $p(n_r = \alpha|N_{status}) = 0$.
- When $n_{new} = 0$, $p(n_r = \alpha|N_{status}) = 1$ for $\alpha = 0$ and $p(n_r = \alpha|N_{status}) = 0$ for $\alpha > 0$.

(2) When $p_s < 1$, we consider the following four cases:

Case 1: When $n_{new} \geq N_p$ and $L_R \geq N_p$, we receive $0 \leq \alpha \leq N_p$. Since α represents the number of received packets successfully in N_p transmissions, then α is a binomial random variable with parameters N_p and p_s. We can write the probability mass function of a binomial random variable as

$$p(n_r = \alpha|N_{status}) = C_{N_p}^{\alpha} (p_s)^{\alpha} (1-p_s)^{N_p-\alpha}. \tag{7}$$

Case 2: When $n_{new} \geq N_p$ and $L_R < N_p$, then, either we receive $0 \leq \alpha < L_R$ packets in N_p mini-slots or we need either L_R number of minimum mini-slots or we need N_p number of maximum mini-slots to receive $\alpha = L_R$ packets. We can find $p(n_r = \alpha|N_{status})$ as

$$p(n_r = \alpha|N_{status}) = \begin{cases} C_{N_p}^{\alpha} (p_s)^{\alpha} (1-p_s)^{N_p-\alpha}, & \text{if } 0 \leq \alpha < L_R \\ \displaystyle\sum_{i=L_R}^{N_p} C_{N_p}^{i} (p_s)^{i} (1-p_s)^{N_p-i}, & \text{if } \alpha = L_R \end{cases} \tag{8}$$

where $C_{N_p}^{i}$ represents the total number of combinations of successful and unsuccessful transmissions given a successful transmission of the L_R-th packet in the i-th mini-slot.

Case 3: When $n_{new} < N_p$ and $L_R \geq N_p$, then we can receive maximum of n_{new} packets. We have the following three possibilities.

- We can receive $\alpha = 0$ packets if all the N_p transmissions are unsuccessful.
- There is a possibility that we receive $0 < \alpha < n_{new}$ packets in N_p mini-slots as n_{new} is less than L_R and N_p.
- We can also receive $\alpha = n_{new}$ packets in either n_{new} number of minimum mini-slots or N_p number of maximum mini-slots.

So, $p(n_r = \alpha|N_{status})$ in this case can be found as

$$p(n_r = \alpha|N_{status}) = \begin{cases} (1-p_s)^{N_p}, & \text{if } \alpha = 0 \\ C_{N_p}^{\alpha}(p_s)^{\alpha}(1-p_s)^{N_p-\alpha}, & \text{if } 0 < \alpha < n_{new} \\ \sum_{i=n_{new}}^{N_p} C_{i-1}^{i-n_{new}}(p_s)^{n_{new}} \cdot (1-p_s)^{i-n_{new}}, & \text{if } \alpha = n_{new} \end{cases} \quad (9)$$

where $C_{i-1}^{i-n_{new}}$ represents the total number of combinations of successful and unsuccessful transmissions given a successful transmission of the n_{new}-th packet in the i-th mini-slot.

Case 4: When $n_{new} < N_p$ and $L_R < N_p$, we have two possibilities: if $n_{new} < L_R$, then in a slot we receive maximum of n_{new} packets, otherwise, we receive maximum of L_R packets in a slot. Thus, we define $g = \min\{n_{new}, L_R\}$ as the maximum number of packets the node can receive in a slot. We can proceed similar to case 3 and find $p(n_r = \alpha|N_{status})$ in this case as

$$p(n_r = \alpha|N_{status}) = \begin{cases} (1-p_s)^{N_p}, & \text{if } \alpha = 0 \\ C_{N_p}^{\alpha}(p_s)^{\alpha}(1-p_s)^{N_p-\alpha}, & \text{if } 0 < \alpha < g \\ \sum_{i=g}^{N_p} C_{i-1}^{i-g}(p_s)^{g}(1-p_s)^{i-g}, & \text{if } \alpha = g \end{cases} \quad (10)$$

where C_{i-1}^{i-g} represents the total number of combinations of successful and unsuccessful transmissions given a successful transmission of the g-th packet in the i-th mini-slot.

For $p(n_{new}|j_i, k)$, we can find it as follows:

(1) If the node has not received any packet from other nodes, i.e., $j_i = 0$, $p(n_{new}|j_i, k)$ is computed as

$$p(n_{new}|j_i, k) = \begin{cases} 0, & \text{if } n_{new} \neq (k-1)J_i \\ 1, & \text{if } n_{new} = (k-1)J_i. \end{cases} \quad (11)$$

(2) If the node has already received some packets from other nodes, i.e., $j_i > 0$, $p(n_{new}|j_i, k)$ is computed using (12) as

$$p(n_{new}|j_i, k) = \begin{cases} 0, & \text{if } k = 1 \text{ and } n_{new} \neq 0 \\ 1, & \text{if } k = 1 \text{ and } n_{new} = 0 \\ 0, & \text{if } e > h \text{ and } e-h < j_i - d \\ \frac{C_{N-1}^{k-1} \times C_h^d \times C_{e-h}^{j_i-d}}{C_{N-1}^{k-1} \times C_e^{j_i}}, & \text{if } e > h \text{ and } e-h \geq j_i - d \\ 1, & \text{if } e = h, n_{new} = e - j_i, d = j_i \\ 0, & \text{if } e = h, n_{new} \neq e - j_i \end{cases} \quad (12)$$

where $e = (N-1)J_i$ denotes the total number of cached packets of file i in the network, $h = (k-1)J_i$ denotes the total number of cached packets of file i in the cluster, and d represents the number of packets already received from the nodes in the cluster. $p(n_{new}|j_i, k)$ for $j_i > 0$ is derived in **Appendix C**.

Using the above defined transition probabilities, we now proceed to find the closed form expressions for the performance metrics in the following section.

IV. DERIVATION OF PERFORMANCE METRICS

In this section, we first define performance metrics and then derive their analytical expressions using different transition probabilities found in Section III-B. When the file is not received completely until deadline then file outage occurs. Therefore, the first performance metric studied here is the average outage probability, which is defined as the probability of files not received completely in time ΔT. We also find the average delay measured by the average number of slots required from request initiation to reception of all the packets of the file. Another performance metric is to explore the joint impact of average delay and outage probability on the number of packets received per slot by the nodes in the network, called the average throughput.

