

# Delay Control in MANETs with Erasure Coding and $f$ -cast Relay

Bin Yang · Juntao Gao · Yuezhi Zhou · Xiaohong Jiang

**Abstract** Packet delay control in Mobile Ad Hoc Networks (MANETs) is critical to support delay-sensitive applications in such networks. By combining erasure coding and packet redundancy techniques, this paper proposes a general two-hop relay algorithm 2HR- $(x, \tau, f)$  for a flexible control of packet delivery delay in MANETs, where a group of  $x$  packets in source node are first encoded into  $x \cdot \tau$  encoded packets based on erasure coding, and each encoded packet is then delivered to at most  $f$  distinct relay nodes ( $f$ -cast) that will help to forward the encoded packet to destination node. To understand the delay performance in a 2HR- $(x, \tau, f)$  MANET, we then develop a discrete time multi-dimensional Markov chain model to depict the packet delivery process in the network, based on which closed-form results on mean and variance of packet delivery delay are further derived. Finally, extensive simulation and theoretical results are provided to illustrate

the efficiency of our delay models as well as the capability of the 2HR- $(x, \tau, f)$  algorithm in delay control.

**Keywords** Mobile ad hoc networks · Two-hop relay · Erasure coding · Packet redundancy · Delay

## 1 Introduction

The mobile ad hoc networks (MANETs) are a class of flexible and distributed peer-to-peer networks without any support from pre-existing infrastructures [1, 2]. As MANETs can be rapidly deployed, extended and re-configured at very low cost, they hold great promises for many important application scenarios, like as disaster relief, battle field communications, wide area sensing and surveillance, video stream and real time monitoring, VoIP, etc [3–6]. For an efficient support of these critical applications with stringent delay requirements in the future MANETs, flexible delay control in such networks has been a critical issue of research.

The erasure coding and packet redundancy are two promising techniques for packet delay control in MANETs, which have different impact on delay performance. Erasure coding technique allows a source node to first encode one packet into multiple equal-sized distinct code blocks and then distribute the code blocks to a number of relay nodes. The destination node can decode the original packet after receiving a certain number of its code blocks [7]. While packet redundancy technique (i.e. simple packet duplication) allows a source node to simply distribute redundant copies of a packet to multiple distinct relay nodes, which will help to finally forward the packet to its destination [9, 14].

The advantage and disadvantage of the two techniques have been shown in Literatures [7, 8] by numerical results. Specifically, erasure coding technique can considerably reduce the delay variance in MANETs,

---

B. Yang · X. Jiang  
School of Systems Information Science, Future University  
Hakodate, Hakodate 041-8655, Japan  
E-mail: yangbinchi@gmail.com

B. Yang  
School of Computer and Information Engineering, Chuzhou  
University, Chuzhou 239000, China

J. Gao  
School of Information Science, Nara Institute of Science and  
Technology, Nara 630-0192, Japan  
E-mail: gaojuntao223@gmail.com

X. Jiang  
E-mail: jiang@fun.ac.jp

Y. Zhou  
Department of Computer Science and Technology, Tsinghua  
University, Beijing 100084, China  
E-mail: zhouyz@mail.tsinghua.edu.cn

while it may lead to a relatively large packet delivery delay, since the early arrived code blocks in destination node have to wait a long time for the arrivals of other code blocks from distinct relay nodes. As for packet redundancy technique, it can efficiently reduce the packet delivery delay due to the fact that multiple relays will carry redundant copies of a packet, increasing the chance of the packet being received by its destination; however, it usually incurs high variance of packet delivery delay.

Available works on applying erasure coding and packet redundancy techniques for delay control in MANETs usually adopt these techniques separately (see Section 2 for related works), and they can not provide a flexible control for packet delay performance regarding delay and variance trade-off. The lack of a flexible delay control may significantly limit their ability to support various delay (and variance) sensitive applications in the future MANETs. To provide a flexible packet delay control in MANETs, we propose to jointly apply these two techniques together. To explore the delay performance under joint erasure coding and packet redundancy, we consider MANETs based on simple yet efficient two-hop relay routing. The simple two-hop relay, since was first introduced in the seminal work [10], has been proved to be an efficient routing protocol for packet delivery in MANETs. In two-hop relay, source node first transmits one packet to a mobile node (relay), and the relay then forward the packet only when it encounters the destination node. Since the source node may directly transmit a packet to its destination node once such transmission opportunity arises, each packet travels at most two hops to reach its destination node.

The main contributions of this paper are as follows.

- We first propose a 2HR- $(x, \tau, f)$  algorithm, where a group of  $x$  packets in a source node are first encoded into  $x \cdot \tau$  encoded packets based on general erasure coding, and each encoded packet is then replicated to at most  $f$  distinct relay nodes which will help to forward these encoded packets to their destination node. The original  $x$  packets can be simultaneously recovered at destination node whenever no less than  $x$  distinct encoded packets are received. The 2HR- $(x, \tau, f)$  algorithm is flexible in the sense the packet delivery process can be flexibly controlled through a proper setting of  $x$ ,  $\tau$  and  $f$ .
- To explore the packet delay of a 2HR- $(x, \tau, f)$  MANET, a discrete time multi-dimensional Markov chain framework is then developed to depict the packet delivery process in such network, based on which expected value and corresponding variance of packet delivery delay are derived.
- Finally, we provide extensive simulation and theoretical results to verify the efficiency of our Markov chain framework and related delay/variance models, and also to illustrate the capability of the 2HR- $(x, \tau, f)$  algorithm in delay control.

The rest of this paper is organized as follows. We introduce the related works in Section 2. System models are presented in Section 3. Section 4 introduces the erasure coding technique and 2HR- $(x, \tau, f)$  algorithm. A discrete time multi-dimensional Markov chain framework is developed in Section 5, based on which the expected value and variance of packet delivery delay are derived in Section 6. We provide numerical results in Section 7 to illustrate the efficiency of our delay models as well as the capability of the 2HR- $(x, \tau, f)$  algorithm in delay control. Finally, we conclude this paper in Section 8.

## 2 Related Works

### 2.1 Delay Control with Erasure Coding

It was first demonstrated through simulation study in [7, 27] that erasure coding technique can reduce delay variance and worst-case packet delivery delay in MANETs with opportunistic routing. By combining probabilistic routing and erasure coding, a novel routing algorithm was proposed in [28] to provide efficient delay control in opportunistic MANETs. Hanbali *et al.* [8] developed a simple theoretical model to analyze delay performance under two-hop relay and erasure coding in a very simple network scenario, where there is only one source-destination pair and the source node has only one single packet to be delivered. Also, a simple coding technique was considered [8], in which a single packet (message) is first divided into multiple blocks and these blocks are then encoded into code blocks for transmission. Later, Liu *et al.* [17] extended the work in [8] to a more general network scenario with multiple source-destination pairs. Recently, Chen *et al.* [29] tried to combine erasure coding and packet redundancy techniques for delay control in opportunistic MANETs. It is notable, however, similar to [8] that the algorithm in [29] is very simple in the sense that the algorithm allows only single packet to be encoded and decoded, and each code block there can have only two redundant copies. Also, only simulation results are provided in [29] for performance evaluation. More recently, Kong *et al.* [30] employed Reed-Solomon codes (erasure coding) to achieve a better throughput-delay trade-off in two-hop relay MANETs. Altman *et al.* [31,32] also proposed an erasure coding-based algorithm for delay control so as to improve packet delivery performance in

delay tolerant networks, i.e., a special class of sparse MANETs with ignoring interference among transmissions.

## 2.2 Delay Control with Packet Redundancy

Applying packet redundancy technique for delay control in MANETs has been explored under various mobility models, like under the i.i.d. mobility model in [14], under the Brownian mobility in [33], as well as under the hybrid random walk model and discrete random direction model in [34]. Delay performance modeling under packet redundancy technique has also been extensively studied recently. The works [35–37] conducted delay modeling under a simple network scenario, where only one source-destination pair is available in the network. Later, Liu *et al.* [9, 38] explored delay modeling under more general network scenarios with multiple source-destination pairs.

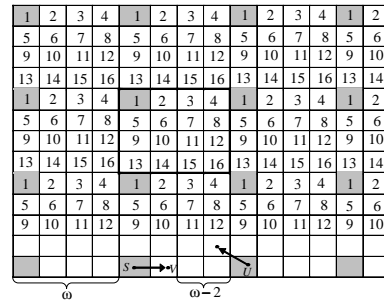
Recently, lot of research efforts have been devoted to the study of packet redundancy-based delay control in DTNs. Spyropoulos *et al.* [39] proposed a single period routing algorithm (called spray and wait) for delay control in DTNs, and Bulut *et al.* [40] extended the algorithm in [39] and further proposed a more general multi-period spraying algorithm for delay control in DTNs. Panagakos *et al.* [36] developed a theoretical framework for delay modeling in DTNs with packet redundancy-based delay control.

The aforementioned works for delay control in MANETs mainly adopt erasure coding and packet redundancy techniques separately, and they can not provide a flexible control for packet delay performance to support various applications with stringent delay/variance requirements. Different from existing works, this paper proposes a new scheme which jointly applies erasure coding and packet redundancy techniques together to provide a flexible control of delay/variance in MANETs. We also develop a discrete time multi-dimensional Markov chain model to depict the packet delivery process under the new scheme, based on which closed-form results for mean and variance of packet delivery delay are derived.

