2014 IEEE 11th Intl Conf on Ubiquitous Intelligence & Computing and 2014 IEEE 11th Intl Conf on Autonomic & Trusted Computing and 2014 IEEE 14th Intl Conf on Scalable Computing and Communications and Its Associated Workshops

Multi-source Broadcast Scheduling Algorithm of Barrage Relay Network in Tactical MANET

Junhua Yan, Chen Tian, Wenyu Liu, Lai Tu, and Benxiong Huang
Huazhong University of Science and Technology
Email: {junhuayan, tianchen, liuwy, tulai, huangbx}@ hust.edu.cn

***Abstract*—Modern tactical edge communication is turning to mobile ad hoc networks (MANETs) since the self-forming and self-healing nature of MANETs are advantageous in such a military environment. However, MANET faces a myriad of challenges in tactical field compared with traditional scenarios. Barrage Relay Network (BRN) is a special MANET designed from the ground up for the demands of tactical edge communications, which defines a rapid, robust and scalable broadcast mechanism based on an autonomous cooperative communication scheme. While, there is no research concerning about time slot scheduling in a multiple source broadcast scenario yet. Naive approach is not optimal in terms of the strict requirements of latency. In this paper, we put forward a centralized topology dependent scheduling algorithm designed for multi-source transmission scenarios in BRNs. Extensive evaluations demonstrate that the performance is not sensitive to the size of network and the throughput of network has improved nearly 23% compared with the naive approach.**

I. Introduction

Tactical edge communication has become more vital than ever before as soldiers now are exchanging vast amounts of data in the battlefield. And modern military communications is turning to mobile ad hoc networks (MANETs) at the edge since the self-forming and self-healing nature of MANETs is advantageous in a tactical military network.

However, the MANETs environment face a myriad of challenges in tactical field different from traditional scenarios. For example, the harsh propagation channel, frequent and rapid changes in the network topology, dynamic and intermittent links, and the requirement for very robust, low latency multimedia information decimation.

Thus TrellisWare Technologies, Inc. has developed a MANET system based on Barrage Relay Networks(BRNs) [1] (Section II for details). BRNs are a type MANETs designed from the ground up for the demands of tactical edge communications, which define a rapid, robust and scalable broadcast mechanism based on an autonomous cooperative communication scheme ([2], [3], [4]). In BRNs, packets ripple out from source nodes rapidly and reliably through the network. And each node, upon receiving a packet, relays the packet with one or more other nodes in a cooperative way without any coordination other than rough TDMA slot-level synchronization [1]. This turns harmful collisions into helpful cooperation and none of cooperating nodes nor the receiving nodes has need to know the number of cooperating nodes or their identity, which are optimal for single source broadcast scenarios.

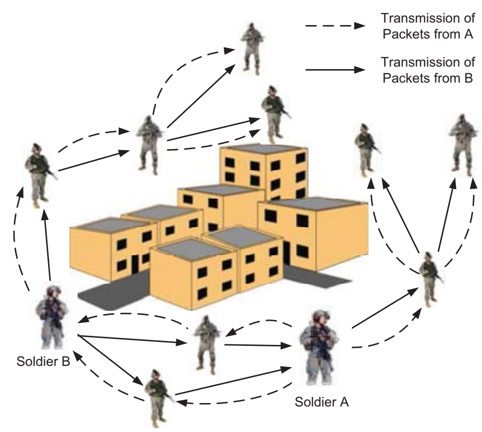

Fig. 1: A possible multi-source broadcast scenario in tactical field. Soldier A and B are broadcasting live-streaming videos captured by their camera.

Although BRNs are based on broadcast mechanism, there is no research concerning about time slot scheduling in a multiple source broadcast scenario yet. For example, as illustrated in Fig.1, a squadron of soldiers are waiting to storm the building while two of them (A and B) in different positions are broadcasting live-streaming videos captured by their camera. And we deem it is important to let each soldier involved in the mission have an idea about what is currently taking place in tactical field to guarantee the smooth completion of mission. More than this, commanders of the force also have need to receive messages from soldiers simultaneously for decision making and task allocation.

Naive approach is to let source nodes transmit their streaming videos one after another as what we did in single source broadcast scenario, which is not optimal due to the strict requirements of latency (Section III for detail). While BRNs are well suit to single-source broadcast scenario and scale optimally for broadcast data rate and latency [5], things are quite different when it comes to multi-source since only identical message can be transmitted and received in a cooperative way.