After defining performance metrics, we now derive their analytical expressions. Using the above mentioned four transition types, we first find the one-step transition probabilities in the Markov chain to construct a transition matrix [52], where each transition represents the reception of $\alpha \in \{0, 1, \ldots, N_p\}$ packets of file i. Let $\mathbf{T}_{\mathbf{M},i} = [p(J_i + j_i + \alpha, \tau+1|J_i + j_i, \tau)]_{s_{tot} \times s_{tot}}$ be the probability transition matrix of discrete-time Markov chain which contains all the one-step transition probabilities $p(J_i + j_i + \alpha, \tau + 1|J_i + j_i, \tau) \geq 0$ of transition from state $(J_i + j_i, \tau)$ to state $(J_i + j_i + \alpha, \tau + 1)$ in Fig. 4. The transition matrix $\mathbf{T}_{\mathbf{M},i}$ can be represented in canonical form [53], [54] as

$$\mathbf{T}_{\mathbf{M},i} = \begin{bmatrix} \mathbf{Q}_i & \mathbf{R}_i \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \quad (13)$$

where $\mathbf{Q}_i$ is a $s_{tr} \times s_{tr}$ matrix which defines the one-step transition probabilities among all s_{tr} transient states in the chain, $\mathbf{R}_i$ is a $s_{tr} \times s_{abs}$ matrix defining the one-step transition probabilities from s_{tr} transient states to s_{abs} absorbing states, $\mathbf{0}$ is zero matrix of size $s_{tr} \times s_{tr}$, and $\mathbf{I}$ is an identity matrix of size $s_{abs} \times s_{abs}$. To find the closed form expressions for the performance metrics, we first find the τ-step transition probability matrix via multiplying matrix $\mathbf{T}_{\mathbf{M},i}$ by itself ΔT times as

$$\mathbf{T}_{\mathbf{M},i}{}^{(\Delta T)} = \mathbf{T}_{\mathbf{M},i}{}^{(\Delta T-1)} \cdot \mathbf{T}_{\mathbf{M},i} \quad (14)$$

where the dot represents the matrix multiplication. After ΔT transitions of $\mathbf{T}_{\mathbf{M},i}$, we obtain new matrices $\mathbf{Q}_i^{(\Delta T)}$ and $\mathbf{R}_i^{(\Delta T)}$. Referring to Fig. 4, it is noted that performance metrics are determined by using the s_{abs} absorbing states. To find the performance metrics from the initial state $(J_i, 0)$, we use the probabilities of the s_{abs} absorbing states given in the first row of $\mathbf{R}_i{}^{(\Delta T)}$, denoted by $[p_i(L, t_{min}), p_i(L, t_{min}+1), \ldots, p_i(L, \Delta T - 1), p_i(J_i{+}0, \Delta T), p_i(J_i{+}1, \Delta T), \ldots, p_i(L{-}1, \Delta T), p_i(L, \Delta T)]$. Here, $p_i(L, t_{abs})$ is the probability of successfully receiving L

packets of file i at slot $t_{abs} \in \{t_{min}, t_{min}+1, \ldots, \Delta T\}$ and $p_i(J_i+q, \Delta T)$ is the probability of receiving $q(\in\{0,1,2,\ldots,L-J_i-1\})$ packets until time ΔT starting from the initial state $(J_i, 0)$.

A. CONTENT OUTAGE PROBABILITY

Time to wait for the request response of each file has a maximum time bound ΔT. Node is considered in an outage if it does not get sufficient coded packets of the requested file in ΔT time and goes to a deadlock state. Average outage probability $\bar{P}_{out}$ of a node n is given by

$$\bar{P}_{out} = \sum_i p_{out}(i) p_r(i) \tag{15}$$

where $p_{out}(i)$ is the outage probability of node n requesting file i and is a sum of probabilities of all the s_d deadlock states in the Markov chain. Expression of $p_r(i)$ is given in (2). $p_{out}(i)$ can be found from the matrix $\mathbf{R}_i^{(\Delta T)}$ using the expression as

$$p_{out}(i) = \sum_{q=0}^{L-J_i-1} p_i(J_i+q, \Delta T). \tag{16}$$

The average number of nodes in outage $\mathrm{u}_o(\Delta T)$ is given by

$$\mathrm{u}_o(\Delta T) = N \cdot \bar{P}_{out}. \tag{17}$$

Since the node can receive all the packets of a file at any of the s_{succ} states, the sum of the probabilities $p_i(L, t_{min}), \ldots, p_i(L, \Delta T - 1)$, and $p_i(L, \Delta T)$ in $\mathbf{R}_i^{(\Delta T)}$ gives the probability of successfully receiving L packets of file i or the absorbing probability $p_{abs}(i)$ of successfully receiving file i. The expression for P_{abs} of successfully receiving files is given by

$$P_{abs} = \sum_i p_{abs}(i) p_r(i) \tag{18}$$

where $p_{abs}(i)$ is the absorbing probability of particular file i and is a sum of the probabilities of all the s_{succ} number of successful states in the Markov chain. $p_{abs}(i)$ can be found from $\mathbf{R}_i^{(\Delta T)}$ using the expression as

$$p_{abs}(i) = \sum_{t_{abs}=t_{min}}^{\Delta T} p_i(L, t_{abs}) = 1 - p_{out}(i). \tag{19}$$

B. DELAY

Expression for the average delay $\bar{D}$ of node n is given by

$$\bar{D} = \sum_i \bar{D}(i) p_r(i) \tag{20}$$

where $\bar{D}(i)$ is the average delay in receiving packets of file i.

It can be observed from the Markov chain that the sojourn time of each state is 1 time slot and after a slot, state transition happens with the reception of $\alpha \geq 0$ packets. To find $\bar{D}(i)$, we find the total number of transitions needed to reach any of the absorbing states (L, t_{abs}), which represents the successful reception of all the L packets at time t_{abs}. Moreover, file outage can also occur due to incomplete reception of file i in ΔT time as shown in Fig. 4. Therefore, we also consider the delay of partially received file i in ΔT, i.e. $p_{out}(i) \cdot \Delta T$. If we denote by τ the total number of slots required to reach any of the (L, t_{abs}) states and, from $\mathbf{R}_i^{(\Delta T)}$, $p_i(L, \tau)$ is the probability of receiving L packets after τ slots. Then the final expression of average delay $\bar{D}(i)$, considering the impact of file outage as well, is represented as

$$\bar{D}(i) = \sum_{\tau=t_{min}}^{\Delta T} \tau \cdot p_i(L, \tau) + p_{out}(i) \cdot \Delta T \tag{21}$$

where $p_{out}(i)$ is defined in (16).

C. THROUGHPUT

Throughput ($\bar{T}_n$) of node n depends on content transfer delay and file outage probability and can be computed by the following expression as

$$\bar{T}_n = \sum_i T_n(i) p_r(i) \tag{22}$$

where $T_n(i)$ is the throughput of node n for file i. If node n needs $L - J_i$ packets of file i then its throughput (in packets per slot) is computed by the following expression as

$$T_n(i) = \frac{L - J_i}{\bar{D}(i)}(1 - p_{out}(i)). \tag{23}$$

Using (2) and (23), we find $\bar{T}_n$ from (22). Average network throughput $\bar{T}$ can be found by the summation of throughput of all the nodes as $\bar{T} = \sum_n \bar{T}_n = N\bar{T}_n$.

V. NUMERICAL RESULTS

A. RESULTS VERIFICATION

We do extensive simulations to verify our analysis. We simulate a network of square grid of 16 clusters, i.e., $M = 4$ with TDMA spatial reuse factor $K = 4$, $\gamma_r = 0.1$, $\gamma_c = 0.4$, $L = 30$, $\Delta T = 80$, $H \in \{120, 180, 240, 300, 360, 420\}$ packets under three parameter settings, i.e., $N = 200$, $F = 100$, $N_p = 7$, $p_s \in \{1, 0.9\}$, and $N = 200$, $F = 150$, $N_p = 7$, $p_s = 1$ with each simulation run of 20000 slots.