## 3 System Models

### 3.1 Network Model and Communication Model

Similar to previous works [11–16], we consider a time-slotted network with a square region of unit torus, where the top (left) and bottom (right) edges of the region are identical to each other. As shown in Fig. 1,



**Fig. 1** Illustration of network cell partition and transmission-group.

the network is evenly divided into  $m \times m$  cells. According to the widely adopted independent and identically distributed (i.i.d.) mobility model [11–16], at the beginning of a time slot, each of  $n$  nodes independently and uniformly chooses a cell among all  $m^2$  cells with equal probability of  $1/m^2$  to move in, and stays it for the whole time slot. Each node repeats this process in every subsequent time slot.

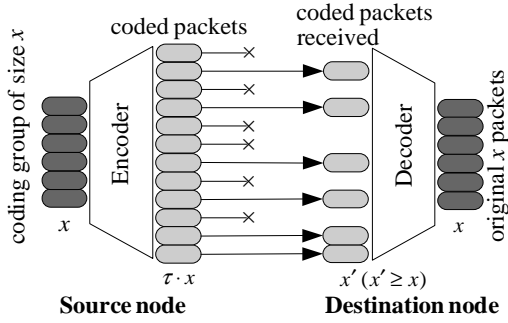
Regarding the traffic model, we consider the widely used permutation traffic pattern [17–19], where each node is the source of one flow and the destination of another flow. Here, one flow corresponds to one source-destination ( $S$ - $D$ ) pair. Without loss of generality, we assume  $n$  source-destination pairs are as follows:  $1 \rightarrow 2, \dots, i \rightarrow i+1, \dots, n \rightarrow 1$ , where the destination of node  $i$  is node  $i+1$ , and the destination of node  $n$  is node 1. We assume that the total number of bits that can be transmitted between a node pair is normalized as one packet per time slot. The protocol model [20] is adopted here to account for interference among simultaneous transmissions. Suppose that node  $i$  is trying to transmit packet to another node  $j$  at time slot  $t$ , and we use  $d_{ij}(t)$  to denote the Euclidean distance between  $i$  and  $j$  at the time slot. To guarantee the successful transmission from  $i$  to  $j$ , for any other node  $k$  that is simultaneously transmitting with node  $i$ , the following condition should be satisfied.

$$d_{kj}(t) \geq (1 + \Delta)d_{ij}(t), \quad (1)$$

where  $\Delta$  is the guarding factor for interference prevention.

### 3.2 Transmission Scheduling

To schedule as many simultaneous transmissions as possible, we consider here a transmission-group based scheduling scheme [9, 21]. As illustrated in Fig.1, with transmission-group scheduling of parameter  $\omega$ , all cells are divided into  $\omega^2$  distinct transmission-groups labeled as group 1 to group  $\omega^2$  (e.g., all shaded cells



**Fig. 2** Illustration of erasure coding with replication factor  $\tau \geq 1$ .

in Fig.1 belong to the group 1), where any two cells in a transmission-group have a horizontal and vertical distance of some multiple of  $\omega$  cells away. We assume that each transmission-group (and thus each cell in the group) can get transmission opportunity in turn in every  $\omega^2$  time slots, and call a cell an active cell if it gets transmission opportunity.

To ensure that all the cells in a transmission-group can transmit simultaneously without interfering each other, we need to properly set the parameter  $\omega$ . We consider a local transmission scenario [10], in which a node in an active cell can transmit to another node in the same cell or in its eight adjacent cells. Thus, the maximum transmission range  $r$  of a node is set as  $r = \sqrt{8}/m$ . Suppose that a node  $S$  in an active cell is selected to transmit to another node  $V$ . As illustrated in Fig.1, any other transmitting node  $U$  in the same transmission-group is at least  $(\omega - 2)/m$  away from  $V$ . According to the protocol model [20], to ensure that the transmission to node  $V$  is successful, the following condition should be satisfied:  $(\omega - 2)/m \geq (1 + \Delta) \cdot r$ . Notice that  $\omega$  is an integer and  $\omega \leq m$ , we set

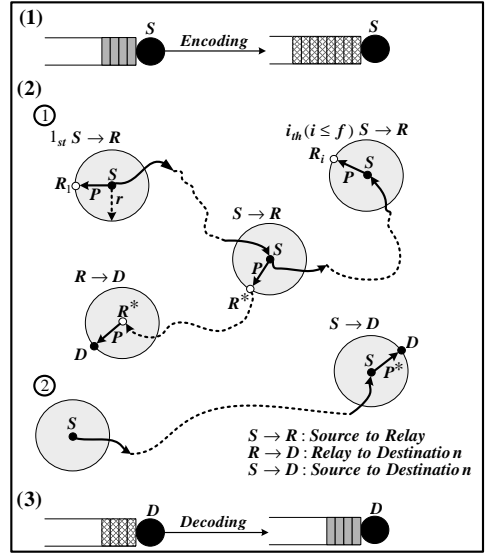
$$\omega = \min\{[(1 + \Delta)\sqrt{8}] + 2, m\}. \quad (2)$$

#### 4 Erasure Coding and 2HR- $(x, \tau, f)$ Algorithm

In this section, we first introduce the erasure coding technique, and then present our 2HR- $(x, \tau, f)$  algorithm.

##### 4.1 Erasure Coding

The main idea of erasure coding with replication factor  $\tau \geq 1$  is shown in Fig. 2, where a coding group of  $x$  packets in source node are first encoded into  $\tau \cdot x$  equal-sized coded packets, and these  $x$  packets can then



**Fig. 3** Illustration of the 2HR- $(x, \tau, f)$  algorithm for a tagged source-destination pair. (1) and (3) denote the encoding and decoding processes at  $S$  and  $D$ , respectively. (2) denotes the packet delivery process, where ① illustrates that  $S$  is transmitting coded packet  $P$  to  $D$  with the help of relay nodes; ② illustrates that  $S$  is directly transmitting coded packet  $P^*$  to  $D$ .

be decoded at destination node when  $x' \geq x$  distinct coded packets are received [7].

We use one simple example here to illustrate the basic encoding and decoding processes in erasure coding. For a coding group  $(s_1, s_2, s_3)^T$  of three packets  $s_1, s_2$  and  $s_3$ , we encode them into six coded packets  $(c_1, c_2, \dots, c_6)^T$  with replication factor  $\tau = 2$  as

$$(c_1, c_2, \dots, c_6)^T = \mathbf{G} \cdot (s_1, s_2, s_3)^T, \quad (3)$$

here  $\mathbf{G}$  is a 6-by-3 generator matrix of the erasure coding. Suppose that coded packets  $c_2, c_3$  and  $c_5$  have been received at destination node, then we have

$$(c_2, c_3, c_5)^T = \mathbf{G}' \cdot (s_1, s_2, s_3)^T, \quad (4)$$

where  $\mathbf{G}'$  is a 3-by-3 submatrix composed of the 2th, 3th and 5th rows of matrix  $\mathbf{G}$ . Based on the property of  $\mathbf{G}$  that a submatrix composed of any of its 3 rows will be an invertible matrix [22], we know that  $\mathbf{G}'$  is invertible. Thus, the original packets  $s_1, s_2$  and  $s_3$  can then be decoded as

$$(s_1, s_2, s_3)^T = (\mathbf{G}')^{-1} \cdot (c_2, c_3, c_5)^T. \quad (5)$$

##### 4.2 2HR- $(x, \tau, f)$ Algorithm

Without loss of generality, we focus on one source-destination pair with source node  $S$  and destination

node  $D$  in our discussion. Fig. 3 shows the mechanism of the 2HR- $(x, \tau, f)$  algorithm, including the processes of erasure coding, packet delivery and decoding. For a specified coding group, the source node  $S$  first encodes  $x$  packets into multiple distinct coded packets, and then  $S$  will distribute redundant copies for each coded packet (e.g., coded packet  $P$ ) to at most  $f$  distinct relay nodes, and these relay nodes (also source node  $S$ ) will finally deliver each coded packet to the destination node  $D$ . After receiving  $x$  distinct coded packets of the coding group,  $D$  can finally decoded the packets group. To simplify the analysis, we assume that each relay node will carry at most one coded packet for any particular coding group. Before introducing the new 2HR- $(x, \tau, f)$  algorithm, we first define the following terms.

- **New coded packet and non-new coded packet:** A coded packet is called a new coded packet if it has not been received yet by its destination; otherwise, it is a non-new coded packet.
- **Helping-node and candidate-node:** A relay node is called a helping-node of a specified coding group if it carries a new coded packet of the coding group; otherwise, it is called a candidate-node.
- **Main-queue:**  $S$  maintains a main-queue to store coded packets of the packets generated at  $S$ , which will be replicated to relays later.
- **Backup-queue:**  $S$  maintains a backup-queue to store its coded packets whose  $f$  copies have been sent out but their reception at  $D$  has not been confirmed yet.
- **Relay-queue:**  $S$  (as a relay node) also maintains  $n - 2$  relay-queues for other  $n - 2$  source-destination pairs to store their coded packets (one queue per source-destination pair).

Based on above definitions, the new 2HR- $(x, \tau, f)$  algorithm is summarized in Algorithm 1.