In this paper, we put forward a centralized topology-dependent scheduling algorithm designed for multi-source transmission scenarios in BRNs. The basic idea is the spatial reuse of time slot, which aims at letting as many nodes as possible transmit its packets in each time slot. We implement

978-1-4799-7646-1/14 $31.00 © 2014 IEEE
DOI 10.1109/UIC-ATC-ScalCom.2014.92

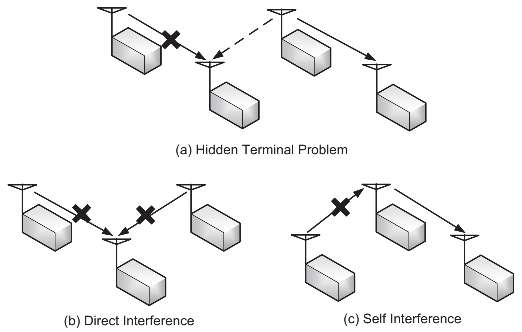

Fig. 2: Examples of collision types in wireless network.

the algorithm through simulations and it has shown that the performance is not sensitive to the size of network and the throughput of network has improved nearly 23% compared with the naive approach when there exists two source nodes.

The remainder of this paper is organized as follows: Section II reviews the background of tactical network and the core barrage relay concept via example. Section III presents the relevant design concerning about performing time slot scheduling in multi-source scenario in BRNs. And Section IV make a further introduction about the algorithm in details. At last, Section V evaluate the performance of our algorithms through simulation.

II. BACKGROUND

A. Interference Model in Wireless Network

Interference is a major issue in the wireless medium since nodes in wireless network are not allowed to transmit and receive messages simultaneously. Besides, when one or two nodes transmit a message to a common neighbor at the same time, the common node will not receive any of these messages.

Thus in multi-hop wireless networks, a single radio channel is spatially reused at different parts of the network and collisions may happen in three cases as illustrated in Fig.2 [6].

B. Tactical MANETs

Tactical networks are required to support military operations in areas without access to a fixed network infrastructure and characterized by frequent changes in network topology due to node mobility and intermittent connection. Modern military communications is turning to MANETs at the edge since the self-forming and self-healing nature of MANETs is advantageous in a tactical military network. More specifically, it can take advantage of MANETs technology to efficiently distribute content using available links, utilize cheap processing and memory to distribute content from the edge, and make use of content correlation to anticipate content need and proactively deliver content [7].

C. BRNs Review

BRNs are a type of MANETs designed for tactical edge communication. This section review the core barrage relay concept via an example and relies heavily on the descriptions previously provided in [1][5][8][9]. In BRNs, the most two critical network capabilities are time division multiple access (TDMA) and a method of autonomous cooperative communication. Autonomous cooperation is a key component of BRNs as it enables highly relay of packets and leads to significant increases in network capacity. It ensures that the concurrent transmission of packet carrying identical data will not result in collision or destructive interference, but rather in a cooperative way that increases the probability of reception. The autonomous cooperative communication capability is based on phase dithering (cf., [10]) and modern, turbo-like error correction (cf., [11]). And all nodes utilize a common TDMA framing format requires coarse slot-level synchronization which can be achieved through a low overhead pilot signaling [1].

D. Single Source Broadcast in BRNs

With the capabilities described above, the barrage relay broadcast mechanism is illustrated in Fig.3 with a single source. Generally, M-slot TDMA framing is assumed for some $M \geq 3$ and here we assume $M = 3$ with slots labeled A, B, C. Suppose on slot A of the first TDMA frame the black node transmits a packet and all nodes that are defined as one hop away from the source receive this packet. These nodes are then relay the same packet on slot B. Hence nodes that are two hops away from the source node receive the packet and can in turn relay it to the nodes that are three hops away from the source on slot C. Nodes that are three hops away from the source relay on slot A of the second TDMA frame. Packets thus propagate outward from the source in a decode-forward way. Besides, to prevent relay nodes from propagating back towards the source, each node relay a given packet only once.

Observe that a number of two-hop nodes in Fig.3 receive the same packet on the same time slot from different one-hop nodes, which indicates that in BRNs nodes at a given hop counts always cooperate to communicate to nodes at the next level.

Another significant feature of BRNs is the spatial reuse of time slot, which enables packets to be pipelined into the source for transmission every M time slot. For example, the source node can simultaneously transmit a second packet since one-hop nodes will not receive the packet transmitted by the three-hop nodes during slot A of the second TDMA frame in Fig.3.

Besides, [12] has mentioned about access control with multiple sources in BRNs and emphasize the concept of *Barrage Access Control* (*BAC*) introduced in [8]. However, none of these research focuses on time slot scheduling algorithm in multi-source transmission scenario.

E. Unicast in BRNs

In order to enhance the network capacity of BRNs, a lot of researches have investigated about spatially separated unicast transmissions in BRNs by establishing *Controlled Barrage Regions* (*CBRs*) ([8], [12], [5], etc.).

