Throughput Capacity of the Group-Based Two-Hop Relay Algorithm in MANETs

Throughput Capacity of the Group-Based Two-Hop Relay Algorithm in MANETs

Jiajia Liu
Tohoku University
Sendai, Japan 980-8579
Email: liu-jia@it.ecei.tohoku.ac.jp

Xiaohong Jiang
Future University Hakodate
Hokkaido, Japan 041-8655
Email: jiang@fun.ac.jp

Hiroki Nishiyama and Nei Kato
Tohoku University
Sendai, Japan 980-8579
Email: {bigtree,kato}@it.ecei.tohoku.ac.jp

***Abstract*—This paper focuses on the per node throughput capacity in mobile ad hoc networks (MANETs) with the general group-based two-hop relay algorithm. Under such an algorithm with packet redundancy limit f and group size g (2HR-(f, g) for short), each packet is delivered to at most f distinct relay nodes and can be accepted by its destination if it is a fresh packet to the destination and also it is among g packets of the group the destination is currently requesting. A general Markov chain-based theoretical framework is first developed to characterize the complicated packet delivery process in the challenging MANET environment. With the help of the new theoretical framework, closed-form expressions are further derived for the throughput capacity of the 2HR-(f, g) algorithm, from which one can easily recover the available throughput capacity results by proper settings of the redundancy limit f and group size g.**

I. INTRODUCTION

The mobile ad hoc network (MANET), a flexible peer-to-peer network technique, holds great promise for a lot of future applications, such as the disaster relief, emergency response, military communication and pedestrian network, etc. Since it enables mobile nodes to randomly move around and freely communicate to each other without any pre-existing infrastructure support or centralized management, the MANET is believed to be one of the most important and indispensable component among the next generation networks [1]–[4]. Due to the complicated issues of node mobility and wireless interference, however, the fundamental performance limits, like the throughput capacity and packet delay, remain unknown for the challenging MANET environment [5].

So far, a significant amount of works has been done to characterize the order sense scaling laws of throughput capacity in MANETs. Grossglauser and Tse in [6] showed that by adopting the basic two-hop relay algorithm, it is possible to achieve a $\Theta(1)$ per node throughput under the i.i.d. model. Later, Li *et al.* in [7] proved that with a variant of the basic two-hop relay the per node throughput capacity is upper bounded by $O(n^{\beta-\alpha-1/2})$, where the network area is evenly divided into $n^{2\alpha}$ cells and each cell is further evenly divided into squares of area $n^{-2\beta}$.

More recently, some closed-form expressions have also been reported in literature for the throughput capacity of MANETs. Neely and Modiano in [8] showed that if the user density is fixed, the per node throughput capacity in a cell partitioned MANET tends to a fixed value as the number of users scales up. Later, the work was further extended in [9] to incorporate the minimum time-average power required to support the per node throughput capacity there. The exact per node throughput capacity has also been examined in [10], [11] where a general two-hop relay with limited packet redundancy and transmission power control was considered.

It is notice that the above works either considered two-hop relay algorithms with in-order reception [8], [10], [11] or adopted the out-of-order ones with/without packet redundancy [6], [9], which actually can be covered as special cases by the general group-based two-hop relay algorithm with redundancy [12], [13]. Under such an algorithm with packet redundancy limit f and group size g (2HR-(f, g) for short), each packet is delivered to at most f distinct relay nodes and can be accepted by its destination if it is a fresh packet to the destination and also it is among g packets of the group the destination is currently requesting. In this paper, we focus on the per node throughput capacity performance in the 2HR-(f, g) MANETs. We first develop a general finite-state absorbing Markov chain theoretical framework to model the complicated packet delivery process in the challenging MANET environment, and then adopt the blocking matrix technique to derive closed-form expressions for the per node throughput capacity there. With our theoretical framework and throughput capacity results, one can easily recover the available throughput capacity results in [8], [10], [11] by proper settings of the control parameters (f, g).

The remainder of this paper is outlined as follow. Section II introduces the system models, transmission scheduling scheme and the 2HR-(f, g) algorithm. We develop the Markov chain theoretical framework in Section III, derive closed-form results for the per node throughput capacity in Section IV, and finally conclude the whole paper in Section V.