Notice that in the above relay-to-destination transmission, node  $S$  acts as a relay that helps to forward coded packets to destinations for other  $n - 2$  source-destination pairs. Regarding the traffic model in 2HR- $(x, \tau, f)$  algorithm, there exist in total  $n$  flows, each of which corresponds to one source-destination pair, since there are  $n$  mobile nodes in the network and each node is the source of one flow and the destination of another flow. Each node can be a potential relay for other  $n - 2$  flows (except the two flows originated from and destined for itself).

## 5 Markov Chain Model

To depict the packet delivery process under the 2HR- $(x, \tau, f)$  algorithm, we adopt a three-tuple  $(i, j, k)$

---

**Algorithm 1** 2HR- $(x, \tau, f)$  Algorithm:

---

**Encoding:**

Source  $S$  encodes a group of  $x$  packets into  $\tau \cdot x$  coded packets that are stored into its main-queue.

**Delivery:**

1. **if**  $S$  gets a transmission opportunity at a time slot **then**
2.     **if**  $D$  is within the transmission range of  $S$  **then**
3.          $S$  executes Procedure 1;
4.     **else**
5.          $S$  selects to perform source-to-relay transmission or relay-to-destination transmission with equal probability;
6.         **if**  $S$  schedules a source-to-relay transmission **then**
7.              $S$  executes Procedure 2;
8.         **else if**  $S$  schedules a relay-to-destination **then**
9.              $S$  executes Procedure 3;
10.         **end if**
11.     **end if**
12. **end if**

**Decoding:**

Destination  $D$  will decode the group of  $x$  packets when it receives  $x$  distinct coded packets of the group;

---



---

**Procedure 1** Source-to-destination transmission:

---

1.  $S$  initiates a handshake to check which coded packets of the coding group have been received by  $D$ .
2. **if** the head-of-line coded packet  $P_h$  in main-queue is a new coded packet **then**
3.      $S$  transmits  $P_h$  to  $D$ ;
4. **else if** there exists a new coded packet waiting behind  $P_h$  in main-queue **then**
5.      $S$  transmits the coded packet to  $D$ ;
6. **else if** there exists a new coded packet in backup-queue of  $S$  **then**
7.      $S$  transmits the coded packet to  $D$ ;
8. **end if**

$S$  deletes all the non-new coded packets in its main-queue and backup-queue;

---

to denote general transient state for coded packets of a coding group, where source  $S$  is delivering the  $j$ th ( $1 \leq j \leq f$ ) copy of the  $i$ th ( $1 \leq i \leq \tau \cdot x$ ) coded packet of the group, and destination  $D$  has received  $k$  ( $0 \leq k < x, k \leq i$ ) of  $\tau \cdot x$  coded packets. We further use to  $(*, *, k)$  to denote the transient state that  $S$  has already finished dispatching all copies of  $\tau \cdot x$  coded packets while  $D$  has only received  $k$  ( $0 \leq k < x$ )

**Procedure 2** Source-to-relay transmission:

1.  $S$  randomly selects a node as relay node  $R$  within its transmission range;
2. **if**  $R$  is a candidate-node **then**
3.  $S$  transmits a copy of head-of-line coded packet  $P_h$  in its main-queue to  $R$ ;
4. **if**  $f$  copies of  $P_h$  have already been delivered out **then**
5.  $S$  puts  $P_h$  to the end of its backup-queue, and then moves ahead remaining coded packets in its main-queue;
6. **end if**
7. **else**
8.  $S$  keeps idle at this time slot;
9. **end if**

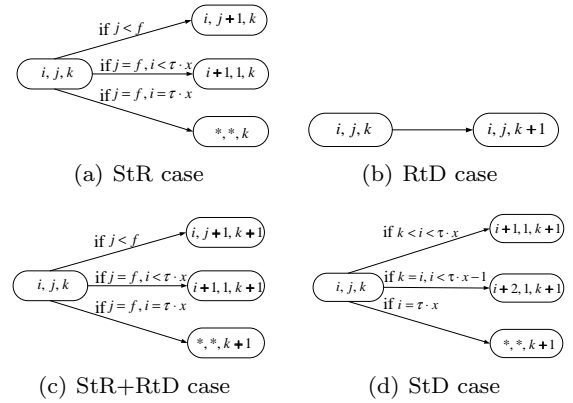
**Procedure 3** Relay-to-destination transmission:

1.  $S$  randomly selects a node as destination node  $V$  within its transmission range;
2.  $S$  initiates a handshake to check which coded packets of the coding group that  $V$  is requesting have been received by  $V$ .
3. **if** there exists a new coded packet of the coding group in its relay-queue specified for  $V$  **then**
4.  $S$  transmits the coded packet to  $V$ ;
5. **else**
6.  $S$  keeps idle at this time slot;
7. **end if**

$S$  deletes all non-new coded packets destined for  $V$  from its relay-queue;

distinct coded packets of them. Suppose that current transient state is  $(i, j, k)$ , based on 2HR- $(x, \tau, f)$  algorithm we can see that only one of the following four transmission cases will happen in the next time slot.

- *StR* case: Source-to-relay transmission, i.e.,  $S$  successfully delivers the  $j$ th copy of the  $i$ th coded packet to a candidate-node. As shown in Fig. 4(a), under the *StR* case, the state  $(i, j, k)$  can transit to any of its three neighboring states depending on indexes  $i$  and  $j$ .
- *RtD* case: Relay-to-destination transmission, i.e., a helping-node successfully delivers a new coded packet to  $D$ . As shown in Fig. 4(b), under the *RtD* case, the state  $(i, j, k)$  can only transit to state  $(i, j, k + 1)$ .
- *StR + RtD* case: Both source-to-relay transmission and relay-to-destination transmission happen simultaneously. As shown in Fig. 4(c), under the *StR + RtD* case, the state  $(i, j, k)$  can transit to any state of  $(i, j + 1, k + 1)$ ,  $(i + 1, 1, k + 1)$  and  $(*, *, k + 1)$ .



**Fig. 4** The transition diagrams of the state  $(i, j, k)$ , where  $1 \leq i \leq \tau \cdot x$ ,  $1 \leq j \leq f$ , and  $0 \leq k < x$ ,  $k \leq i$ .

- *StD* case: Source-to-destination transmission, i.e.,  $S$  successfully delivers a new coded packet to  $D$ . As shown in Fig. 4(d), under the *StD* case, the state  $(i, j, k)$  can transit to any of states  $(i + 1, 1, k + 1)$ ,  $(i + 2, 1, k + 1)$  and  $(*, *, k + 1)$ , depending on indexes  $i$  and  $k$ .

Notice that the source  $S$  always delivers out coded packet sequentially, thus a coded packet delivered out earlier from its source  $S$  will be likely received early at its destination  $D$ . To simplify the analysis, under the *StD* case we assume that for the transient state  $(i, j, k)$  with  $k < i < \tau \cdot x$ ,  $S$  is delivering the  $i$ th coded packet but less than  $i$  distinct coded packets have been received by  $D$ . Thus, under the *StD* case in Fig.4(d), the transient state  $(i, j, k)$  will always transit to the state  $(i + 1, 1, k + 1)$  when  $k < i < \tau \cdot x$ .

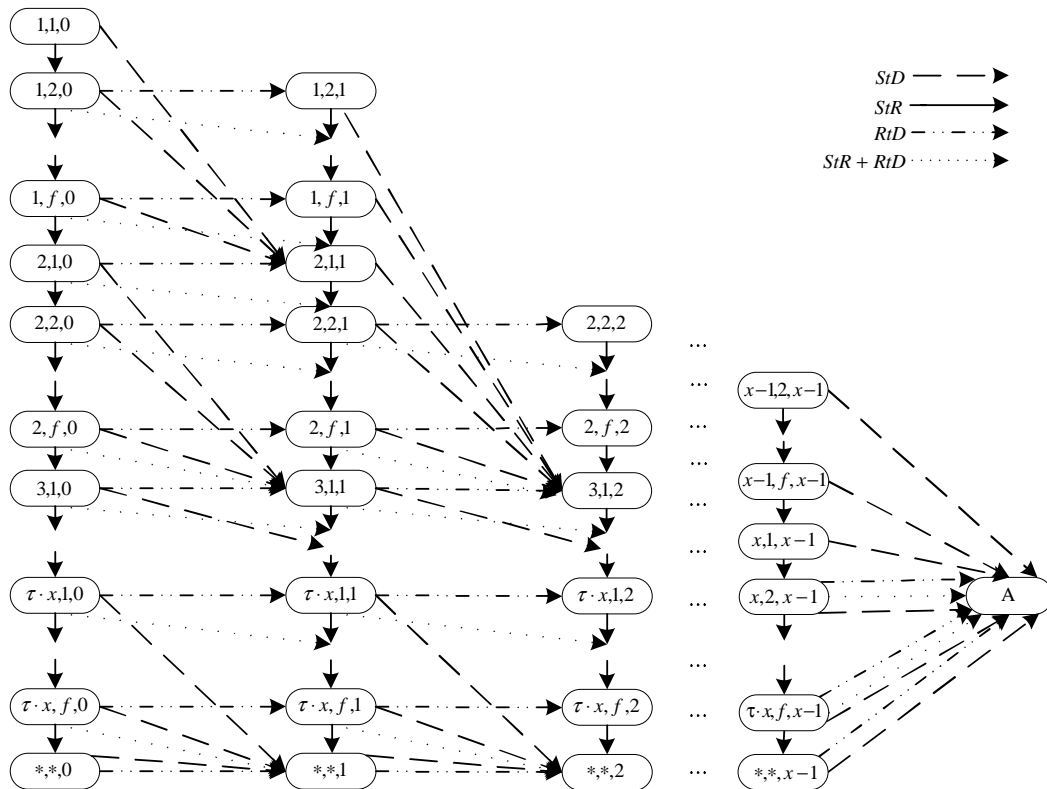
Based on the transient states in Fig.4, the packet delivery process under the 2HR- $(x, \tau, f)$  algorithm can be depicted by a discrete time multi-dimensional Markov chain model shown in Fig.5, where  $A$  denotes the absorbing state that destination  $D$  has received  $x$  distinct coded packets of the specified coding group.