In Fig. 5, we summarize the results of performance metrics obtained from both analysis and simulations. It can be observed clearly that for the tested network parameter settings theoretical results match well with the simulation results. By increasing H, each node caches more packets of files and every node can find its requested file from other nodes in comparatively less time while visiting different clusters. Due to this reason, in Fig. 5(a), we see a decreasing trend in outage probability with an increase of cache size H at each node. Consequently, we see a decrease of delay and an increase of throughput, respectively, as shown in Figs. 5(b) and 5(c). We also see an increase of outage probability and delay, and a decrease of throughput when we reduce p_s. Moreover, by increasing F, performance of network also decreases due

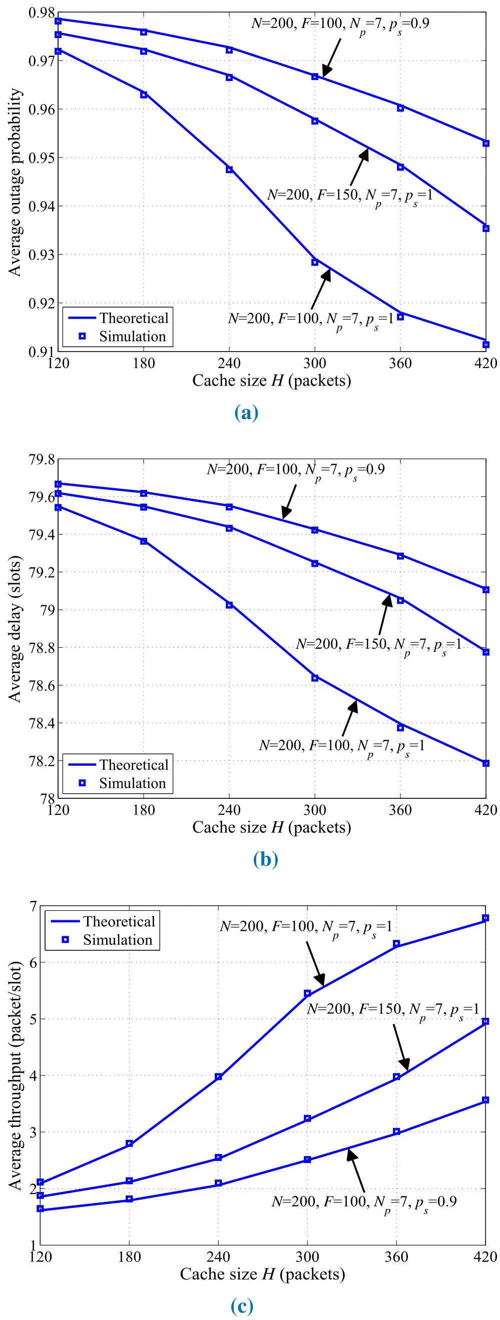

FIGURE 5. **Comparison of performance metrics obtained from analysis and simulations. (a) Outage probability. (b) Average delay. (c) Average throughput.**

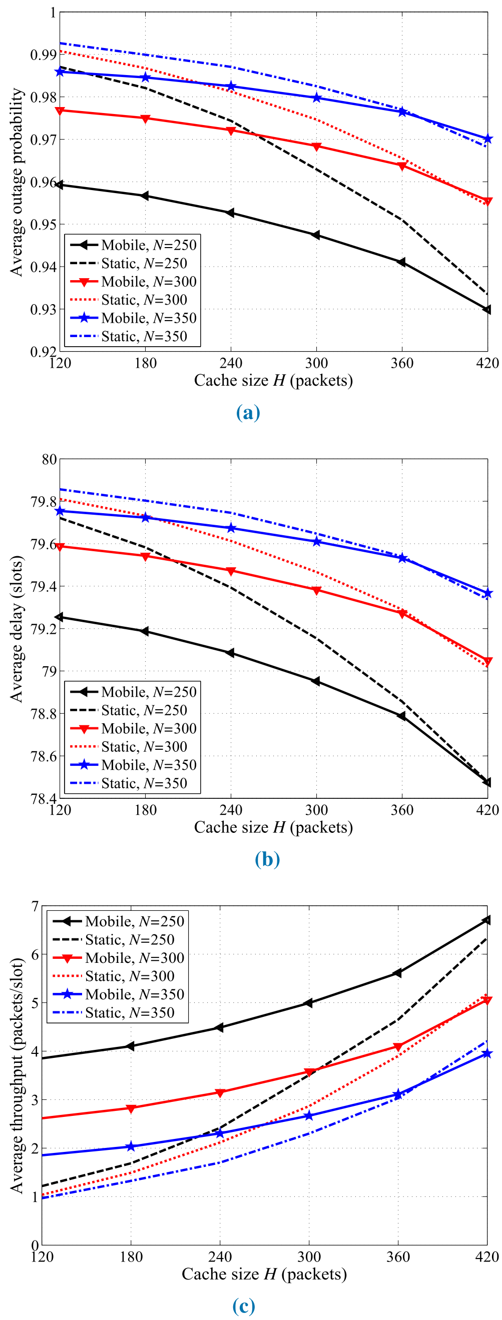

FIGURE 6. **Impact of the number of nodes and cache size on network performance. (a) Outage probability. (b) Average delay. (c) Average throughput.**

to reduced number of cached packets of each file at each node. With less number of cached packets of each file, the node either needs more scheduling opportunities, large N_p or more time to recover the complete file. Consequently, outage probability and delay increase, and throughput decreases.

B. PERFORMANCE ANALYSIS

In this subsection, we apply our analysis to five different network scenarios, evaluate the impact of different parameter settings on performance metrics and compare the results with the baseline, i.e., the static network scenario (where the nodes are randomly deployed and static in their clusters). We simulate a network with default parameters $M = 4$, $K = 4$, $F = 150$, and $H \in \{120, 180, 240, 300, 360, 420\}$ packets (except the figures where we consider to change specific parameters).

In the first scenario, as shown in Fig. 6, we study the impact of varying cache size per node and node number $N \in \{250, 300, 350\}$ on the network performance under $\gamma_r = 0.1$, $\gamma_c = 0.4, L = 30, \Delta T = 80, N_p = 9$, and $p_s = 1$. It is evident from Fig. 6 that we achieve a decrease of outage probability and delay, and an increase of throughput with an increase of H. The performance gap between the mobile and the static cases is large at small H and this gap reduces gradually as H and N increase. This is due to the reason that at small H

node mobility increases the probability to find the requested file even it is less popular. But when H increases, the number of cached packets of different files also increase, which helps to receive more packets of the requested files from inside of the cluster. For this reason, the network performance in the static case at $H = 420$ and $N = 300$ or $N = 350$ is better than that in the mobile case. Different from [34] (where the nodes cache complete files), we can observe from Fig. 6 that by increasing N in the network we see an increase of outage probability and average delay, and a decrease of average throughput. This is due to the reason that by increasing the number of nodes in the cluster, the number of requests in the cluster also increases, which reduces the per node scheduling probability. Further, when nodes cache packets based on popularity (i.e., few packets of each file), each node needs to find more useful transmitters compared with caching complete file and thus needs more scheduling opportunities. Accordingly, nodes getting complete files in ΔT reduce, which limits the network throughput.

In the second scenario, as shown in Fig. 7, we analyze the impact of varying file caching parameter $\gamma_c \in \{0.4, 0.5, 0.6\}$ on performance metrics with respect to $N \in \{200, 240, 280, 320, 360, 400\}$ by fixing $H = 240$, $\gamma_r = 0.3$, $L = 30$, $\Delta T = 80$, $N_p = 9$, and $p_s = 1$. We can observe from Fig. 7 that both outage probability and delay increase, and throughput decreases with an increase of N. Moreover, we get higher throughput, and lower outage and delay with an increase of γ_c, e.g., throughput at $\gamma_c = 0.5$ is higher than that at $\gamma_c = 0.4$. This is due to the fact that when γ_c is large we cache more packets of files based on popularity. Therefore, the nodes take less time to receive all packets of requested files at large γ_c and more files are received in ΔT time. For this reason, outage probability and file transfer delay decrease, and we see an increase of throughput at large γ_c. In addition, compared with the mobile case, there is a slight improvement in delay in Fig. 7(b) at $\gamma_c = 0.6$ for $N < 260$ in the static case, as caching more packets of files helps to receive them from own cluster in less time at the small number of nodes. When the number of nodes increases, scheduling probability of each node reduces, which increases the delay in the static case compared with the mobile case.