II. PRELIMINARIES

A. System Models

The considered network is a two-dimensional torus with n mobile nodes and unit area, which is evenly divided into $m \times m$ equal cells, each cell of side length $1/m$ as shown in Fig. 1. Time is slotted and the i.i.d. model [8] is adopted here as the mobility model, where each node first independently and randomly selects a destination cell over all m^2 cells at the beginning of each time slot and then stays inside for a whole

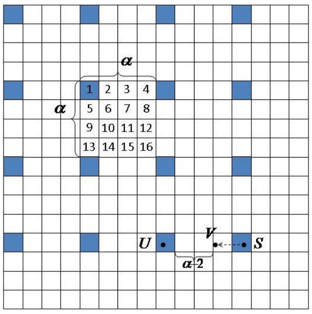


Fig. 1. Illustration of the cell-partitioned network and cells in a concurrent-set with $m = 16$ and $\alpha = 4$. The distribution information of all other mobile nodes in the network is not shown for simplicity.

time slot. We adopt the protocol model introduced in [14] as the interference model and denote by Δ the guard factor.

Similar to [10], [13], we assume that only one-hop transmissions are available within each time slot, and each node with transmission range r can only send data to nodes in the same cell or its eight adjacent cells (two cells are called adjacent if they share a common point). Thus, the transmission range can be accordingly determined as $r = \sqrt{8}/m$. To simplify the analysis, we assume that the total number of bits transmitted per slot is fixed and normalized to 1 packet. We further assume a permutation traffic pattern, in which each node has a locally generated traffic flow to deliver to its destination, and also needs to receive another traffic flow originated from some other node. Thus, there are in total n distinct traffic flows in the network. Without loss of generality, we focus on a tagged flow hereafter and denote by S and D the source node and the destination node, respectively.

B. Transmission Scheduling

In order to schedule as many simultaneous link transmissions as possible, we define here the "concurrent-set" for transmission scheduling [13].

Concurrent-set: A concurrent-set is a subset of cells in which any two of them have a vertical and horizontal distance of some multiple of α cells and all cells there can transmit simultaneously without interfering with each other.

It is easy to see that with such a concurrent-set definition, all m^2 cells are actually divided into α^2 distinct concurrent-sets. As illustrated in Fig. 1 for the case $\alpha = 4$, there are in total 16 concurrent-sets and all shaded cells belong to the same concurrent-set. In order to guarantee simultaneous link transmissions in a concurrent-set, the parameter α should be carefully tuned. As shown in Fig. 1, suppose in some time slot node S is scheduled to transmit a packet to node V. One can see that another transmitting node U in some other active cell is at least $(\alpha-2)/m$ away from V. According to the protocol interference model [14], the condition that U will not interfere with the packet reception at V is that $(\alpha-2)/m \geq (1+\Delta)r$. Substituting $r = \sqrt{8}/m$, then we have

$$\alpha = \min\{\lceil (1+\Delta)\sqrt{8} + 2 \rceil, m\} \tag{1}$$

If we let each concurrent-set become active (i.e., have link transmissions) alternatively, then each cell will also become active every α^2 time slots. If there are more than one nodes in an active cell, a transmitting node is randomly selected from them, and the selected node then follows the following 2HR-(f,g) algorithm for packet transmission.

C. 2HR-(f,g) Algorithm

The 2HR-(f,g) algorithm was first proposed in [12], and we include it here for completeness. When operating under such algorithm, each node maintains n individual queues at its buffer: one local-queue for storing locally generated packets, one already-sent-queue for storing packets whose f copies have already been distributed but their reception status are not confirmed yet (from the destination node), and $n-2$ relay-queues for storing packets of other $n-2$ traffic flows (except the two ones originated from and destined for itself).

A sequence number-based mechanism is adopted in the 2HR-(f,g) algorithm to avoid network congestion. Specifically, for the tagged flow, node S divides packets waiting at its local-queue into consecutive groups, g packets per group, and labels each packet P with a *send group number* $SG(P)$ and *sequence number* $SN(P)$ $(1 \leq SN(P) \leq g)$. The destination node D also maintains a *request group number* $RG(D)$ and *indicator vector* $IN(D)$, where $RG(D)$ denotes the packet group number that D is currently requesting for and $IN(D)$ records the reception status of all packets in the current requesting group, i.e., group $RG(D)$. To simplify the analysis, each relay is assumed to carry at most one packet for any particular group. Then we are ready to introduce the 2HR-(f,g) algorithm.