As illustrated in Fig.5, we denote by  $\alpha$  the total number of transient states in the Markov chain model, then  $\alpha$  is determined as

$$\alpha = (2\tau x^2 - x^2 + 3x - 2) \cdot f / 2 + 1, \quad (6)$$

where all  $\alpha$  transient states are arranged into  $x$  columns. We number these transient states sequentially as  $1, 2, 3, \dots, \alpha$ , and number the absorbing state  $A$  as  $\alpha + 1$ , in a top-to-down and left-to-right way. Thus, the number of transient states  $c_k$  in the  $k$ th column ( $0 \leq k \leq x - 1$ ) can be determined as

$$c_k = \begin{cases} \tau x \cdot f + 1 & \text{if } k = 0, \\ (\tau x + 1 - k) \cdot f & \text{if } 1 \leq k \leq x - 1. \end{cases} \quad (7)$$



**Fig. 5** Absorbing Markov chain for the 2HR- $(x, \tau, f)$  algorithm. For simplicity, the transition back to each transition state itself is not shown.

For the  $l_{th}$  transient state of the  $k_{th}$  column in Fig.5,  $l \in [1, c_k]$ ,  $k \in [0, x - 1]$ , the number of helping-nodes  $u_h$  and the number of candidate-nodes  $u_c$  can be determined as:

- When  $k = 0$

$$u_h = l - 1, \quad (8)$$

$$u_c = n - l - 1. \quad (9)$$

- When  $k \in [1, x - 1]$

$$u_h = \begin{cases} 0 & \text{if } l < f, \\ l - f & \text{if } l \geq f, \end{cases} \quad (10)$$

$$u_c = \begin{cases} n - 2 & \text{if } l < f, \\ n - 2 - l + f & \text{if } l \geq f, \end{cases} \quad (11)$$

## 6 Packet Delivery Delay Modeling

Based on the Markov chain model in Fig.5, we proceed to analyze the packet delivery delay and related delay variance under the 2HR- $(x, \tau, f)$  algorithm. Regarding the packet delivery delay under the 2HR- $(x, \tau, f)$  algorithm, we have the following definition.

*Delivery Delay*<sup>1</sup>: For a specified coding group, the delivery delay of a packet in it is defined as the time duration starting from the time slot when source  $S$  starts to replicate the first coded packet of the group to the time slot when destination  $D$  has received  $x$  distinct coded packets of the group.

*Remark 1* With 2HR- $(x, \tau, f)$  routing, packets of a coding group are first encoded together as encoded packets, so essentially they are dispatched from  $S$  at the same time and also they are received by  $D$  at the same time (i.e., when  $x$  distinct coded packets are received). Thus, each packet of a coding group experiences the same delivery delay defined above.

### 6.1 Expected Packet Delivery Delay and Delay Variance

For the Markov chain model in Fig.5, we use random variable  $t_k$  to denote the time it takes for the chain

<sup>1</sup> The packet delivery delay in MANETs is mainly dominated by node mobility, interference and medium contention [30, 42]. Nowadays, computation power is very powerful, making coding/decoding to be processed very fast and thus allowing us to neglect the time cost of coding/decoding process in our delay analysis [7, 8, 30]

to reach the absorbing state  $A$  starting from the  $k_{th}$  transient state ( $1 \leq k \leq \alpha$ ). Thus, the expected value  $E\{t_1\}$  of  $t_1$  just corresponds to the expected packet delivery delay under the 2HR- $(x, \tau, f)$  algorithm.

To derive  $E\{t_1\}$ , we first need to determine the values of vector  $\mathbf{t} = (E\{t_1\}, E\{t_2\}, \dots, E\{t_\alpha\})^T$ . Using the first step analysis, we have

$$E\{t_k\} = \sum_{l=1}^{\alpha+1} q_{kl}(1 + E\{t_l\}) = 1 + \sum_{l=1}^{\alpha} q_{kl}E\{t_l\} \quad (12)$$

where  $q_{ij}$  denotes the transition probability from the  $i_{th}$  state to the  $j_{th}$  state. Notice that  $E\{t_l\} = 0$  when  $l = \alpha + 1$ .

We define a matrix  $\mathbf{Q} = (q_{ij})_{(\alpha+1) \times (\alpha+1)}$  and a sub-matrix  $\mathbf{P}$  consisting of rows 1 through  $\alpha$  and columns 1 through  $\alpha$  of matrix  $\mathbf{Q}$ . Then, we can rewrite (12) as

$$\mathbf{t} = (1, 1, \dots, 1)^T + \mathbf{P}\mathbf{t}. \quad (13)$$

Thus, we have

$$\mathbf{t} = (\mathbf{I} - \mathbf{P})^{-1} \cdot (1, 1, \dots, 1)^T, \quad (14)$$

where  $\mathbf{I}$  denotes a  $\alpha$ -by- $\alpha$  identity matrix.

Let  $\mathbf{N}$  denote the fundamental matrix of the Markov chain in Fig.5. According to Markov chain theory [23], we have

$$\mathbf{N} = (\mathbf{I} - \mathbf{P})^{-1}. \quad (15)$$

By substituting (15) into (14), we have

$$E\{t_1\} = \sum_{i=1}^{\alpha} \mathbf{N}(1, i), \quad (16)$$

where  $\mathbf{N}(1, i)$  denotes the  $(1, i)$ -entry of  $\mathbf{N}$ .

We proceed to derive variance  $Var\{t_1\}$  of packet delivery delay. Since

$$Var\{t_1\} = E\{t_1^2\} - (E\{t_1\})^2, \quad (17)$$

we need to determine  $E\{t_1^2\}$  to obtain  $Var\{t_1\}$ . Notice that

$$\begin{aligned} E\{t_i^2\} &= \sum_{l=1}^{\alpha+1} q_{il}E\{(1 + t_l)^2\} \\ &= \sum_{l=1}^{\alpha} q_{il}E\{t_l^2\} + 2 \sum_{l=1}^{\alpha} q_{il}E\{t_l\} + 1. \end{aligned} \quad (18)$$

We define  $\mathbf{t}_{sq} = (E\{t_1^2\}, E\{t_2^2\}, \dots, E\{t_\alpha^2\})^T$ , then we can rewrite (18) as

$$\mathbf{t}_{sq} = \mathbf{P} \cdot \mathbf{t}_{sq} + 2\mathbf{P} \cdot \mathbf{t} + (1, 1, \dots, 1)^T. \quad (19)$$

Combining (14), (15) and (19), we obtain

$$\mathbf{t}_{sq} = (2\mathbf{N} \cdot \mathbf{P} \cdot \mathbf{N} + \mathbf{N}) \cdot (1, 1, \dots, 1)^T. \quad (20)$$

Thus,  $E\{t_1^2\}$  can be evaluated based on (20).

We can see from (16), (17) and (20) that we need to determine  $\mathbf{P}$  and  $\mathbf{N}$  for the evaluation of  $E\{t_1\}$  and  $Var\{t_1\}$ .

## 6.2 Derivation of Matrix $\mathbf{P}$

To simplify the calculation, we arrange  $\mathbf{P}$  as the following partitioned matrices

$$\mathbf{P} = \begin{bmatrix} \mathbf{P}_0 & \mathbf{P}'_0 & & & \\ & \mathbf{P}_1 & \mathbf{P}'_1 & & \\ & & \ddots & \ddots & \\ & & & \mathbf{P}_k & \mathbf{P}'_k \\ & & & & \ddots & \ddots \\ & & & & & \mathbf{P}_{x-2} & \mathbf{P}'_{x-2} \\ & & & & & & \mathbf{P}_{x-1} \end{bmatrix}, \quad (21)$$

where  $\{\mathbf{P}_k\}$  and  $\{\mathbf{P}'_k\}$  denote the main diagonal and upper diagonal blocks (sub-matrices) of  $\mathbf{P}$ , and all other blocks are zero matrices and thus are omitted here. The block  $\mathbf{P}_k$  of size  $c_k \times c_k$  defines the transient probabilities among the transient states of the  $k_{th}$  column in the Markov chain model, while the block  $\mathbf{P}'_k$  of size  $c_k \times c_{k+1}$  defines the transient probabilities from the transient states of the  $k_{th}$  column to that of the  $(k+1)_{th}$  column in the Markov chain model.