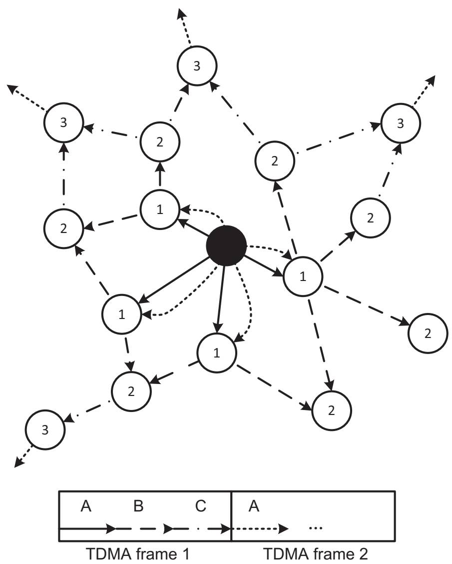

Fig. 3: Barrage relay network broadcast protocol for a three time slot TDMA frame format. The source node is black while relay nodes are numbered by their distance in hops from the source.

Multiple unicast transmissions are established in different portions of the network by specifying a set of buffer (i.e., sentry) nodes around the cooperating nodes based on the broadcast transmission of request-to-send (RTS) and clear-to-send (CTS) messages. For example, if the source broadcasts a RTS message and each forwarding nodes increments a hop-count field, the destination will have knowledge of the shortest cooperative path from the source. Likewise, the source will also know the distance from destination through the CTS message broadcasted by destination. Meanwhile, each forwarding nodes will know the shortest cooperative path and the length of the cooperative path on which it lies. Therefore, a CBR containing only the shortest cooperative path length (S) can be established by making all nodes on the cooperative paths larger than $(S+1)$ buffer nodes.

The buffer nodes in BRNs can only receive packets but not relay them. Thus the external packets will not propagate into the control region, nor do the internal packets propagate to the rest of network. Fig.4 shows three CBRs in a BRN with buffer nodes isolating the transmission.

III. Design

A. Intuition

While BRNs are well suit to single-source broadcast scenario and scale optimally for broadcast data rate and latency, things are quite different when it comes to multi-source. A naive approach is to let source nodes transmit their streaming videos one after another as described in Fig.5. Source node 2 broadcasts a packets first and three time slots are required for each node in network to receive this packet. Thus source node 8 will not broadcast its own packet until the forth time slot considering the collision. In the same way, node 2 will broadcast its second packet after four time slots when the latest packet broadcasted from 8 has been received by all nodes in network. In total, seven time slots are needed for node 2 and node 8 respectively broadcast a packet in the network.

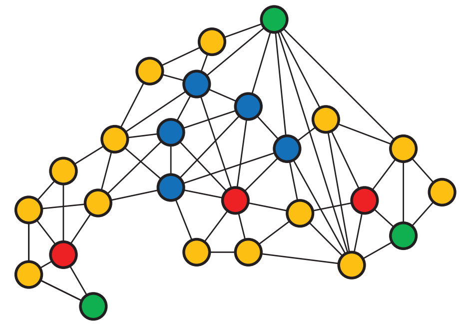

Fig. 4: Three unicast flows in a BRN. Source, destination, relay, and buffer nodes are colored green, red, blue, and yellow, respectively.

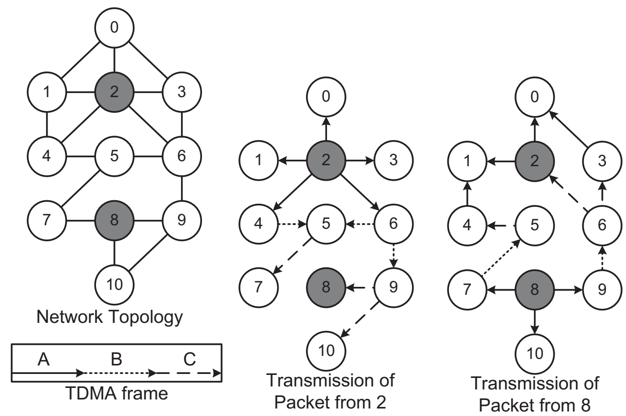

Fig. 5: Barrage relay process in a multi-source(node 2 and 8 in gray) broadcast scenario with naive approach.

Therefore, we consider to introduce the spatial reuse of time slot when transmitting different packets in BRNs for the sake of reducing the number of time slots required for transmission. For example, when node 2 broadcasts a packet to its one-hop neighbors, node 8 can also broadcast its packet simultaneously since no collision will be caused. Thus the number of time slots required for transmission can be reduced to seven.

B. Overview

The core of our algorithm is the spatial reuse of time slot. Different from the spatial reuse of time slot we have mentioned in a single broadcast scenario in Section II, it aims at letting more nodes transmit packets from different sources concurrently without collision.