2HR-(f,g) Algorithm: Every time node S wins a transmission opportunity, it operates as follows.

Step 1: ("source-to-destination") If node D is inside the one-hop range, after obtaining $RG(D)$ and $IN(D)$ from D, S transmits directly to D a fresh packet of group $RG(D)$. A packet is called a *fresh packet* if it has not been received yet by its destination; otherwise, it is called a non-fresh packet.

Step 2: Otherwise, S randomly chooses to perform one of the following operations with equal probability:

- ("source-to-relay") It randomly selects one node, say R, from the one-hop neighbors, and checks whether R is a fresh node. If not, it delivers to R a copy of the head-of-line packet P_h at its local-queue; otherwise it remains idle for this time slot. For a tagged packet group, a node is called a *fresh node* if it is carrying a fresh packet for the group; otherwise, it is called a non-fresh node. After distributing out f copies for P_h, S puts P_h to the end of the already-sent-queue and moves ahead the remaining packets at the local-queue.
- ("relay-to-destination") The node S acts as a relay and randomly selects a node (say V) from the one-hop neighbors. After obtaining the $RG(V)$ and $IN(V)$ from V, S checks whether there exists a fresh packet of group $RG(V)$ in its relay-queue specified for V. If so, it transmits this packet to V and removes all packets with

$SG \leq RG(V)$ from its relay-queue for V; otherwise, it remains idle for this time slot.

III. Markov Chain Framework

A. Some Basic Probabilities

Lemma 1: For a time slot and the tagged flow, if we use p_1 to denote the probability that S conducts a "source-to-destination" transmission with node D and use p_2 to denote the probability that S conducts a "source-to-relay" transmission or "relay-to-destination" transmission with some other node, then p_1 and p_2 can be given by

$$p_1 = \frac{1}{\alpha^2}\left(\frac{9n-m^2}{n(n-1)} - \left(1-\frac{1}{m^2}\right)^{n-1}\frac{8n+1-m^2}{n(n-1)}\right) \quad (2)$$

$$p_2 = \frac{1}{\alpha^2}\left(\frac{m^2-9}{n-1}\left(1-\left(1-\frac{1}{m^2}\right)^{n-1}\right) - \left(1-\frac{9}{m^2}\right)^{n-1}\right) \quad (3)$$

Lemma 2: For a time slot and the tagged flow, suppose that node S is delivering copies for some packet group i, node D is requesting packets of the group i, and there are currently t_1 fresh nodes and t_2 non-fresh nodes for the group i in the network. If we denote by $p_r(t_1)$ the probability that D will receive a fresh packet, denote by $p_d(t_2)$ the probability that S will successfully deliver out a new copy to some non-fresh node, and denote by $p_s(t_1,t_2)$ the probability of both "source-to-relay" transmission and "relay-to-destination" transmission in the next time slot, then we have

$$p_r(t_1) = p_1 + \frac{t_1}{2(n-2)}p_2 \quad (4)$$

$$p_d(t_2) = \frac{t_2}{2(n-2)}p_2 \quad (5)$$

$$p_s(t_1,t_2) = \frac{t_1t_2(m^2-\alpha^2)}{4m^2\alpha^4}\sum_{k=0}^{n-5}\binom{n-5}{k}h(k) \cdot\left\{\sum_{t=0}^{n-4-k}\binom{n-4-k}{t}h(t)\left(1-\frac{18}{m^2}\right)^{n-4-k-t}\right\} \quad (6)$$

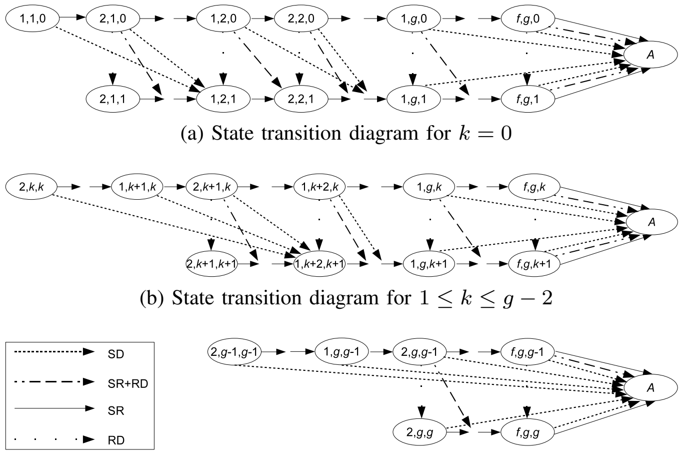
where

$$h(x) = \frac{9\left(\frac{9}{m^2}\right)^{x+1} - 8\left(\frac{8}{m^2}\right)^{x+1}}{(x+1)(x+2)} \quad (7)$$