We first establish the following lemmas regarding some basic probabilities in the Markov chain model of Fig.5, which will help us to derive the matrix  $\mathbf{P}$ .

**Lemma 1** For a time slot and a given  $S$ - $D$  pair, let  $p_0$  denote the probability that  $S$  is scheduled to conduct  $StD$  transmission, and let  $p_1$  denote the probability that  $S$  is scheduled to conduct  $StR$  transmission or  $RtD$  transmission. Then we have

$$p_0 = \frac{1}{w^2} \left( \frac{9n - m^2}{n(n-1)} - \left( 1 - \frac{1}{m^2} \right)^{n-1} \frac{8n + 1 - m^2}{n(n-1)} \right), \quad (22)$$

$$\begin{aligned} p_1 &= \frac{1}{w^2} \left( \frac{m^2 - 9}{n-1} \left( 1 - \left( 1 - \frac{1}{m^2} \right)^{n-1} \right) \right. \\ &\quad \left. - \left( 1 - \frac{9}{m^2} \right)^{n-1} \right). \end{aligned} \quad (23)$$

**Lemma 2** For a given  $S$ - $D$  pair, suppose that at current time slot there are  $h$  helping-nodes and  $c$  candidate-nodes. For the next time slot, we use  $p_{rev}(h)$ ,  $p_{dev}(c)$



and  $p_{sim}(h, c)$  to denote the probability that destination  $D$  will receive a new coded packet, the probability that  $S$  will successfully deliver out a coded packet to a candidate-node and the probability of simultaneous  $StR$  and  $RtD$  transmissions, respectively. Then we have

$$p_{rev}(h) = p_0 + \frac{h}{2(n-2)}p_1, \quad (24)$$

$$p_{dev}(c) = \frac{c}{2(n-2)}p_1, \quad (25)$$

$$p_{sim}(h, c) = \frac{hc(m^2 - \omega^2)}{4m^2\omega^4} \sum_{k=0}^{n-5} \binom{n-5}{k} \psi(k) \cdot \left\{ \sum_{t=0}^{n-k-4} \binom{n-k-4}{t} \psi(t) \left(1 - \frac{18}{m^2}\right)^{n-k-t-4} \right\}, \quad (26)$$

where

$$\psi(\theta) = \frac{9\left(\frac{9}{m^2}\right)^{\theta+1} - 8\left(\frac{8}{m^2}\right)^{\theta+1}}{(\theta+1)(\theta+2)}. \quad (27)$$

The proof of lemma 1 and lemma 2 is similar to that in [9], so we omit it here. Based on the results of above lemmas, we can determine matrix  $\mathbf{P}$  as follows.

- When  $k = 0$ , the non-zero entries of  $\mathbf{P}_0$  and  $\mathbf{P}'_0$  can be determined as

$$\mathbf{P}_0(i, i) = \begin{cases} 1 - p_{rev}(\tau x \cdot f) & \text{if } i = c_0, \\ 1 - p_{dev}(u_c) - p_{rev}(u_h) & \\ + p_{sim}(u_h, u_c) & \text{if } i \in [1, c_0), \end{cases} \quad (28)$$

$$\mathbf{P}_0(i, i+1) = p_{dev}(u_c) - p_{sim}(u_h, u_c) \quad \text{if } i \in [1, c_0), \quad (29)$$

$$\mathbf{P}'_0(i, i) = p_{sim}(u_h, u_c) \quad \text{if } i \in [2, c_0), \quad (30)$$

$$\mathbf{P}'_0(i, i-1) = \begin{cases} p_{rev}(\tau x \cdot f) & \text{if } i = c_0, \\ p_{rev}(u_h) - p_0 & \\ -p_{sim}(u_h, u_c) & \text{if } i \in [2, c_0), \end{cases} \quad (31)$$

$$\mathbf{P}'_0(i, f \cdot \lceil \frac{i}{f} \rceil) = p_0 \quad \text{if } i \in [1, c_0). \quad (32)$$

- When  $k \in [1, x-1]$ , the non-zero entries of  $\mathbf{P}_k$  can be determined as

$$\mathbf{P}_k(i, i) = \begin{cases} 1 - p_0 - p_{dev}(u_c) & \text{if } i \in [1, f], \\ 1 - p_{dev}(u_c) - p_{rev}(u_h) & \\ + p_{sim}(u_h, u_c) & \text{if } i \in [f+1, c_k), \\ 1 - p_{rev}(u_h) & \text{if } i = c_k, \end{cases} \quad (33)$$

$$\mathbf{P}_k(i, i+1) = \begin{cases} p_{dev}(u_c) & \text{if } i \in [1, f], \\ p_{dev}(u_c) - p_{sim}(u_h, u_c) & \text{if } i \in [f+1, c_k). \end{cases} \quad (34)$$

- When  $k \in [1, x-2]$ , the non-zero entries of  $\mathbf{P}'_k$  can be determined as

$$\mathbf{P}'_k(i, i-f+1) = p_{sim}(u_h, u_c) \quad \text{if } i \in [f+1, c_k), \quad (35)$$

$$\mathbf{P}'_k(i, i-f) = \begin{cases} p_{rev}(u_h) & \text{if } i = c_k, \\ p_{rev}(u_h) - p_0 & \\ -p_{sim}(u_h, u_c) & \text{if } i \in [f+1, c_k), \end{cases} \quad (36)$$

$$\mathbf{P}'_k(i, f) = p_0 \quad \text{if } i \in [1, f], \quad (37)$$

$$\mathbf{P}'_k(i, f \cdot \lceil \frac{i}{f} \rceil) = p_0 \quad \text{if } i \in [f+1, c_k). \quad (38)$$

### 6.3 Derivation of the Matrix $\mathbf{N}$

For the Markov chain model in Fig.5, we can actually partition the fundamental matrix  $\mathbf{N}$  into  $x$ -by- $x$  blocks  $\mathbf{N} = (\mathbf{N}_{ij})_{x \times x}$ , where the block  $\mathbf{N}_{ij}$  corresponds to the expected number of times in the transient states of the  $(j-1)_{th}$  column of the Markov chain model given that Markov chain starts from the transient states of the  $(i-1)_{th}$  column. We define a matrix  $\mathbf{H} = \mathbf{I} - \mathbf{P}$ , so we obtain  $\mathbf{H}^{-1} = \mathbf{N}$ . Since  $\mathbf{H}$  can also be defined in block structure, we use  $\{\mathbf{H}_k\}$  and  $\{\mathbf{H}'_k\}$  to denote the main diagonal and upper diagonal blocks of  $\mathbf{H}$ , respectively. Then we have

$$\mathbf{H}'_k(i, j) = -\mathbf{P}'_k(i, j), \quad (39)$$

$$\mathbf{H}_k(i, j) = \begin{cases} 1 - \mathbf{P}_k(i, j) & \text{if } i = j, \\ -\mathbf{P}_k(i, j) & \text{otherwise.} \end{cases} \quad (40)$$

Based on the definition of  $\mathbf{P}_k$ , we know that  $0 < \mathbf{P}_k(i, i) < 1$ ,  $\mathbf{P}_k(i, i+1) > 0$ , so  $0 < \mathbf{H}_k(i, i) < 1$ ,  $\mathbf{H}_k(i, i+1) < 0$ , and all other entries of  $\mathbf{H}_k$  are zero. It is easy to see that  $|\mathbf{H}_k| \neq 0$ , so  $\mathbf{H}_k$  is an invertible matrix.

To derive  $\mathbf{N} = \mathbf{H}^{-1}$  based on elementary row operations, we first construct a combined matrix  $[\mathbf{H} \mid \mathbf{I}]$

consisting of matrix  $\mathbf{H}$  and the identity matrix  $\mathbf{I}$  of the same size. By applying elementary row operations to the combined matrix, we get  $[\mathbf{I} | \mathbf{N}]$ , so we have

$$\mathbf{N} = \begin{bmatrix} \mathbf{H}_0^{-1} & \cdots & \cdots & \cdots & \mathbf{N}_{1j} & \cdots \\ & \ddots & & & & \\ & & \mathbf{H}_i^{-1} & \cdots & \mathbf{N}_{ij} & \cdots \\ & & & \ddots & & \\ & & & & & \cdots \\ & & & & & \mathbf{H}_{x-1}^{-1} \end{bmatrix}. \quad (41)$$

Notice that  $\mathbf{N}$  is an upper triangle matrix, the  $(i, j)$ -entry  $\mathbf{N}_{ij}$  of  $\mathbf{N}$  is then determined as

$$\mathbf{N}_{ij} = (-1)^{j-i} \left( \prod_{k=i-1}^{j-2} \mathbf{H}_k^{-1} \mathbf{H}'_k \right) \mathbf{H}_{j-1}^{-1}, \quad (42)$$

where  $i \in [1, x]$ ,  $j \in (i, x]$ .