To achieve this, data structures should be redefined since nodes will not merely relay what they have received in the next time slot with the existence of multiple sources. Besides, we define the status of nodes in BRNs to establish the collision detection mechanism for collision-free concurrent transmissions.

C. Data Structure

Due to the existence of packets from different sources in a multi-source broadcast scenario, nodes are no longer allowed to immediately relay packet in the next time slot after they have received it. Instead, they will store the packet first upon receiving and forward it on a specific time slot.

In order to make efficient utilization of every time slot to maximize network throughput, *unnecessary* transmission should be avoided since it may prevent nodes within two-hop distances from transmission considering the collision (see Fig.2). Here, *unnecessary* means a node relays the packet to its one-hop neighbors while all of them have already received it. Thus it requires each node to check whether all of its neighbors have received this packet before they store it for transmission. To achieve this goal, we introduce the method presented in [13] to make each node learn what packets its neighbors have based on reception report and intelligence guess. Specifically, each node announces to its neighbors the packet it receives upon reception and in the absence of deterministic information, it estimates the probability that a particular neighbor has a packet as the delivery probability [14] of the link between the packet's previous hop and the neighbor.

To sum up, each node maintains the following data structures:

- *Output queue*: each node has a output queue of packets that is necessarily to be forwarded.
- *Packet info*: the node keeps a hash table keyed on packet-id, which we call the *packet info*, to indicate the probability of each table having the packet. Upon receiving a packet, the node will first check its *hash info* to decide whether it is necessary to transmit the packet.

D. Node Status

In the process of scheduling, nodes in BRNs are classified into four statuses: *Request*, *Idle*, *Block* and *Relay*. And the detailed explanation of each status is presented as follows.

- *Request*: source nodes are set in *Request* status at the start of transmission and will not change its status until they have transmitted all of its own packets. And packets from other *Request* nodes always have priority over its own considering latency if no priorities are assigned to each source node. That is to say, once a *Request* node receives a packet producing by other *Request* nodes, it will relay this packet first other than transmitting new packet when alloted a time slot.
- *Idle*: nodes in *Idle* status are neither receiving a packet in current time slot nor having some packets in *output queue* to relay in the following time slot.
- *Block*: nodes that are currently receiving a packet or have some packets to forward in the *output queue* are set to be *Block*. Since packets are transmitted in a cooperative way in BRNs, *Block* nodes at a given hop count are further divided into a group based on the content of packets and nodes in the same group will be scheduled to transmit identical data simultaneously for the utilization of cooperative communication. For example, nodes that are one hop away from the source in Fig.3 are in *Block* status during time slot A of the first TDMA frame since they are receiving packet currently. Meanwhile, they are in the same *Block* group and will transmit the packet in a cooperative way.
- *Relay*: nodes that are currently transmitting packets are in *Relay* status and will change its status into *Idle* or *Block* according to whether they have other packets in its *output queue* for relaying in the following time slot.

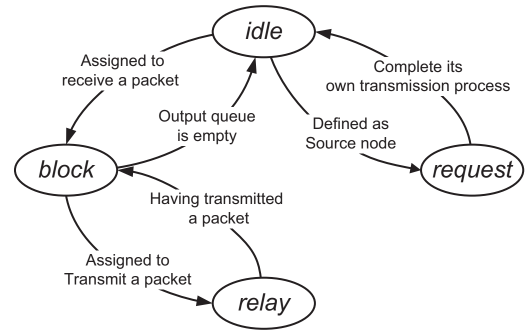

Fig. 6: Status transition in the process of scheduling.

All nodes are initialized into *Idle* at the start of transmission and changes its status per time slot according to whether it will receive or relay a packet except for source nodes, which are always in *Request* status until the end of its own transmission. The transition process is presented in Fig.6.

E. Collision Type

According to our definition about nodes' status, it is clear that only *Request* nodes and *Block* nodes will contend for time slot to transmit packet. Thus collisions will only happen among *Block* nodes and *Request* nodes within two-hop distance. More specifically, since *Block* nodes are divided into different *Block* groups, we should investigate the *Block* nodes in each *Block* groups to see whether they will collide with *Request* nodes or *Block* nodes in other group.

In order to maximize throughout of the network, *Request* nodes always give priority to transmit packets than *Block* nodes in case of collision.

F. An Example

Here we illustrate the transmission process in the first eight time slots through algorithm scheduling within a network of 11 nodes illustrated in Fig.7. Three source nodes (numbered 2, 6, and 8 respectively) are assigned in this scenario and presented in gray.