In the third scenario, we analyze the impact of selection of $\gamma_c \in \{0.3, 0.4, 0.5\}$ with respect to $\gamma_r \in \{0.15, 0.2, 0.25, 0.3, 0.35, 0.4\}$ under $N = 250$, $\Delta T = 80$, $p_s = 1$, $L = 30$, $H = 300$, and $N_p = 9$. From Fig. 8, we can observe that network performance improves with the increase of γ_r and γ_c. When γ_r is small, node mobility helps to receive the diverse files. Therefore, performance gap between the static and the mobile cases is prominent at small γ_r. Moreover, when each node caches more packets of popularity based files and requests files with large γ_r, we see a small improvement in delay in the static case and delay further reduces when we increase file requesting parameter γ_r. Thus, caching files at large γ_c and requesting popular files helps to find useful transmitters from inside the cluster and reduces delay in the static case, specifically. In addition, due to limited number

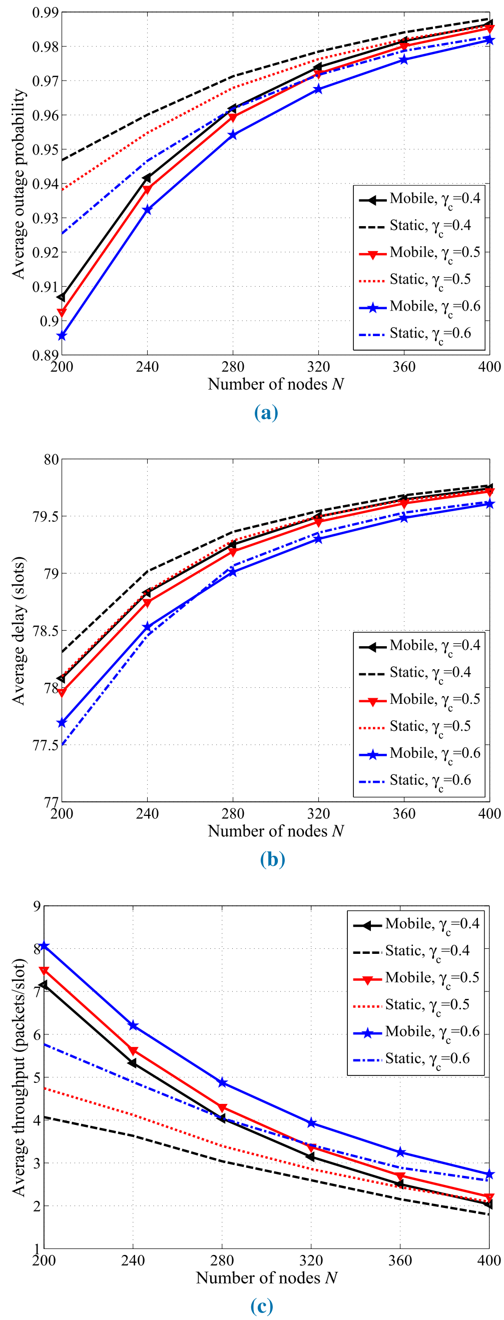

FIGURE 7. **Impact of number of nodes and γ_c on network performance. (a) Outage probability. (b) Average delay. (c) Average throughput.**

of nodes and small cache size of each node in the cluster, all the files cannot be received from the nearby nodes in the cluster; therefore, despite lower delay, outage probability is higher and throughput is lower in the static case at large γ_r compared with the mobile case.

In the fourth scenario, we analyze the impact of varying the number of mini-slots $N_p \in \{7, 8, 9\}$ with respect to $N \in \{200, 240, 280, 320, 360, 400\}$ under $H = 240$, $N = 250$, $\Delta T = 80$, $p_s = 1$, $L = 30$, $\gamma_r = 0.3$, and $\gamma_c = 0.4$. It can be noticed from Fig. 9 that when N_p increases, more packets can be transmitted in a time slot. Consequently, outage probability and delay decrease, and throughput increases. Moreover, there is a small improvement in network performance in

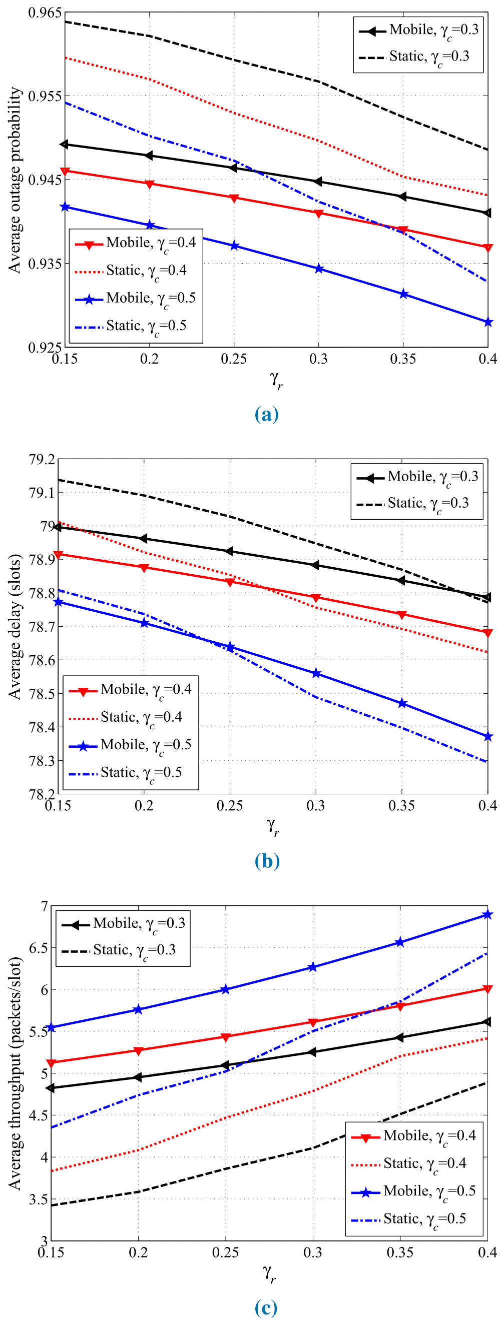

FIGURE 8. Impact of file request and caching parameters on network performance. (a) Outage probability. (b) Average delay. (c) Average throughput.