The (41) and (42) indicate that inverse matrix  $\mathbf{H}_k^{-1}$  needs to be derived. Based on elementary row operations, we have

$$\mathbf{H}_k^{-1} = \begin{bmatrix} \frac{1}{\mathbf{H}_k(1,1)} & \cdots & \cdots & \cdots & \mathbf{H}_k^{-1}(1,j) & \cdots \\ & \ddots & & & & \\ & & \frac{1}{\mathbf{H}_k(i,i)} & \cdots & \mathbf{H}_k^{-1}(i,j) & \cdots \\ & & & \ddots & & \\ & & & & & \cdots \\ & & & & & \frac{1}{\mathbf{H}_k(c_k, c_k)} \end{bmatrix}. \quad (43)$$

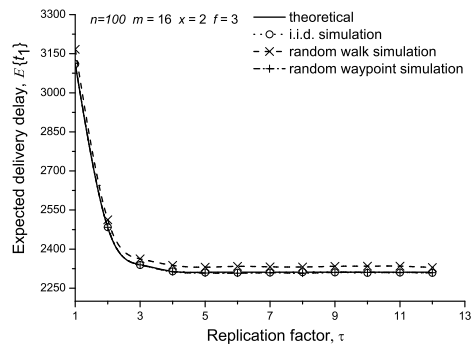
We can see that matrix  $\mathbf{H}_k^{-1}$  is also an upper triangle matrix, and its  $(i, j)$ -entry  $\mathbf{H}_k^{-1}(i, j)$  can be evaluated as

$$\mathbf{H}_k^{-1}(i, j) = (-1)^{j-i} \left( \prod_{z=i}^{j-1} \frac{\mathbf{H}_k(z, z+1)}{\mathbf{H}_k(z, z)} \right) \frac{1}{\mathbf{H}_k(j, j)} \quad (44)$$

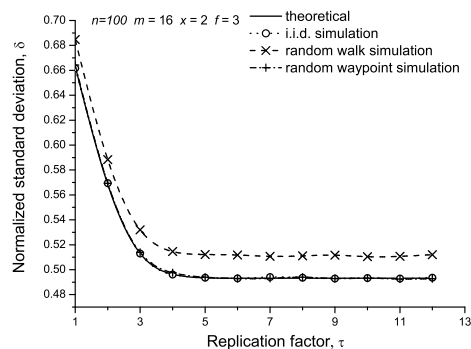
where  $k \in [0, x-1]$ ,  $i \in [1, c_k]$ , and  $j \in (i, c_k]$ .

## 7 Numerical Results

In this section, we first validate our theoretical models on expected packet delivery delay and delay variance, and then apply these models to illustrate the capability of the 2HR- $(x, \tau, f)$  algorithm in delay control.



(a)  $E\{t_1\}$  vs.  $\tau$



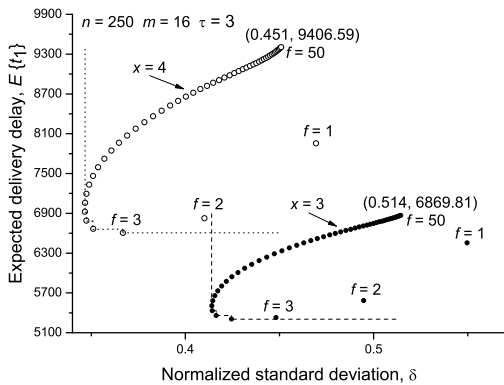
(b)  $\delta$  vs.  $\tau$

**Fig. 6** Theoretical and simulation results for model validation.

### 7.1 Model Validation

A network simulator <sup>2</sup> in C++ was developed to simulate the packet delivery process under the 2HR- $(x, \tau, f)$  routing and i.i.d. mobility model, where transmission group with guard factor  $\Delta = 1$  is adopted for transmission scheduling. For comparison, another two realistic mobility models, random walk model [25] and random waypoint model [26], were also implemented in the simulator. Based on the simulator, extensive simulations have been conducted for a network with  $n = 100$ ,  $m = 16$ ,  $x = 2$  and  $f = 3$ . Under different setting of replication factor  $\tau$ , the corresponding theoretical and simulation results on expected value  $E\{t_1\}$  and normalized standard deviation  $\delta = \sqrt{\text{Var}\{t_1\}}/E\{t_1\}$  of packet delivery delay are summarized in Fig.6. Here the

<sup>2</sup> In this paper, these network functions, like i.i.d. node mobility, transmission-group based scheduling scheme and packet delivery process of our algorithm, can be easily implemented by a customized C++ simulator (now publicly available at [24]) without going through a complicated network simulator (like NS2 and OPNET).



**Fig. 7** Delay region  $(\delta, E\{t_1\})$ .

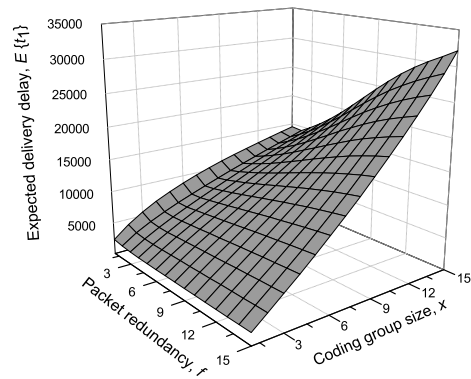
simulation results are reported with 95% confidence intervals.

We can see from Fig.6 that our theoretical models on expected packet delivery delay and delay variance are very efficient in capturing the delay behavior under the i.i.d. mobility and 2HR- $(x, \tau, f)$  routing. It is interesting to notice in Fig.6 that with 2HR- $(x, \tau, f)$  routing, the delay behaviors under the i.i.d. mobility and random waypoint are very similar each other, while the delay under the random walk exhibits a different behavior. Thus, our theoretical models, although was developed under the i.i.d. mobility model, can be used to predict the delay behavior under the random waypoint mobility model as well. The results in Fig.6 also imply that in general both expected delay  $E\{t_1\}$  and standard deviation  $\delta$  monotonically decrease as replication factor  $\tau$  increases.

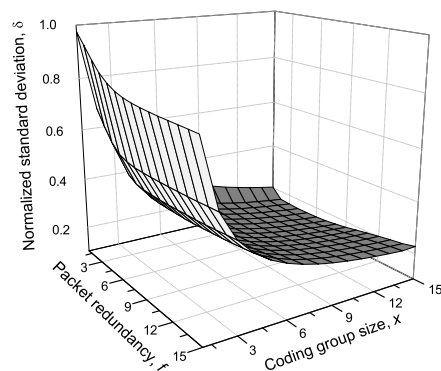
## 7.2 Delay Control

We illustrate how delay could be controlled by a proper setting of parameters in the 2HR- $(x, \tau, f)$  algorithm, so as to adapt to various applications with different delay/variance requirements.

We first explore how the delay performance can be controlled in a delay region in terms of  $(\delta, E\{t_1\})$  under the 2HR- $(x, \tau, f)$  algorithm. For network scenarios of  $n = 250, m = 16, \tau = 3$  and  $x = \{3, 4\}$ , Fig. 7 shows that the delay performance can be controlled in a delay region consisting of multiple discrete points  $(\delta, E\{t_1\})$  as  $f$  increases from 1 to 50. Another observation of Fig. 7 is that the delay region there can actually be determined by some horizontal and vertical lines defined by several key points (i.e., the Pareto optimal points [41]). For example, when  $x = 3$ , the delay region is co-determined by three points, i.e., the point



(a)  $E\{t_1\}$  vs.  $(f, x)$

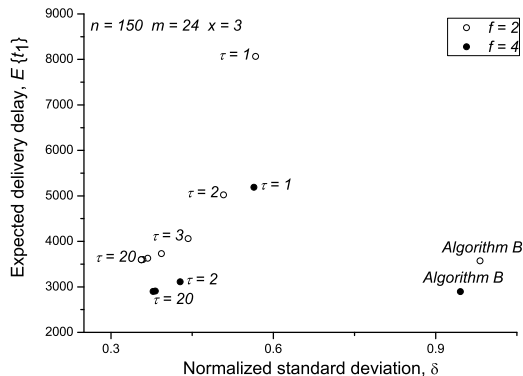


(b)  $\delta$  vs.  $(f, x)$

**Fig. 8** Delay performance for a network with  $n = 200, m = 16$  and  $\tau = 3$ .

$(0.424, 5307.38)$  ( $f = 4$ ), point  $(0.416, 5360.03)$  ( $f = 5$ ) and point  $(0.414, 5431.69)$  ( $f = 6$ ), where the point at  $f = 4$  results in the minimum  $E\{t_1\}$  of 5307.38 and the point at  $f = 6$  results in the minimum  $\delta$  of 0.414. When  $x = 4$ , the delay region is co-determined by four points, i.e., point  $(0.366, 6604.71)$  ( $f = 3$ ), point  $(0.351, 6666.38)$  ( $f = 4$ ), point  $(0.347, 6790.05)$  ( $f = 5$ ), and point  $(0.346, 6923.56)$  ( $f = 6$ ). Thus, for a specified coding group size  $x$ , the delay performance can be controlled by the 2HR- $(x, \tau, f)$  algorithm to meet any delay requirement in terms of  $(\delta, E\{t_1\})$  as long as it falls within the corresponding delay region.