As is shown in Fig.7, eight time slots are enough for each source node broadcasts one packet respectively in the network. And in the eighth time slot, source node 8 are allowed to transmit the second packet to its one-hop neighbors when its three-hop neighbors are relaying the first packet concurrently. While in the same circumstances, the naive approach (only one packet is broadcast in network and the transmission of a new packet won't start until all nodes in network have received the last one) needs ten time slots in total.

The core of our algorithm is the spatial reuse of time slot, which will be detailed in Section IV. And with the incremental of the scale of network, spatial reuse of time slot will be more common and the time slot can be reduced further compared with the naive approach, which will be testified in Section V through simulation.

IV. Scheduling Algorithm

The algorithm target for multiple source broadcast scenario in BRNs is based on the spatial reuse of time slot, which aims at letting as many nodes as possible transmit different packets to its neighbors simultaneously. And it is executed each time slot to make assignment for the next time slot beforehand.

A. Scheduler

To perform centralized scheduling algorithm, a ***Scheduler*** is introduced in BRNs, which maintains a *Request Queue* (*RQ*), a *Block Group* (*BG*), a *Forwarding Queue* (*FQ*) , and the topology of network. The *Request Queue* and *Block Group* are obtained for detecting collision and assigning time slot based on the topology of network. And the scheduling result about the next time slot, including designated nodes, packet id, and time slot information, is recorded in the $Forwarding\ Info$, which can be accessed by each node in network.

Request Queue records the identity information (i.e., node id) about *Request* nodes that will contend for time slot to transmit new packet. The ***Scheduler*** will first investigate about all *Request* nodes in network, and a *Request* node will be added in the *Request Queue* only on the condition that the latest new packet it transmitted has been broadcast two hops away.

Block Group records the packet id to indicate packets that are waited to be transmitted in network together with relevant information about the transmitters, considering that they can be transmitted in a cooperative way to improve throughput. Once a packet is allowed to transmit in the next time slot, the relevant information stored in *Block Group* will be deleted. One important thing should be pointed out is that even all *Block* nodes in network are classified into one or more *Block* groups according to the packets in its $output\ queue$, not all nodes in *Block Group* are in *Block* status considering that *Request* nodes not only broadcast new packets but also relay what they have received if necessary.

When ***Scheduler*** have decided nodes for transmission in the next time slot, it will immediately record relevant scheduling information in the *Forwarding Queue*, which is keyed on packet id and garbaged collected every few seconds. Besides, in the process of scheduling, relevant information in *Forwarding Queue* will help the ***Scheduler*** to decide whether it is necessary to put a node in relevant *Block* group by checking the transmission records to see whether all neighbors have received this packet before.

B. Detection

As we have mentioned in Section III, collisions will only happen between *Request* nodes and *Block* nodes in BRNs. And the *Block* nodes in the same *Block* group are treated as a whole to check whether it will collide with *Request* nodes in the *Request Queue* or other *Block* groups if they transmit a packet simultaneously.

Thus by investigating the topology information about nodes in *Request Queue* and *Block Group* it is easy for the ***Scheduler*** to find possible collisions in network.

C. Algorithm

The algorithm is based on the spatial reuse of time slot and able to establish TDMA slot assignment that are non-conflicting with high probability. To maximize throughput of the network, we have established two principles in the process of time slot scheduling. First, concurrent transmission of different packets without collision should be given most priority to in network regardless of the order of packets to make full use of time slot. Second, nodes in *Request Queue* always have priority to transmit its packets in network in case of collision.

In each time slot, ***Scheduler*** executes the procedure presented in Algorithm 1 for collision detection and time slot assignment.

TABLE I: SUBROUTINE USED

$SizeOf(V)$	The number of nodes in V
$\_SizeOf(V)$	The number of block groups in V
$Distance(m,n)$	Distance between node m and n measured by hops
Request(m,n)	Corresponding operations when *Request* nodes m,n are able to transmit simultaneously
ReqBlock(m,n)	Corresponding operations when *Request* node m and a group of *Block* nodes n are able to transmit simultaneously
Block(m,n)	Corresponding operations when *Block* group m and n are able to transmit simultaneously
RequestOne()	Corresponding operations when only a *Request* node has packets to transmit or collide if more than two *Request* nodes transmit concurrently
ReqBlockCollision()	Corresponding operations when *Request* node and *Block* groups collide if transmit concurrently
BlockOne()	Corresponding operations when only a *Block* group exists
BlockCollision()	Corresponding operations when *Block* groups collide if transmit concurrently