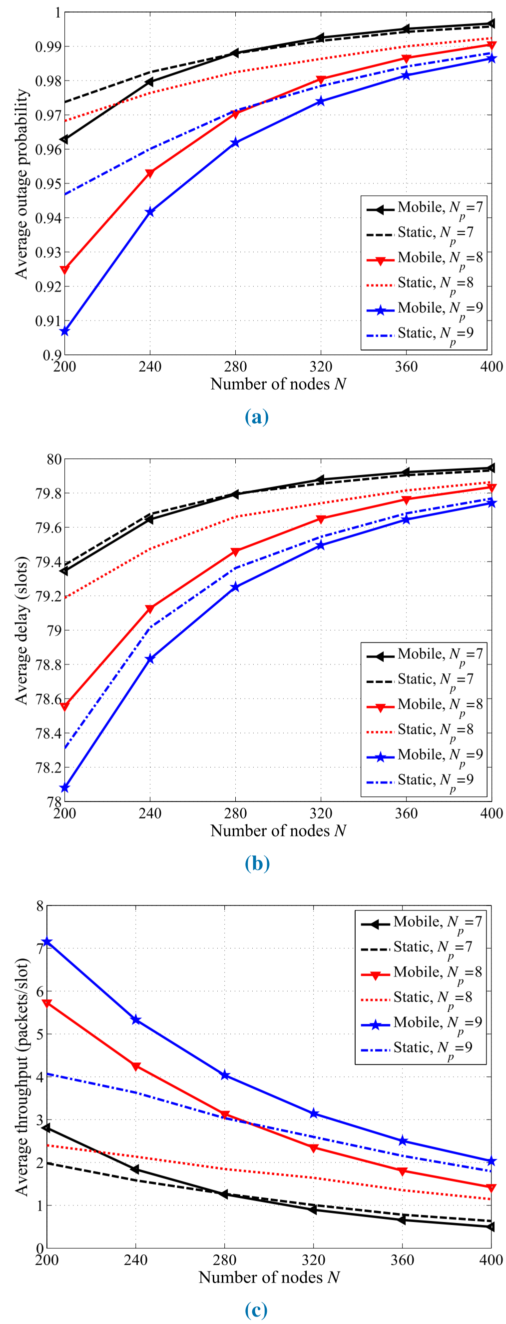

FIGURE 9. Impact of the number of nodes and mini-slots on network performance. (a) Outage probability. (b) Average delay. (c) Average throughput.

the static case when $N \geq 280$ and $N_p = 7$ as compared with the mobile case. The reason is that when N_p is small, each receiver needs to be scheduled more times to receive data and thus needs more useful transmitters. As the useful transmitters do not move to other clusters in the static case, more packets can be received from these useful transmitters in the future scheduling opportunities. But in the mobile case, finding the useful transmitter and obtaining scheduling opportunity may take time. As a result, we see little increase of outage probability and delay, and a decrease of throughput in the mobile case compared with the static case.

In the fifth scenario, we fix $\gamma_c = 0.4$ with $\gamma_r = 0.1$, $N = 200$, $F = 100$, $L = 30$, $\Delta T = 80$, $N_p = 7$, $H \in \{120, 180, 240, 300, 360\}$, and analyze the network performance by varying packet's successful transmission probability, i.e., $p_s \in \{1.0, 0.8, 0.6\}$. It can be observed from Fig. 10 that when we reduce success probability p_s of packet transmission we see an increase of outage probability and delay and consequently, a decrease of throughput. Besides, the performance gap between the mobile and the static cases also reduces with the decrease of p_s and gradually the performance of static case dominates the mobile case.

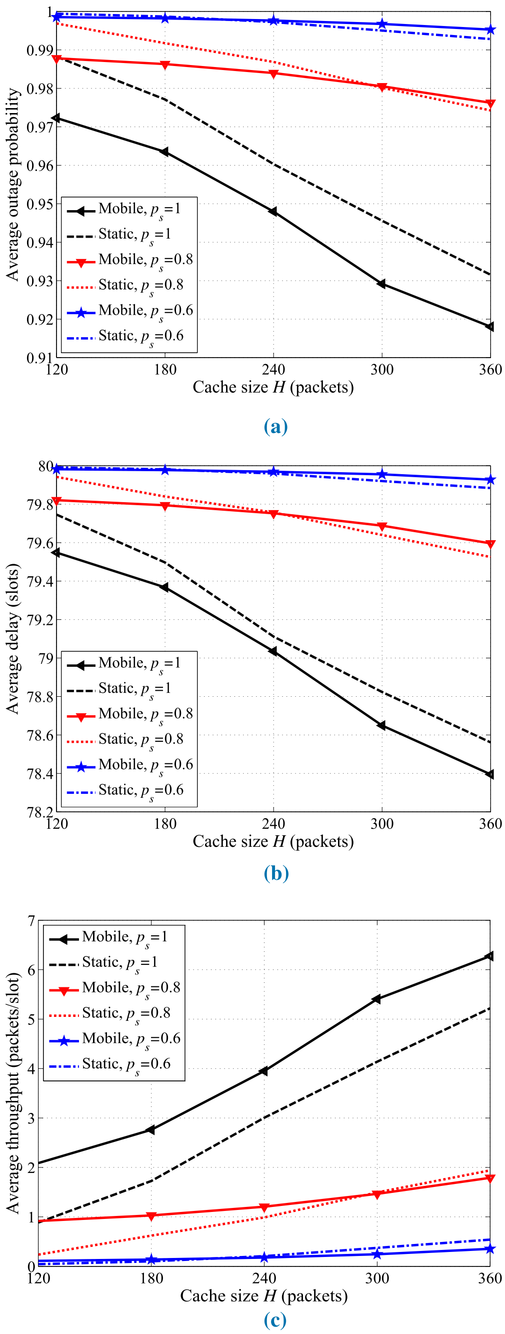

FIGURE 10. Impact of transmission error on network performance. (a) Outage probability. (b) Average delay. (c) Average throughput.

For instance, in Fig. 10, at $p_s = 0.8$ and at $H \geq 300$ there is a small improvement in network performance in the static case as compared to the mobile case. The reason behind the improvement in the static case is that request life time ΔT is not sufficient to receive sufficient packets of files as the number of mini-slots wasted in each slot increases at high transmission error. Therefore, it becomes difficult to find sufficient useful transmitters in ΔT in the mobile case while in the static case useful transmitters do not move to other clusters and can retransmit unsuccessful packets during future scheduling opportunities.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have studied the impact of node mobility in a cluster based wireless caching network. The nodes generate requests for the files and receive them from the nearby nodes on a D2D link under fast mobility scenario. We have investigated the effect of node mobility on network performance using a 2-D Markov chain framework and derived closed-form expressions for the performance metrics, namely outage probability, delay, and throughput. In addition, we have validated our analysis comprehensively using simulation results. From the analysis it is found that network performance improves in the mobile scenario with respect to the static scenario when the nodes have limited cache size. The performance gap between the static and the mobile network cases reduces if node number or cache size increases but per-node scheduling probability is reduced or low, which means enabling simultaneous transmissions and/or receptions in each cell will help further exploit the benefit of node mobility and caching capability. Furthermore, it is also found that the performance of the static case can dominate the mobile case when successful transmission probability is small or the duration of each contact time is short. For the future work, we will extend our analysis to include multiple transmissions in a cluster with rate adaptation and study other mobility models as well.