We next explore how the delay performance can be controlled according to a specified delay target. For the network scenario of  $n = 200, m = 16$  and  $\tau = 3$ , Fig. 8 illustrates how the delay performance  $(\delta, E\{t_1\})$  varies with  $x$  and  $f$  there. One can observe from Fig. 8(a) that for a specified target  $t^*$  of expected delay value, we can define a target plane that intersects the  $z$ -axis orthogonally at the point  $(1, 1, t^*)$ , and thus can get a set of  $(x, f)$ -pairs from the intersection of the surface in Fig. 8(a) and the defined target plane. Similarly, for a



**Fig. 9** Delay performance comparison.

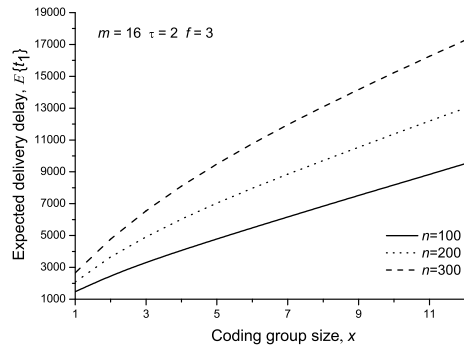
target  $\delta^*$  of the normalized standard deviation, we can also get a set of  $(x, f)$ -pairs in Fig. 8(b). By calculating the intersection of these two sets of  $(x, f)$ -pairs, we can then determine the set of  $(x, f)$ -pairs to meet the delay target in terms of  $(\delta^*, t^*)$ .

### 7.3 Performance Comparison

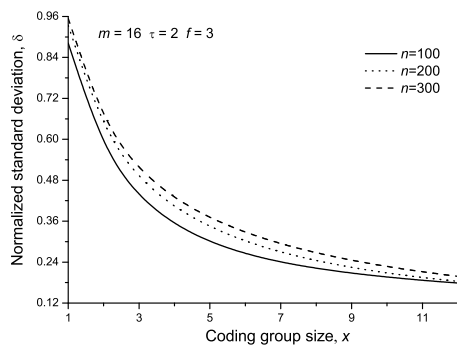
In this subsection, we compare the delay performance of state-of-the-art algorithms and that of the 2HR- $(x, \tau, f)$  algorithm proposed in this paper to show the efficiency of our algorithm. Specifically, we choose two well known algorithms, namely two-hop relay [30] with pure erasure coding technique (algorithm A for short) from subsection 2.1 and two-hop relay [14] with pure packet redundancy technique (algorithm B for short) from subsection 2.2. The corresponding results are summarized in Fig. 7 and Fig. 9.

Fig. 7 illustrates the performance comparison between our algorithm and algorithm A (i.e., the special case of our algorithm by setting  $f = 1$ ). As shown in Fig. 7 that in comparison with algorithm A, our algorithm can take the advantage of redundant copies of coded packets to improve the delay performance ( $\delta$ ,  $E\{t_1\}$ ) for the network setting considered there. For example, with the setting of  $x = 4$  and  $f = 1$ ,  $E\{t_1\}$  (resp.  $\delta$ ) under algorithm A is 7955.58 (resp. 0.469), but such performance is improved to 6604.71 (resp. 0.366) by adopting our algorithm with the setting of  $f = 3$ .

Fig. 9 illustrates the performance comparison between our algorithm and algorithm B. We can see from Fig. 9 that although algorithm B achieves very similar expected delay performance as our algorithm, it usually results in a significantly larger normalized standard deviation. For example, for the case of  $f = 2$  (resp.  $f = 4$ ) and  $\tau \geq 6$  (resp.  $\tau \geq 4$ ),  $E\{t_1\}$  under our algorithm is



(a)  $E\{t_1\}$  vs. coding group size  $x$



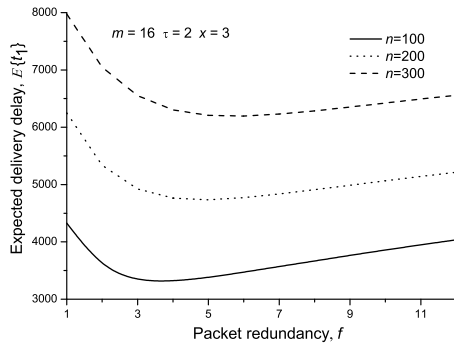
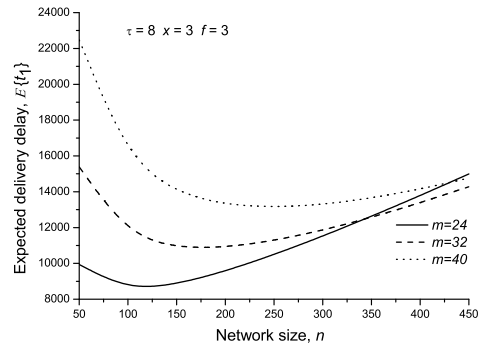
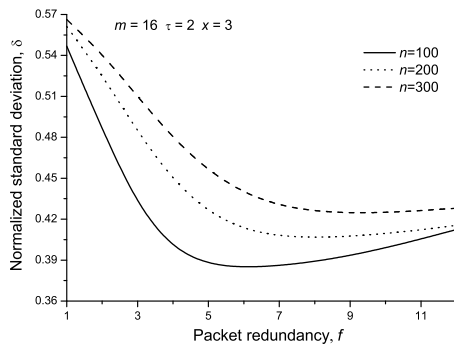
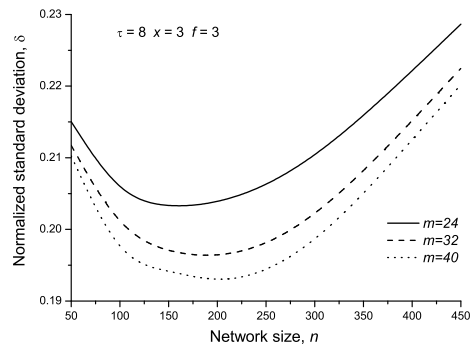
(b)  $\delta$  vs. coding group size  $x$

**Fig. 10** Delay performance vs. coding group size  $x$ .

3602.27 (resp. 2904.33), which is similar to the  $E\{t_1\}$  of 3572.17 (resp. 2897.34) under algorithm B, but the corresponding  $\delta$  of our algorithm is 0.358 (resp. 0.377), which is notably less than the  $\delta$  of 0.981 (resp. 0.945) under algorithm B.

### 7.4 Performance Analysis

We now explore how the packet delivery delay performance ( $\delta$ ,  $E\{t_1\}$ ) of the 2HR- $(x, \tau, f)$  algorithm varies with various parameters. With  $n = \{100, 200, 300\}$ ,  $m = 16$ ,  $\tau = 2$  and  $f = 3$ , we examine in Fig. 10 how  $E\{t_1\}$  and  $\delta$  vary with coding group size  $x$ . One can observe from Fig. 10 that as  $x$  increases,  $E\{t_1\}$  monotonically increases while corresponding  $\delta$  monotonically decreases. For example, for the setting of  $n = 100$ , the  $E\{t_1\}$  (resp.  $\delta$ ) at  $x = 3$  is 3317.71 (resp. 0.429), which is almost 0.61 (resp. 1.62) times that of  $x = 6$ . The results in Fig. 10 indicate through a proper control of coding group size  $x$ , a trade-off between  $E\{t_1\}$  and  $\delta$  can be initialized according different


 (a)  $E\{t_1\}$  vs. packet redundancy  $f$ 

 (a)  $E\{t_1\}$  vs. network size  $n$ 

 (b)  $\delta$  vs. packet redundancy  $f$ 

 (b)  $\delta$  vs. network size  $n$ 
**Fig. 11** Delay performance vs. packet redundancy  $f$ .

**Fig. 12** Delay performance vs. network size  $n$ .

delay (and variance) requirements of various applications.

For the scenarios of  $n = \{100, 200, 300\}$ ,  $m = 16$ ,  $\tau = 2$  and  $x = 3$ , Fig.11 illustrates how  $E\{t_1\}$  and  $\delta$  vary with packet redundancy  $f$ . It is easy to see from Fig.11 that for given scenario, as  $f$  increases, the  $E\{t_1\}$  (resp. $\delta$ ) first decreases and then increases, and there exists an optimum setting of  $f$  to achieve the minimum  $E\{t_1\}$  (resp. $\delta$ ). For example, for the case  $n = 100$  in Fig.11, a minimal  $E\{t_1\}$  (resp.  $\delta$ ) of 3310.21 (resp. 0.384) is achieved at  $f = 4$  (resp.  $f = 6$ ). An increase in packet redundancy  $f$  has two-fold effects on delay performance: on one hand, it increases the speed at which the destination receives a coded packet and thus decreases packet delay; on the other hand, it decreases the speed at which the source distributes copies of a coded packet and thus increases packet delay. When the first effect dominates the second one,  $E\{t_1\}$  decreases as  $f$  increases; when the second effect dominates the first one,  $E\{t_1\}$  increases as  $f$  further increases.

Finally, for the given setting of  $m = \{24, 32, 40\}$ ,  $\tau = 8$ ,  $x = 3$  and  $f = 3$ , we show in Fig.12 how  $E\{t_1\}$  and  $\delta$  vary with network size  $n$ . One can see

from Fig.12 that for a given setting of  $m$ , we can find a most suitable network size  $n^*$  (and thus most suitable average node density  $n/m^2$ ) to achieve the minimum  $E\{t_1\}$  (resp.  $\delta$ ). For example, for the setting of  $m = 24, 32$  and  $40$ , the most suitable network size is 100, 150 and 250 (resp. 150, 200 and 200) for a minimum  $E\{t_1\}$  (resp.  $\delta$ ). Actually, an increase in network size  $n$  has two-fold effects on delay performance: on one hand, it increases the speed at which a coded packet is distributed and thus decreases packet delay; on the other hand, it decreases the speed at which the destination receives a coded packet due to the negative effects of interference and medium contention issues and thus increases packet delay. When the network is sparse, the first effect dominates the second one, and thus  $E\{t_1\}$  decreases as  $n$  increases; when the network users become relatively densely distributed, the second effect dominates the first one, and thus  $E\{t_1\}$  increases as  $n$  further increases.