The derivations of (2), (3), (4), (5) and (6) are omitted here due to space limit, and please refer to [13] for details.

B. Markov Chain-Based Theoretical Framework

Before proceeding to develop the Markov chain theoretical framework, we first introduce two service processes, i.e., the *packet dispatching process* at the source node S and the *packet receiving process* at the destination node D. According to the 2HR-(f,g) algorithm, S delivers out at most f distinct copies for each packet; D accepts packets according to their group numbers, i.e., D will accept a packet as long as it is fresh and also among the packet group D is currently requesting for. Thus, we can see that under the 2HR-(f,g) algorithm, the inter-group packet reception at D is strictly in-group-order, while the intra-group packet reception at D is totally out-of-order.



(a) State transition diagram for $k=0$

(b) State transition diagram for $1 \leq k \leq g-2$

(c) State transition diagram for $g-1 \leq k \leq g$

Fig. 2. Transition diagram of the Markov chain for the fastest packet dispatching process at S. For each transient state there, the transition back to itself is not shown for simplicity.

Consider a general packet group of the tagged flow and the initial network state when D is just starting to request for this packet group (i.e., the network state when D receives the last packet of the former group). Due to the network dynamics of node mobility and traffic arrivals, such initial network state may vary significantly from group to group. Specifically, if we denote by (i,j,k) a general network state that S is delivering the i_{th} $(1 \leq i \leq f)$ copy for the j_{th} $(1 \leq j \leq g)$ packet of the group while D has received any k $(0 \leq k \leq j)$ of the j packets, and denote by $(*,*,k)$ the state that S has already finished copy distribution for the tagged group while D has only received k $(0 \leq k < g)$ packets of them, then the set of initial network states for the tagged group can be determined as $\{(i,j,0)\} \cup \{*,*,0\}$ [1], $i \in [1,f], j \in [1,g]$.

According to [10], [11], we can see that in order to derive the throughput capacity of the general 2HR-(f,g) algorithm, we need to first characterize the fastest packet dispatching process at S and the fastest packet receiving process at D. It is easy to see that the fastest packet dispatching process at S corresponds to the initial network state $(1,1,0)$ (i.e., when S starts to deliver the first copy for the first packet, D is also requesting for this group); while the fastest packet receiving process at D corresponds to the initial network state $(*,*,0)$ (i.e., when D starts to request for the group, S has already finished copy distribution for this group). If we further denote by A the absorbing state (i.e., the termination of the service process), the corresponding fastest packet dispatching process at S and the fastest packet receiving process at D can be defined by two finite-state absorbing Markov chains shown in Fig. 2 and Fig. 3, respectively.

[1]Notice that state $(1,1,0)$ also corresponds to the case that when D starts to request for the tagged packet group, say group SG, S is still delivering packets for the former group (i.e., group $SG-1$).

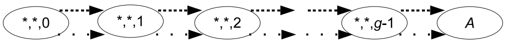


Fig. 3. Transition diagram of the Markov chain for the fastest packet receiving process at D. For each transient state there, the transition back to itself is not shown for simplicity.

For the Markov chain in Fig. 2, Figs. 2a, 2b and 2c each represents some cases of the full chain. Specifically, Fig. 2a represents the cases when S finishes copy distribution for the tagged group, no more than one packet is received at D; Fig. 2b illustrates the cases that D may receive at most one more packet by the time S distributes out all packet copies, given that D has already received k packets of them, $1 \leq k \leq g-2$; Fig. 2c defines the cases that $g-1$ or all g packets of the tagged group are received at D by the end of packet dispatching process at S. The transitions of SD, SR, RD and SR+RD in Fig. 2 correspond to the transmissions of "source-to-destination", "source-to-relay", "relay-to-destination" and both "source-to-relay" and "relay-to-destination" in the 2HR-(f, g) algorithm, respectively. Since the Markov chain in Fig. 3 represents the case that when D starts to request for the tagged packet group S has already finished copy distribution for the group, there are only SD and RD transitions among neighboring states there.