APPENDIX A
DERIVATION OF (4)

To derive the expression for probability $p(J_i + j_i, \tau + 1|J_i + j_i, \tau)$, where $j_i \in \{0, 1, 2, \ldots, L - J_i - 1\}$, let A_0 be an event that node n has $J_i + j_i$ packets of file i at the end of current time slot $\tau + 1$, B be an event that node n received $J_i + j_i$ packets of file i at the end of the previous time slot τ, $C = 0$ be an event which defines the cluster of node n as inactive, and $C = 1$ be an event which denotes the cluster of node n as active. From these events, we find the probability of receiving $\alpha = 0$ packets as

$$\begin{aligned} p(J_i + j_i, \tau + 1|J_i + j_i, \tau) &= p(A_0|B) \\ &= \sum_C p(A_0|B, C)p(C|B) \\ &= \sum_C p(A_0|B, C)p(C) \end{aligned} \quad (24)$$

where the final expression holds due to independent relationship between B and C. According to the TDMA spatial reuse, we have

$$p(C) = \begin{cases} 1 - p^{act}_{cluster}, & \text{if } C = 0 \\ p^{act}_{cluster}, & \text{if } C = 1. \end{cases} \quad (25)$$

If $C = 0$ then $p(A_0|B, C = 0) = 1$; otherwise, using the law of total probability, $p(A_0|B, C = 1)$ can be found by considering all the possible number of nodes in the cluster as

$$p(A_0|B, C = 1) = \sum_{k=1}^{N} p(A_0|B, C = 1, k)p(k|B, C = 1). \quad (26)$$

Since k has no dependency on B and C, we can write $p(k|B, C = 1) = p(k)$, which is equivalent to the probability of finding $k - 1$ number of other nodes in the cluster as

$$p(k) = \begin{cases} (1 - \frac{1}{M^2})^{N-k}, & \text{if } k = 1 \\ C_{N-1}^{k-1}\left(\frac{1}{M^2}\right)^{k-1}\left(1 - \frac{1}{M^2}\right)^{N-k}, & \text{if } k > 1. \end{cases} \quad (27)$$

It is obvious that $p(A_0|B, C = 1, k) = 1$ for $k = 1$, then (26) can be simplified as

$$p(A_0|B, C = 1) = p(k = 1) + \sum_{k=2}^{N} p(A_0|B, C = 1, k)p(k). \quad (28)$$

For a cluster with $k > 1$ nodes, we define events $S = 1$ and $S = 0$ to represent whether or not node n is scheduled as the receiver in the cluster, respectively. Then, we have

$$p(A_0|B, C = 1, k) = \sum_{S} p(A_0|B, C = 1, k, S) \cdot p(S|B, C = 1, k). \quad (29)$$

Given $j_i < L - J_i$, because S is independent of B, we can write $p(S|B, C = 1, k) = p(S|C = 1, k)$ which is found as

$$p(S|C = 1, k) = \begin{cases} 1 - \frac{1}{k}, & \text{if } S = 0 \\ \frac{1}{k}, & \text{if } S = 1. \end{cases} \quad (30)$$

Obviously, $p(A_0|B, C = 1, k, S = 0) = 1$. For $S = 1$, we check the number of new packets $n_{new} \in \{0, 1, 2, \ldots, (k-1) \times J_i\}$ in the cluster as

$$\begin{aligned} p(A_0|B, C = 1, k, S = 1) &= \sum_{n_{new}} p(A_0|B, C = 1, k, S = 1, n_{new}) \\ &\quad \cdot p(n_{new}|B, C = 1, k, S = 1). \end{aligned} \quad (31)$$

Since the number of new packets n_{new} in the cluster depends on the number of already received packets j_i and number of nodes k in the cluster, $p(n_{new}|B, C = 1, k, S = 1)$ can be rewritten as

$$p(n_{new}|B, C = 1, k, S = 1, j_i) = p(n_{new}|j_i, k) \quad (32)$$

where the final expression holds due to independent relationship between n_{new} and $C = 1$ and $S = 1$. Since information of number of received packets can be found from j_i, to remove redundancy in events, we omit B and write j_i only in (32). For notation simplicity, let $N_{status} = (B, C = 1, k, S = 1, n_{new})$ and n_r denote the number of received packets, then

$$p(A_0|B, C = 1, k, S = 1, n_{new}) = p(n_r = 0|N_{status}). \quad (33)$$

From the above derivations (24)−(33), we derive the final expression for finding the probability $p(J_i + j_i, \tau + 1|J_i + j_i, \tau)$, as shown in (4).

APPENDIX B
DERIVATION OF (5)

To find $p(J_i + j_i + \alpha, \tau + 1|J_i + j_i, \tau)$ for $j_i \in \{0, 1, 2, \ldots, L - J_i - 1\}$ and $\alpha > 0$, we proceed similar to the proof of $p(J_i + j_i, \tau + 1|J_i + j_i, \tau)$ in **Appendix A**. By reusing those events defined in **Appendix A** and a new event A_α to denote node n has $J_i + j_i + \alpha$ packets of file i at the end of current time slot $\tau + 1$, we can write

$$\begin{aligned} p(J_i + j_i + \alpha, \tau + 1|J_i + j_i, \tau) &= p(A_\alpha|B) \\ &= p(A_\alpha|B, C = 1)p(C = 1|B) \\ &= p(A_\alpha|B, C = 1)p_{cluster}^{act}. \end{aligned} \quad (34)$$

Using the law of total probability, $p(A_\alpha|B, C = 1)$ can be found by considering all possible number of nodes in the cluster as

$$p(A_\alpha|B, C = 1) = \sum_{k=2}^{N} p(A_\alpha|B, C = 1, k) \cdot p(k|B, C = 1). \quad (35)$$

As k has no dependency on B and C, we can write $p(k|B, C = 1) = p(k)$ which can be found from (27).

We find the node scheduling probability as

$$p(A_\alpha|B, C = 1, k) = p(A_\alpha|B, C = 1, k, S = 1) \cdot p(S = 1|B, C = 1, k). \quad (36)$$

Given $j_i < L - J_i$, because S is independent of B, we can write $p(S = 1|B, C = 1, k) = p(S = 1|C = 1, k)$, which is defined in (30).

We now check the number of new packets $n_{new} \in \{0, 1, 2, \ldots, (k-1) \times J_i\}$ in the cluster as

$$\begin{aligned} p(A_\alpha|B, C = 1, k, S = 1) &= \sum_{n_{new}} p(A_\alpha|B, C = 1, k, S = 1, n_{new}) \\ &\quad \cdot p(n_{new}|B, C = 1, k, S = 1) \end{aligned} \quad (37)$$

where $p(n_{new}|B, C = 1, k, S = 1)$ can be found from expression (32).

Similar to (33), we represent the probability of received packets $n_r = \alpha > 0$, under condition $(B, C = 1, k, S = 1, n_{new})$, as

$$p(A_\alpha|B, C = 1, k, S = 1, n_{new}) = p(\bar{n}_r = \alpha|N_{status}). \quad (38)$$

From the above derivations, we formulate the final expression for finding the probability $p(J_i + j_i + \alpha, \tau + 1|J_i + j_i, \tau)$ as shown in (5).

APPENDIX C
DERIVATION OF (12)

To derive a general expression of $p(n_{new}|j_i, k)$ for n_{new} new packet(s) in the cluster of k nodes, let d denote the number of packets received by node n from the nodes in the cluster, i.e., $d(\in \{0, 1, \ldots, \min\{j_i, (k-1)J_i\})$. Excluding the packets of file i cached by node n, total number of cached packets of file i in the network and that in the cluster are denoted by $e = (N-1)J_i$ and $h = (k-1)J_i$, respectively.

We consider the following four cases to derive expression for $p(n_{new}|j_i, k)$ as

Case 1: $k = 1$

In this case, there is only one node in the cluster, which results in $h = 0$. So, we have $p(n_{new} \neq 0|j_i, k) = 0$ and $p(n_{new} = 0|j_i, k) = 1$.

Case 2: $e > h$ and $e - h < j_i - d$

In this case, the number of packets of file i cached outside the cluster is less than $j_i - d$, i.e., the number of packets of file i node n received from nodes outside the cluster, which should not happen. So $p(n_{new}|j_i, k) = 0$.