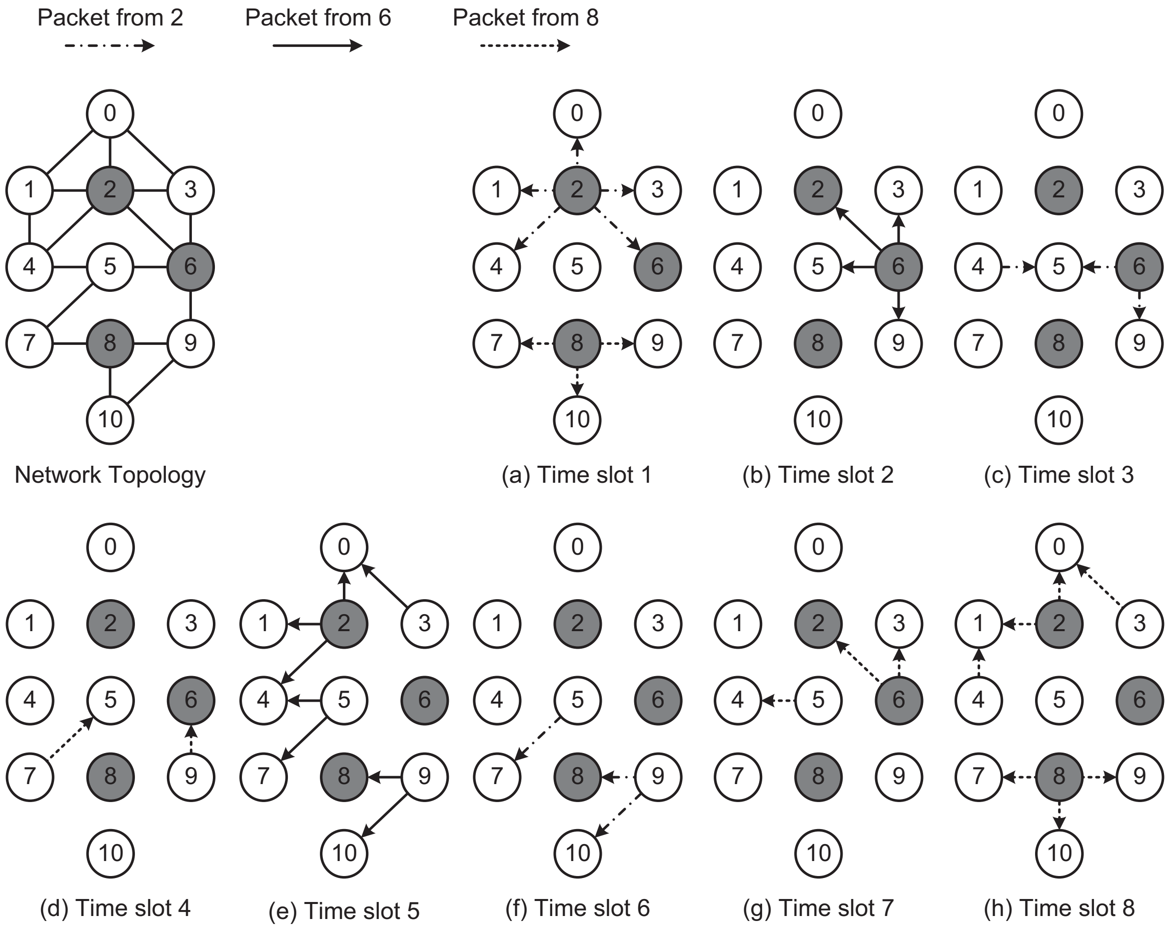

Fig. 7: Scheduling result about packets transmission in the first eight time slot with three source nodes(2, 6 and 8 in gray) in a barrage relay network. The transmission of packets from different source nodes are presented with different kinds of lines.

Having detected the collision that may happen in network, corresponding operations are executed such as creating new *Block* groups, erasing old-dated informations, changing nodes' status, and putting designated nodes in *Forward Queue* etc. Here we merely describe the process when two *Request* nodes are designated to transmit simultaneously in Algorithm 2 due to the limitation of space.

V. EVALUATION

A. Setup

In simulation, we set the video at a frame rate of 25 frames per second and the number of time slots per frame $M = 3$, while the maximum lag should be less than 500 milliseconds.

In order to compare the naive barrage relay broadcast scheme and our algorithm described above, the following methodology is employed. Let n be the number of nodes to be simulated and let R_{80} be the distance corresponding to SNR_{80}. These n nodes are distributed randomly and independently in an $L(n) \times L(n)$ square where $L(n) = \frac{4}{3} R_{80} \sqrt{\frac{\pi n}{\log n}}$ based on geometric graph model (cf., [15]). When randomly assigned different number of source nodes in BRNs, a 30 broadcast burst is injected into the designated nodes for transmission. This experiment is then repeated for enough node distributions so as to ensure statistical significance.

The performance of our scheduling algorithm in multi-source broadcast scenario is studied through the following two key metrics and compared with the traditional barrage broadcast scheme.