One can easily observe from Fig. 2 that in the Markov chain model there, the total number of transient states β is given by

$$\beta = \frac{g}{2}(g \cdot f + 3f - 2) \tag{8}$$

A further careful observation of Fig. 2 indicates that these β transient states are actually arranged into $g+1$ rows, where the number of transient states L_k in the k_{th} $(0 \leq k \leq g)$ row is determined as

$$L_k = \begin{cases} g \cdot f & \text{if } k = 0, \\ (g-k+1)f - 1 & \text{if } 1 \leq k \leq g. \end{cases} \tag{9}$$

For the Markov chain model in Fig. 2, if we use u_r and u_o to denote the number of fresh nodes and the number of non-fresh nodes in the t_{th} $(1 \leq t \leq L_k)$ state of the k_{th} $(0 \leq k \leq g)$ row, respectively, then after applying some derivations similar to that in [13], the u_r and u_o can be given by

$$u_r = \begin{cases} t - 1 & \text{if } k = 0, \\ t - f & \text{if } 1 \leq k \leq g. \end{cases} \tag{10}$$

$$u_o = \begin{cases} n - 1 - t & \text{if } k = 0, \\ n - 2 - t + k - (k-1)f & \text{if } 1 \leq k \leq g. \end{cases} \tag{11}$$

IV. Per Node Throughput Capacity Analysis

A. Throughput Capacity

A traffic input rate is called feasible under the 2HR-(f, g) algorithm if with such input rate the queue length at each node will not grow to infinity as the time goes to infinity. The per node throughput capacity is defined as the maximum feasible input rate at each node. Then we have the following theorem regarding the per node throughput capacity under the 2HR-(f, g) algorithm.

Theorem 1: For the considered MANET with i.i.d. mobility model, if we denote by μ (packets/slot) the per node (flow) throughput capacity under the general 2HR-(f, g) algorithm, then the throughput capacity μ can be determined as

$$\mu = \min\left\{\frac{g}{\mathbb{E}\{X_S\}}, \frac{g}{\mathbb{E}\{X_D\}}\right\} \tag{12}$$

where $\mathbb{E}\{X_S\}$ denotes the mean time it takes the Markov chain in Fig. 2 to become absorbed given that the chain starts from the state $(1, 1, 0)$, and $\mathbb{E}\{X_D\}$ denotes the mean time it takes the Markov chain in Fig. 3 to get absorbed given that the chain starts from the state $(*, *, 0)$.

Proof: Without loss of generality, we focus on the tagged flow. According to [10], [11], we can see that the per node throughput capacity μ under the 2HR-(f, g) algorithm is actually determined by the minimum one of two service rates for a general packet group, i.e., the fastest service rate of packet dispatching process at the source node S and the fastest service rate of packet receiving process at the destination node D. From the Markov chain framework in Section III, we can see that $\mathbb{E}\{X_S\}$ and $\mathbb{E}\{X_D\}$ actually correspond to the shortest service time of the dispatching process at S and that of the receiving process at D, respectively. Thus, the per node throughput capacity can be determined as (12), and then we finish the proof for Theorem 1. ■

It is easy to see that for the state $(*, *, k)$ $(k \in [0, g-1])$ of the Markov chain in Fig. 3, there are $(g-k)f$ fresh nodes and thus the probability of RD transition there is $p_r((g-k)f) - p_1$. According to the Markov chain theory [15], the expected service time $\mathbb{E}\{X_D\}$ can be given by

$$\mathbb{E}\{X_D\} = \sum_{k=0}^{g-1} \frac{1}{p_1 + \frac{(g-k)f}{2(n-2)} p_2} \tag{13}$$