Case 3: $e > h$ and $e - h \geq j_i - d$

To find $p(n_{new}|j_i, k)$ in this case, we can write it as

$$\begin{aligned} p(n_{new}|j_i, k) &= p(n_{new} = h - d|j_i, k) \\ &= p(d = h - n_{new}|j_i, k) \end{aligned} \tag{39}$$

If there are some nodes outside the cluster and the total number of packets outside the cluster are greater than or equal to the number of packets node n received from outside cluster $(j_i - d)$ under the condition $j_i \geq d$ then we find expression for $p(d|j_i, k)$ as

$$p(d|j_i, k) = \frac{C_{N-1}^{k-1} \times C_h^d \times C_{e-h}^{j_i-d}}{C_{N-1}^{k-1} \times C_e^{j_i}} \tag{40}$$

where C_{N-1}^{k-1} represents the number of combinations of $k-1$ neighboring nodes of n in the cluster selected from the total $N-1$ nodes in the network (excluding node n), C_h^d represents the number of combinations of $d = (h - n_{new})$ packets received from the total h packets in the cluster, $C_{e-h}^{j_i-d}$ represents the number of combinations of $j_i - d$ packets received from the total $e-h$ packets of file i outside the cluster and $C_e^{j_i}$ represents the number of combinations of j_i packets received from the total e packets in the network.

Case 4: $e = h$

In this case, all the nodes are in the same cluster. Then $n_{new} = e - j_i$ and $d = j_i$. So $p(n_{new} = e - j_i|j_i, k) = 1$, which can also be derived from (40) by setting $e = h$ and $d = j_i$.

From the derivation of above four cases, we formulate a generalized expression as shown in (12).

REFERENCES

[1] A. Ahmed, H. Shan, and A. Huang, "D2D transmission modeling for mobile caching networks," submitted to *Proc. IEEE ICC*, Shanghai, China, May 2019.

[2] H. Liu, Z. Chen, X. Tian, X. Wang, and M. Tao, "On content-centric wireless delivery networks," *IEEE Wireless Commun.*, vol. 21, no. 6, pp. 118–125, Dec. 2014.

[3] GSM Association (GSMA) Intelligence, "The mobile economy 2015," GSMA, London, U.K., White Paper, 2015. [Online]. Available: https://www.gsma.com/mobileeconomy/global/2015

[4] Cisco Systems, Inc., "Cisco visual networking index: Global mobile data traffic forecast update," Cisco Syst. Inc., San Jose, CA, USA, White Paper 1454457600805266, Mar. 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html

[5] J. Erman, A. Gerber, M. Hajiaghayi, D. Pei, S. Sen, and O. Spatscheck, "To cache or not to cache: The 3G case," *IEEE Internet Comput.*, vol. 15, no. 2, pp. 27–34, Mar. 2011.

[6] S. Woo, E. Jeong, S. Park, J. Lee, S. Ihm, and K. Park, "Comparison of caching strategies in modern cellular backhaul networks," in *Proc. ACM MobiSys*, Jun. 2013, pp. 319–332.

[7] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, You Tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. ACM SIGCOMM Internet Meas.*, Oct. 2007, pp. 1–14.

[8] B. A. Ramanan, L. M. Drabeck, M. Haner, N. Nithi, T. E. Klein, and C. Sawkar, "Cacheability analysis of HTTP traffic in an operational LTE network," in *Proc. Wireless Telecommun. Symp.*, Apr. 2013, pp. 1–8.

[9] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "FemtoCaching: Wireless content delivery through distributed caching helpers," *IEEE Trans. Inf. Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[10] Z. Liu, M. Dong, B. Gu, C. Zhang, Y. Ji, and Y. Tanaka, "Fast-start video delivery in future Internet architectures with intra-domain caching," *Mobile Netw. Appl.*, vol. 22, no. 1, pp. 98–112, Feb. 2017.

[11] E. Nygren, R. K. Sitaraman, and J. Sun, "The Akamai network: A platform for high-performance Internet applications," *ACM SIGOPS Oper. Syst. Rev.*, vol. 44, no. 3, pp. 2–19, 2010.

[12] X. Wang, A. V. Vasilakos, M. Chen, Y. Liu, and T. T. Kwon, "A survey of green mobile networks: Opportunities and challenges," *Mobile Netw. Appl.*, vol. 17, no. 1, pp. 4–20, Feb. 2012.

[13] J. Xu, K. Ota, and M. Dong, "Saving energy on the edge: In-memory caching for multi-tier heterogeneous networks," *IEEE Commun. Mag.*, vol. 56, no. 5, pp. 102–107, May 2018.

[14] Z. Zhou, K. Ota, M. Dong, and C. Xu, "Energy-efficient matching for resource allocation in D2D enabled cellular networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 6, pp. 5256–5268, Jun. 2017.

[15] J. Long, M. Dong, K. Ota, and A. Liu, "A green TDMA scheduling algorithm for prolonging lifetime in wireless sensor networks," *IEEE Syst. J.*, vol. 11, no. 2, pp. 868–877, Jun. 2017.

[16] D. Li, M. Dong, Y. Yuan, J. Chen, K. Ota, and Y. Tang, "Seer-MCache: A prefetchable memory object caching system for IoT real-time data processing," *IEEE Internet Things J.*, vol. 5, no. 5, pp. 3648–3660, Oct. 2018.

[17] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.

[18] M. A. Maddah-Ali and U. Niesen, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *ACM/IEEE Trans. Netw.*, vol. 23, no. 4, pp. 1029–1040, Aug. 2015.

[19] F. Rezai and B. H. Khalaj, "Stability, rate, and delay analysis of single bottleneck caching networks," *IEEE Trans. Commun.*, vol. 64, no. 1, pp. 300–313, Jan. 2016.

[20] K. Li, C. Yang, Z. Chen, and M. Tao, "Optimization and analysis of probabilistic caching in N-tier heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 2, pp. 1283–1297, Feb. 2018.

[21] D. Liu and C. Yang, "Caching policy toward maximal success probability and area spectral efficiency of cache-enabled HetNets," *IEEE Trans. Commun.*, vol. 65, no. 6, pp. 2699–2714, Jun. 2017.

[22] B. Serbetci and J. Goseling, "On optimal geographical caching in heterogeneous cellular networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6.

[23] K. Doppler, M. Rinne, C. Wijting, C. B. Ribeiro, and K. Hugl, "Device-to-device communication as an underlay to LTE-advanced networks," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 42–49, Dec. 2009.

[24] J. Jiang, S. Zhang, B. Li, and B. Li, "Maximized cellular traffic offloading via device-to-device content sharing," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 82–91, Jan. 2016.

[25] M. Grossglauser and D. N. C. Tse, "Mobility increases the capacity of ad hoc wireless networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 4, pp. 477–486, Aug. 2002.

[26] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Commun. Mag.*, vol. 51, no. 4, pp. 142–149, Apr. 2013.

[27] N. Golrezaei, P. Mansourifard, A. F. Molisch, and A. G. Dimakis, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Trans. Wireless Commun.*, vol. 13, no. 7, pp. 3665–3676, Jul. 2014.

[28] X. Wang, M. Chen, T. Taleb, A. Ksentini, and V. C. M. Leung, "Cache in the air: Exploiting content caching and delivery techniques for 5G systems," *IEEE Commun. Mag.*, vol. 54, no. 52, pp. 131–139, Feb. 2014.

[29] C. Yang, Y. Yao, Z. Chen, and B. Xia, "Analysis on cache-enabled wireless heterogeneous networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 131–145, Jan. 2016.

[30] W. Wang, R. Lan, J. Gu, A. Huang, H. Shan, and Z. Zhang, "Edge caching at base stations with device-to-device offloading," *IEEE Access*, vol. 5, pp. 6399–6410, 2017.

[31] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 1, pp. 176–189, Jan. 2016.