(i) *Latency*: the number of time slots required for each source node that located randomly in the network successfully delivers one packet.

(ii) *Goodput*: the number of broadcast packets successfully delivered per node and per time slot average over nodes in network. Considering that our algorithm is designed to perform time slot scheduling in the scenario when multiple live-streaming video are broadcast in BRNs simultaneously, there involved a strict limitation for transmission delay (500 milliseconds as defined above). Thus only if packets reach nodes ahead of due time will we regard it successfully.

Algorithm 1: Collision Detect and Time-slot Schedule

```
Procedure TimeSlotScheduling()
for all node i in Request status do
    if the latest packet it has transmitted has been
    broadcasted two-hop away then
        RQ ← i
    end if
end for
if SizeOf(RQ) == 0 && _Sizeof(BG) == 0 then
    return
end if
if SizeOf(RQ) > 1 then
    for ∀m,n ∈ RQ do
        if Distance(m,n) > 2 then
            Request(m,n)
            return
        end if
    end for
end if
if SizeOf(RQ) && _Sizeof(BG) ≥ 1 then
    for ∀m ∈ RQ, n ∈ BG do
        if Distance(m,n) > 2 then
            ReqBlock(m,n)
            return
        end if
    end for
end if
if SizeOf(RQ) == 0 && _Sizeof(BG) > 1 then
    for ∀m,n ∈ BG do
        if Distance(m,n) > 2 then
            Block(m,n)
            return
        end if
    end for
end if
if SizeOf(RQ) ≥ 1 then
    if _Sizeof(BG) == 0 then
        RequestOne()
        return
    else
        ReqBlockCollision()
        return
    end if
else
    if _Sizeof(BG) == 1 then
        BlockOne()
        return
    else
        BlockCollision()
        return
    end if
end if
end procedure
```

Algorithm 2: Corresponding operation when only a *Block* group exists

```
Procedure BlockOne()
m ∈ BG
FQ ← m
create BG(packet_m)
BG = BG − {m}
for all node i that is adjacent to m do
    if not all of i's neighbors have received this packet
    then
        BG(packet_m) ← i
        if Status_i(t)! = Request then
            Status_i(t+1) = Block
        end if
    end if
end for
end procedure
```

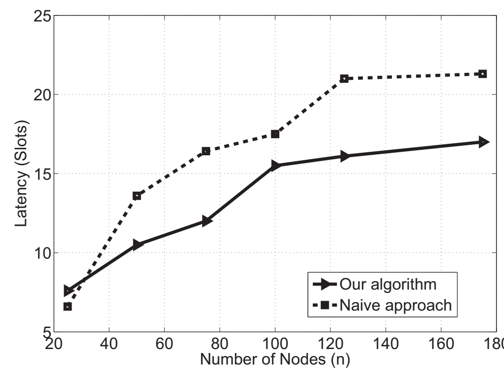

Fig. 8: Latency comparison of scheduling algorithm and the naive broadcast mechanism.

In addition, we further evaluate the required time slot with different number of source nodes broadcasting simultaneously in a fixed network and check the improvement by repeating experiment with the naive approach.

B. Different Network Size

In this section, we make an evaluation about our algorithm in different network size by investigating the metrics described above. Thus the number of source nodes (*N*) in network should be fixed, and we assign *N* to 2.

Fig.8 and Fig.9 compare the latency and goodput respectively, of barrage relay scheduling algorithm to the naive broadcast mechanism as the size of network increases in multi-source transmission scenario. The latency for each source in network broadcast one packet in BRNs has decreased nearly 20% with our scheduling algorithm according to Fig.8, which means the improvement of throughput and the reduction of the propagation delay. Furthermore, the average goodput per time slot has also improved in comparison with the naive approach on the delay limitation of 500 milliseconds. More specifically, the average goodput is nearly 0.089 with our algorithm while

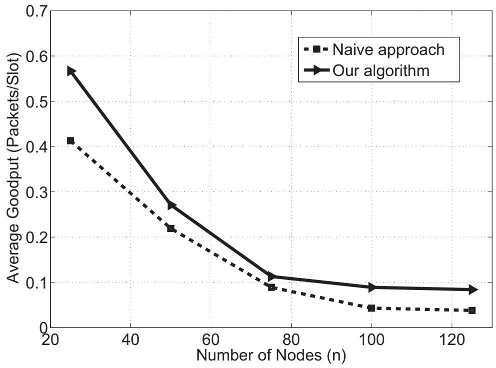

Fig. 9: Goodput comparison of scheduling algorithm and traditional broadcast mechanism with different size of network.