Now we proceed to derive the expected service time $\mathbb{E}\{X_S\}$. As illustrated in Fig. 2, all β transient states there are arranged into $g+1$ rows. We number these transient states sequentially as $1, 2, \ldots, \beta$ in a left-to-right and top-to-down way. If we denote by q_{ij} the transition probability from transient state i to transient state j, $i, j \in [1, \beta]$, then we can get a matrix $\mathbf{Q} = (q_{ij})_{\beta \times \beta}$ defining the transition probabilities among all transient states there. According to the Markov chain theory [15], we can see that the fundamental matrix $\mathbf{N}$ of the Markov chain in Fig. 2 is determined as

$$\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1} \tag{14}$$

where $\mathbf{I}$ is the unit matrix and $\mathbf{N} = (a_{ij})_{\beta \times \beta}$. Since the ij-entry of matrix $\mathbf{N}$, i.e., a_{ij}, represents the expected number of times in state j given that the chain starts from state i, the expected service time $\mathbb{E}\{X_S\}$ can be given by

$$\mathbb{E}\{X_S\} = \mathbf{b} \cdot \mathbf{N} \cdot \mathbf{c} \tag{15}$$

where $\mathbf{b} = (1, 0, \ldots, 0)$ and $\mathbf{c} = (1, 1, \ldots, 1)^T$.

Combining the above results in (12), (13), (14) and (15), we can see that in order to derive the throughput capacity μ, the only remaining issue is to derive the matrices $\mathbf{Q}$ and $\mathbf{N}$, as discussed in the following section.

B. Derivations of Matrices $\mathbf{Q}$ and $\mathbf{N}$

Consider the Markov chain in Fig. 2. It is noticed that for the $g+1$ rows of transient states there, the transitions only happen among transient states of the same row or neighboring rows. Thus, the matrix $\mathbf{Q}$ can be defined in a blocking way as follows

$$\mathbf{Q} = \begin{bmatrix} \mathbf{Q}_0 & \mathbf{Q}_0^{'} & & & & & \\ & \mathbf{Q}_1 & \mathbf{Q}_1^{'} & & & & \\ & & \ddots & \ddots & & & \\ & & & \mathbf{Q}_k & \mathbf{Q}_k^{'} & & \\ & & & & \ddots & \ddots & \\ & & & & & \mathbf{Q}_{g-1} & \mathbf{Q}_{g-1}^{'} \\ & & & & & & \mathbf{Q}_g \end{bmatrix} \quad (16)$$

where sub-matrix $\mathbf{Q}_k$ corresponds to the transition probabilities among the k_{th} row of transient states, and sub-matrix $\mathbf{Q}_k^{'}$ corresponds to the transition probabilities from transient states of the k_{th} row to that of the $(k+1)_{th}$ row. All the other sub-matrices in (16) are zero-matrices and omitted for simplicity.

Derivations of $\mathbf{Q}_k$: the non-zero ij-entry $\mathbf{Q}_k(i,j)$ $(k \in [0,g], i,j \in [1,L_k])$ of sub-matrix $\mathbf{Q}_k$ is determined as

$$\mathbf{Q}_k(i,i+1) = p_d(u_o) - p_s(u_r,u_o) \quad \text{if } 1 \le i < L_k \quad (17)$$

$$\mathbf{Q}_k(i,i) = 1 - p_d(u_o) - p_r(u_r) + p_s(u_r,u_o) \quad (18)$$

Derivations of $\mathbf{Q}_k^{'}$: the non-zero ij-entry $\mathbf{Q}_k^{'}(i,j)$ $(k \in [0,g-1], i \in [1,L_k], j \in [1,L_{k+1}])$ of sub-matrix $\mathbf{Q}_k^{'}$ can be determined as

$$\mathbf{Q}_k^{'}(i,i-d(k)) = p_r(u_r) - p_s(u_r,u_o) - p_1 \text{ if } d(k)+1 \le i \le L_k \quad (19)$$

$$\mathbf{Q}_k^{'}(i,i-d(k)+1) = p_s(u_r,u_o) \quad \text{if } d(k)+1 \le i < L_k \quad (20)$$

where

$$d(k) = \begin{cases} 1 & \text{if } k=0, \\ f & \text{if } 1 \le k \le g-1. \end{cases} \quad (21)$$

$$\mathbf{Q}_k^{'}(i,e(i)\cdot f) = p_1 \quad \text{if } 0 \le k < g-1, 1 \le i \le L_k - f \quad (22)$$

where

$$e(i) = \begin{cases} \lceil \frac{i}{f} \rceil & \text{if } k=0, \\ 1 & \text{if } 1 \le k < g-1, 1 \le i \le f-1, \\ \lfloor \frac{i}{f} \rfloor & \text{if } 1 \le k < g-1, f \le i \le L_k - f. \end{cases} \quad (23)$$