## 8 Conclusion

By integrating erasure coding technique with packet redundancy technique, this paper proposed a general 2HR- $(x, \tau, f)$  algorithm for flexible delay control in MANETs. A theoretical framework was further developed to reveal the delay performance under the 2HR- $(x, \tau, f)$  algorithm. Extensive numerical results provided in this paper indicate that the 2HR- $(x, \tau, f)$  algorithm has the capability of controlling packet delivery delay in a large region, and it also enables a flexible trade-off between expected delivery delay and delay variance to be initiated through a proper setting of coding group size  $x$ , replication factor  $\tau$  and packet redundancy  $f$ . It is expected that the 2HR- $(x, \tau, f)$  algorithm can facilitate future MANETs to support various application with different requirements on delay and delay variance.

**Acknowledgments** This work is supported in part by the Significant Natural Science Foundation of the Education Department of Anhui Province under Grant No.KJ2011ZD06, the Key Program of Natural Science Foundation of Chuzhou University under Grant No.2012kj002Z and the Chuzhou University Excellent Young Talents Fund Project under Grant No.2013RC005.

## References

- Andrews J., Shakkottai S., Heath R., Jindal N., Haenggi M., Berry R., Guo D., Neely M., Weber S., Jafar S., & Yener A. (2008). Rethinking information theory for mobile ad hoc networks. *IEEE Communications Magazine*, 46(12), 94–101.
- Wang Z., Chen Y., & Li C. (2012). Corman: A novel cooperative opportunistic routing scheme in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 30(2), 289–296.
- Goldsmith A., Effros M., Koetter R., Medard M., Ozdaglar A., & Zheng L. (2011). Beyond shannon: The quest for fundamental performance limits of wireless ad hoc networks. *IEEE Communications Magazine*, 49(5), 195–205.
- Kannhavong B., Nakayama H., Nemoto Y., Kato N., & Jamalipour A. (2007). A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications Magazine*, 14(5), 85–91.
- Kannhavong B., Nakayama H., Kato N., Jamalipour A., & Nemoto Y. (2007). A study of a routing attack in olsr-based mobile ad hoc networks. *International Journal of Communication Systems*, 20(11), 1245–1261.
- Zhou Y., Zhang Y., Xie Y., Zhang H., Yang L. T., & Min G. (2014). TransCom: A virtual disk-based cloud computing platform for heterogeneous services. *IEEE Transactions on Network and Service Management*, 11(1), 46–59.
- Wang Y., Jain S., Martonosi M., & Fall K. (2005). Erasure-coding based routing for opportunistic networks. In *Proceedings of ACM SIGCOMM WDTN*.
- Hanbali A. A., Kherani A. A., & Nain P. (2007). Simple models for the performance evaluation of a class of two-hop relay protocols. In *Proceedings of IFIP Networking*.
- Liu J., Jiang X., Nishiyama H., & Kato N. (2011). Delay and capacity in ad hoc mobile networks with f-cast relay algorithms. *IEEE Transactions on Wireless Communications*, 10(8), 2738–2751.
- Grossglauser M., & Tse D. N. (2001). Mobility increases the capacity of ad hoc wireless networks. In *Proceedings of INFOCOM*.
- Li P., Fang Y., Li J., & Huang X. (2012). Smooth trade-offs between throughput and delay in mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 11(3), 427–438.
- Urgaonkar R., & Neely M. J. (2011). Network capacity region and minimum energy function for a delay-tolerant mobile ad hoc network. *IEEE/ACM Transactions on Networking*, 19(4), 1137–1150.
- Ying L., Yang S., & Srikant R. (2008). Optimal delay-throughput trade-offs in mobile ad hoc networks. *IEEE Transactions on Information Theory*, 54(9), 4119–4143.
- Neely M. J., & Modiano E. (2005). Capacity and delay tradeoffs for ad-hoc mobile networks. *IEEE Transactions on Information Theory*, 51(6), 1917–1936.
- Liu J., Jiang X., Nishiyama H., & Kato N. (2012). Exact throughput capacity under power control in mobile ad hoc networks. In *Proceedings of INFOCOM*.
- Zhang C., Fang Y., & Zhu X. (2009). Throughput-delay tradeoffs in large scale manets with network coding. In *Proceedings of INFOCOM*.
- Liu J., Jiang X., Nishiyama H., & Kato N. (2011). Performance modeling for two-hop relay with erasure coding in manets. In *Proceedings of Globecom*.
- Ciullo D., Martina V., Garetto M., & Leonardi E. (2010). Impact of correlated mobility on delay-throughput performance in mobile ad-hoc networks. In *Proceedings of INFOCOM*.
- Garetto M., Giaccone P., & Leonardi E. (2009). Capacity scaling in ad hoc networks with heterogeneous mobile nodes: the subcritical regime. *IEEE/ACM Transactions on Networking*, 17(6), 1888–1901.
- Gupta P., & Kumar P. (2000). The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2), 388–404.
- Li P., Fang Y., & Li J. (2010). Throughput, delay, and mobility in wireless ad hoc networks. In *Proceedings of INFOCOM*.
- Rizzo L. (1997). Effective erasure codes for reliable computer communication protocols. *Computer Communication Review*, 27(2), 24–36.
- Grinstead C. M., & Snell J. L. (1997). Introduction to Probability, 2nd ed. *American Mathematical Society*.
- C++ simulator for the 2hr-(x;  $\tau$ ; f) manets. [Online]. Available: <http://wlcsearch.blogspot.jp>.
- Gamal A. E., Mammen J., Prabhakar B., & Shah D. (2006). Optimal throughput-delay scaling in wireless networks-part i: The fluid model. *IEEE Transactions on Information Theory*, 52(6), 2568–2592.
- Zhou S., & Ying L. (2010). On delay constrained multicast capacity of largescale mobile ad-hoc networks. In *Proceedings of INFOCOM*.
- Liao Y., Tan K., Zhang Z., & Gao L. (2006). Estimation based erasure coding routing in delay tolerant networks. In *Proceedings of IWCMC*.
- Tsapeli F., & Tsaoussidis V. (2012). Routing for opportunistic networks based on probabilistic erasure coding. In *Proceedings of WWIC*.
- Chen L., Yu C., Sun T., Chen Y., & Chu H. (2006). hybrid routing approach for opportunistic networks. In *Proceedings of ACM SIGCOMM Workshop on Challenged Networks*.

30. Kong Z., Yeh E., & Soljanin E. (2012). Coding improves the throughput-delay tradeoff in mobile wireless networks. *IEEE Transactions on Information Theory*, 58(11), 6894–6906.
31. Altman E., & Pellegrini F. D. (2011). Forward correction and fountain codes in delay-tolerant networks. *IEEE/ACM Transactions on Networking*, 19(1), 1-13.
32. Altman E., Sassatelli L., & Pellegrini F. D. (2013). Dynamic control of coding for progressive packet arrivals in dtns. *IEEE Transactions on Wireless Communications*, 12(2), 725-735.
33. Sharma G., & Mazumdar R. (2004). On achievable delay/capacity trade-offs in mobile ad hoc networks. In *Proceedings of WiOpt*.
34. Sharma G., Mazumdar R., & Shroff N. (2007). Delay and capacity tradeoffs in mobile ad hoc networks: A global perspectives. *IEEE/ACM Transactions on Networking*, 15(5), 981-992.
35. Groenevelt R., Nain P., & Koole G. (2005). The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1-4), 210-228.
36. Panagakos A., Vaioas A., & Stavarakakis I. (2007). Study of two-hop message spreading in dtns. In *Proceedings of WiOpt*.
37. Hanbali A. A., Nain P., & Altman E. (2008). Performance of ad hoc networks with two-hop relay routing and limited packet lifetime-extended version. *Performance Evaluation*, 65(6-7), 463-483.
38. Liu J., Jiang X., Nishiyama H., & Kato N. (2012). Generalized two-hop relay for flexible delay control in manets. *IEEE/ACM Transactions on Networking*, 20(6), 1950-1963.
39. Spyropoulos T., Psounis K., & Raghavendra C. S. (2005). Spray and wait: An efficient routing scheme for intermittently connected mobile networks. In *Proceedings of ACM SIGCOMM Workshop*.
40. Bulut E., Wang Z., & Szymanski B. K. (2010). Cost effective multi-period spraying for routing in delay tolerant networks. *IEEE/ACM Transactions on Networking*, 18(5), 1530-1543.
41. Boche H., Naik S., & Schubert M. (2011). Pareto boundary of utility sets for multiuser wireless systems. *IEEE/ACM Transactions on Networking*, 19(2), 589-601.
42. Garetto M., & Leonardi E. (2010). Restricted mobility improves delay-throughput trade-offs in mobile ad-hoc networks. *IEEE Transactions on Information Theory*, 56(10), 5016-5029.