[32] M. Ji, G. Caire, and A. F. Molisch, "Fundamental limits of caching in wireless D2D networks," *IEEE Trans. Inf. Theory*, vol. 62, no. 2, pp. 849–869, Feb. 2016.

[33] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[34] M. Ji, G. Caire, and A. F. Molisch, "The throughput-outage tradeoff of wireless one-hop caching networks," *IEEE Trans. Inf. Theory*, vol. 61, no. 12, pp. 6833–6859, Oct. 2015.

[35] G. Alfano, M. Garetto, and E. Leonardi, "Content-centric wireless networks with limited buffers: When mobility hurts," *ACM/IEEE Trans. Netw.*, vol. 24, no. 1, pp. 299–311, Feb. 2016.

[36] R. Wang, X. Peng, J. Zhang, and K. B. Letaief, "Mobility-aware caching for content-centric wireless networks: Modeling and methodology," *IEEE Commun. Mag.*, vol. 54, no. 8, pp. 77–83, Aug. 2016.

[37] S. Krishnan and H. S. Dhillon, "Effect of user mobility on the performance of device-to-device networks with distributed caching," *IEEE Wireless Commun. Lett.*, vol. 6, no. 2, pp. 194–197, Apr. 2017.

[38] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Mobility-aware caching in D2D networks," *IEEE Trans. Wireless Commun.*, vol. 16, no. 8, pp. 5001–5015, Aug. 2017.

[39] V. Conan, J. Leguay, and T. Friedman, "Fixed point opportunistic routing in delay tolerant networks," *IEEE J. Sel. Areas Commun.*, vol. 26, no. 5, pp. 773–782, Jun. 2008.

[40] T. Deng, G. Ahani, P. Fan, and D. Yuan, "Cost-optimal caching for D2D networks with user mobility: Modeling, analysis, and computational approaches," *IEEE Trans. Wireless Commun.*, vol. 17, no. 5, pp. 3082–3094, May 2018.

[41] R. Wang, J. Zhang, S. H. Song, and K. B. Letaief, "Exploiting mobility in cache-assisted D2D networks: Performance analysis and optimization," *IEEE Trans. Wireless Commun.*, vol. 17, no. 8, pp. 5592–5605, Aug. 2018.

[42] A. Shabani, S. P. Shariatpanahi, V. Shah-Mansouri, and A. Khonsari, "Mobility increases throughput of wireless device-to-device networks with coded caching," in *Proc. IEEE ICC*, May 2016, pp. 1–6.

[43] A. F. Molisch, *Wireless Communications*. New York, NY, USA: Wiley, 2011.

[44] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *IEEE Trans. Inf. Theory*, vol. 46, no. 2, pp. 388–404, Mar. 2000.

[45] M. J. Neely and E. Modiano, "Capacity and delay tradeoffs for ad hoc mobile networks," *IEEE Trans. Inf. Theory*, vol. 51, no. 6, pp. 1917–1937, Jun. 2005.

[46] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, "Delay and capacity in ad hoc mobile networks with *f*-cast relay algorithms," *IEEE Trans. Wireless Commun.*, vol. 10, no. 8, pp. 2738–2751, Aug. 2011.

[47] J. Liu, X. Jiang, H. Nishiyama, and N. Kato, "Throughput capacity of MANETs with power control and packet redundancy," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 3035–3047, Jun. 2013.

[48] T. Ho *et al.*, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413–4430, Oct. 2006.

[49] A. Malik, S. H. Lim, and W.-Y. Shin, "On the effects of subpacketization in content-centric mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 8, pp. 1721–1736, Aug. 2018.

[50] N. Jacobson, *Lectures Abstract Algebra: III. Theory Fields Galois Theory*, vol. 32. New York, NY, USA: Springer-Verlag, 2012.

[51] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, vol. 1, pp. 126–134, 1999.

[52] S. M. Ross, *Introduction to Probability Models*, 10th ed. New York, NY, USA: Academic, 2010.

[53] C. M. Grinstead and J. L. Snell, *Introduction to Probability*, 2nd ed. Providence, RI, USA: AMS, 1997.

[54] J. Liu and N. Kato, "A Markovian analysis for explicit probabilistic stopping-based information propagation in postdisaster ad hoc mobile networks," *IEEE Trans. Wireless Commun.*, vol. 15, no. 1, pp. 81–90, Jan. 2016.

AMEER AHMED received the M.S. degree in communication engineering from the Chalmers University of Technology, Gothenburg, Sweden, in 2010. He is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. He was with ZTE Pakistan, from 2005 to 2008, where he was a Senior Intranet Engineer. From 2011 to 2013, he was initially a Lecturer and then an Assistant Professor with COMSATS University Islamabad at Sahiwal Campus, Pakistan. He is currently with the Zhejiang Provincial Key Laboratory of Information Processing and Communication Networks, Hangzhou. His research interests are mainly in wireless caching networks, the performance evaluation of mobile ad hoc networks, and the Internet of Things.

HANGGUAN SHAN (M'10) received the B.Sc. degree in electrical engineering from Zhejiang University, Hangzhou, China, in 2004, and the Ph.D. degree in electrical engineering from Fudan University, Shanghai, China, in 2009. From 2009 to 2010, he was a Postdoctoral Research Fellow with the University of Waterloo, Waterloo, ON, Canada. Since 2011, he has been with the College of Information Science & Electronic Engineering, Zhejiang University, where he is currently an Associate Professor. He is also with the Zhejiang Provincial Key Laboratory of Information Processing & Communication Networks, Hangzhou. His current research interests include cross-layer protocol design, resource allocation, and the quality-of-service provisioning in wireless networks.

Dr. Shan has served as a Technical Program Committee Member of various international conferences, including the IEEE Global Communications Conference, the IEEE International Conference on Communications, the IEEE Wireless Communications and Networking Conference, and the IEEE Vehicular Technology Conference (VTC). He currently serves as the Track Co-Chair for green communications and networks track of the IEEE VTC 2016-Fall and the leading Track Co-Chair for future trends and emerging technologies track of the IEEE VTC 2017-Fall. He has also served as the Publicity Co-Chair for the third and fourth IEEE International Workshop on Wireless Sensor, Actuator, and Robot Networks, and the fifth International Conference on Wireless Communications and Signal Processing. He has coreceived the Best Industry Paper Award from the 2011 IEEE WCNC held in Quintana Roo, Mexico. He is currently an Editor of the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING.

AIPING HUANG (SM'08) received the degree from the Nanjing Institute of Posts and Telecommunications, Nanjing, China, in 1977, the M.S. degree from the Nanjing Institute of Technology, Southeast University, Nanjing, in 1982, and the Licentiate of Technology degree from the Helsinki University of Technology (HUT), Aalto University, Espoo, Finland, in 1997. From 1977 to 1980, she was an Engineer with the Design and Research Institute of Post and Telecommunications Ministry, China. From 1982 to 1994, she was initially an Assistant Professor and then an Associate Professor with Zhejiang University (ZJU), Hangzhou, China. From 1994 to 1998, she was a Visiting Scholar and a Research Scientist with HUT. Since 1998, she has been a Full Professor with ZJU. She is also with the Zhejiang Provincial Key Laboratory of Information Processing and Communication Networks, China.

She has published a book, and she has published over 210 papers in refereed journals and conferences on communications and networks, and signal processing. Her current research interests include heterogeneous networks, performance analysis and cross-layer design, and the planning and optimization of cellular mobile communication networks. She has served as the Vice Chair for the IEEE ComSoc Nanjing Chapter.