From the above derivations for sub-matrices $\{\mathbf{Q}_k\}$ and $\{\mathbf{Q}_k^{'}\}$, the matrix $\mathbf{Q}$ in (16) can then be accordingly derived. Since the fundamental matrix $\mathbf{N} = (\mathbf{I} - \mathbf{Q})^{-1}$, we can derive $\mathbf{N}$ based on $\mathbf{Q}$. The derivation for matrix $\mathbf{N}$ is omitted here due to space limit and please refer to [13] for details.

V. Conclusion

In this paper, we have investigated the per node throughput capacity in MANETs with the general 2HR-(f,g) algorithm, which covers all the available two-hop relay algorithms as special cases. A finite-state absorbing Markov chain-based theoretical framework was developed to model the fastest packet dispatching process at the source node and also the fastest packet receiving process at the destination node. Based on the new theoretical framework, closed-form expressions were further derived for the throughput capacity of the 2HR-(f,g) algorithm by adopting the blocking matrix technique. It is expected that our throughput capacity results can help network designers to select appropriate control parameters f and g of the 2HR-(f,g) algorithm so as to meet a specific per node throughput requirement.

Acknowledgment

A part of this work was supported by the national project "Research and Development of Technologies for Realizing Disaster-Resilient Networks", promoted by the Ministry of Internal Affairs and Communications (MIC), Japan.

References

[1] B. Kannhavong, H. Nakayama, A. Jamalipour, Y. Nemoto, and N. Kato, "A survey of routing attacks in manet," *IEEE Wireless Communications Magazine*, vol. 14, no. 5, pp. 85–91, October 2007.

[2] S. Kurosawa, H. Nakayama, N. Kato, A. Jamalipour, and Y. Nemoto, "Detecting blackhole attack on aodv-based mobile ad hoc networks by dynamic learning method," *International Journal of Network Security*, vol. 5, no. 3, pp. 338–346, November 2007.

[3] J. Chen, X. Cao, Y. Zhang, W. Xu, and Y. Sun, "Measuring the performance of movement-assisted certificate revocation list distribution in vanet," *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 888–898, November 2011.

[4] L. Zheng, S. Parkinson, D. Wang, L. Cai, and C. Crawford, "Energy efficient communication network design for demand response in smart grid," in *WCSP*, 2011.

[5] L. X. Cai, L. Cai, X. Shen, and J. W. Mark, "Capacity analysis of uwb networks in three-dimensional space," *IEEE/KICS Journal of Communications and Networks*, vol. 11, no. 3, pp. 287–296, 2009.

[6] M. Grossglauser and D. N. Tse, "Mobility increases the capacity of ad hoc wireless networks," in *INFOCOM*, 2001.

[7] P. Li, Y. Fang, J. Li, and X. Huang, "Smooth trade-offs between throughput and delay in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 11, no. 3, pp. 427–438, March 2012.

[8] M. J. Neely and E. Modiano, "Capacity and delay tradeoffs for ad-hoc mobile networks," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1917–1936, June 2005.

[9] R. Urgaonkar and M. J. Neely, "Network capacity region and minimum energy function for a delay-tolerant mobile ad hoc network," *IEEE/ACM Transactions on Networking*, vol. 19, no. 4, pp. 1137–1150, August 2011.

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

[11] ——, "Exact throughput capacity under power control in mobile ad hoc networks," in *INFOCOM*, 2012.

[12] ——, "Group-based two-hop relay with redundancy in manets," in *HPSR*, 2011.

[13] ——, "Generalized two-hop relay for flexible delay control in manets," IEEE/ACM Transactions on Networking, 2012, to appear.

[14] P. Gupta and P. Kumar, "The capacity of wireless networks," *IEEE Transactions on Information Theory*, vol. 46, no. 2, pp. 388–404, March 2000.

[15] C. M. Grinstead and J. L. Snell, *Introduction to Probability: Second Revised Edition.* American Mathematical Society, 1997.