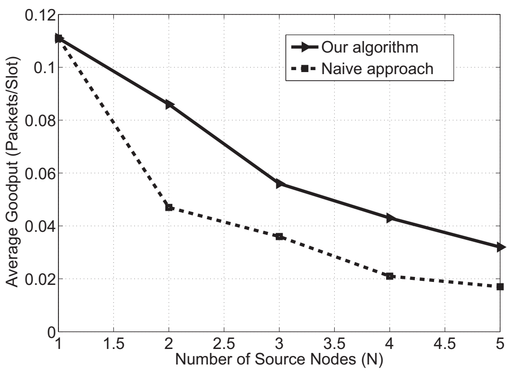

Fig. 10: Goodput comparison of scheduling algorithm and traditional broadcast mechanism in a fixed network.

the average goodput is 0.043 with the naive approach for 100 nodes, which has approximately improved 50%.

C. Different Source Number

Having evaluated the performance of our algorithm in different size of the network, we make a further investigation in a fixed network (75 nodes in total) with the number of source nodes ranging from one to five.

As is illustrated in Fig.10, the goodput has nearly doubled in a fixed network with different number of source nodes when compared with the naive approach, which indicates the validity of packets can be further guaranteed through our scheduling algorithm.

VI. Conclusion

This paper propose a centralized time slot scheduling algorithm designed for multiple source broadcast scenario in barrage relay networks. Although no previous research has concerned about this scenario before, we think this is strategic since with the support of advanced hardware, multi-source live-streaming video transmission will be common in future tactical field. And the simulation apparently shows that our algorithm has increased the throughput of network under the limitation of latency when compared with the naive approach, which applies the mechanism about single source broadcast in BRNs to multi-source scenario.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments. This work is partially supported by "National Natural Science Foundation of China (No. 61202107, No.61100220, No. 61202303)", by "National High Technology Research and Development Program of China (863 Program No. 2014AA01A702)", by "Natural Science Foundation of Hubei Province (No. 2014CFB1007), by "National Key Technology Research and Development Program of China (No. 2012BAH46F03)", by "National Science and Technology Major Project of China (Grand No. 2012ZX03001035-003)", and by the "Fundamental Research Funds for the Central Universities".

References

[1] A. Blair, T. Brown, K. M. Chugg, and M. Johnson, "Tactical mobile mesh network system design," in *Military Communications Conference, 2007. MILCOM 2007. IEEE*. IEEE, 2007, pp. 1–7.
[2] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *Communications Magazine, IEEE*, vol. 42, no. 10, pp. 74–80, 2004.
[3] D. K. Lee and K. M. Chugg, "A pragmatic approach to cooperative communication," in *Military Communications Conference, 2006. MILCOM 2006. IEEE*. IEEE, 2006, pp. 1–7.
[4] J. N. Laneman, "Cooperative communications in mobile ad hoc networks," *IEEE Signal Processing Magazine*, vol. 1053, no. 5888/06, 2006.
[5] T. R. Halford, K. M. Chugg, and A. Polydoros, "Barrage relay networks: System & protocol design," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2010 IEEE 21st International Symposium on*. IEEE, 2010, pp. 1133–1138.
[6] A. A. Bertossi and M. A. Bonuccelli, "Code assignment for hidden terminal interference avoidance in multihop packet radio networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 3, no. 4, pp. 441–449, 1995.
[7] K. Gremban, "Content-based mobile edge networking (cbmen):darpa-baa 11-51," 2011.
[8] A. Blair, T. Brown, K. Chugg, T. Halford, and M. Johnson, "Barrage relay networks for cooperative routing in tactical manets," in *Proc. IEEE Military Comm. Conf., San Diego, CA*, 2008.
[9] R. Ramanathan, "Challenges: a radically new architecture for next generation mobile ad hoc networks," in *Proceedings of the 11th annual international conference on Mobile computing and networking*. ACM, 2005, pp. 132–139.
[10] I. Hammerstrom, M. Kuhn, and A. Wittneben, "Cooperative diversity by relay phase rotations in block fading environments," in *Signal Processing Advances in Wireless Communications, 2004 IEEE 5th Workshop on*. IEEE, 2004, pp. 293–297.
[11] K. M. Chugg, P. Thiennviboon, G. D. Dimou, P. Gray, and J. Melzer, "New class of turbo-like codes with universally good performance and high-speed decoding," in *Military Communications Conference, 2005. MILCOM 2005. IEEE*. IEEE, 2005, pp. 3117–3126.
[12] T. R. Halford and K. M. Chugg, "Barrage relay networks," in *Information Theory and Applications Workshop (ITA), 2010*. IEEE, 2010, pp. 1–8.
[13] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: practical wireless network coding," in *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 4. ACM, 2006, pp. 243–254.
[14] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
[15] P. Gupta and P. R. Kumar, "Stochastic analysis, control, optimization and applications: A volume in honor of wh fleming," 1998.