Causal Ordering Group Communication for Cognitive Radio Ad Hoc Networks

Liming Xie*†, Xiaohua Jia† and Kunxiao Zhou*†
*Computer School, Wuhan University, Wuhan, China
†Department of Computer Science, City University of Hong Kong, Hong Kong
xliming2@student.cityu.edu.hk, jia@cs.cityu.edu.hk, zkunxiao2@student.cityu.edu.hk

***Abstract*—In this paper, we discuss the causal ordering group communication for cognitive radio (CR) networks. The issue of causal ordering has been studied extensively by previous works. However, these works considered the problem at the application layer and the methods in the works incurred high communication overhead and the long latency for message delivery. In this paper, we discuss the causal ordering at the network layer.The problem of our concern is: given a group communication request in a CR network, how to set up the connection so that the causal ordering of the group communication can be preserved, and the bandwidth consumption of the communication is minimized. We propose a two-phase method to solve it. In the method, we first construct a multicast tree for the communication, and then assign slots for all tree links. It is proved that the method can preserve the causal ordering of messages without extra communication overhead nor the latency for delivering messages. Simulations are conducted to show the performance of our proposed method.**

***Index Terms*—Causal ordering, Group communication, Slot assignment, Cognitive radio networks.**

I. Introduction

Cognitive Radio (CR) networks have received great attentions in recent years because they can exploit the existing wireless spectrum opportunistically and improve the spectral efficiency. In such networks, the secondary users equipped with cognitive radio can sense and access the "spectrum hole" [1] unoccupied by the primary users. Since nodes have different surrounding environment in CR networks, the secondary users have different available spectrum bands or channels. This characteristic introduces additional complexities for coordination and communication between nodes in CR networks.

In this paper, we study the issue of supporting causal ordered group communication in CR networks. Group communication [2]–[4] basically implies multipoint to multipoint communication. In this paper, it refers to such kind of communication where each member of the group can multicast to the group and a message from a member must be delivered to all other group members. An example of such applications is the video conference system, where each message sent by a participant should be delivered to all other participants and every participant can be a sender. Causal ordered group communication means that if the messages for the communication have the *precede* relation [5], then they should be delivered to all the destinations in the same causal order. The causal ordering of messages makes all group members have the same logical view of all group activities. Causal ordered group communication is an important and widely used communication means for many applications, such as group ware, group discussion, group editing and so on.

Extensive researches have been done on causal ordered group communication, such as [5], [6]. However, the existing works considered the causal ordering at the application layer. There are two disadvantages. One is that in order to ensure the causal ordering, all the algorithms requires to attach a large amount of extra dependency information to each message. This brings lots of communication overhead to the network. The other is that all the methods incur a long latency for delivering the messages. For example, in paper [5], the messages need to be ordered at all much high nodes. When a message arrives a node but some messages preceding it have not yet arrived, this message has to be buffered waiting for the arrival of all the messages preceding it. This will incur a long latency for delivering messages.

In this paper, we study how to preserve the causal ordering in group communication without extra communication overhead nor the latency for message delivery. We propose a novel method to preserve the causal ordering at the network layer. The task we face is, given a group communication request in a CR network, to find a routing tree and schedule the message transmission so that the causal ordering of the group communication is preserved, and the bandwidth consumption of the communication is minimized. The bandwidth consumption of the group communication is measured as the total bandwidth consumption for delivering all the messages to all group members. In our method, we first construct a multicast tree and then schedule the transmissions along the tree links. We prove that the method can ensure the causal ordering without requiring the message to carry the extra information about causality nor incurring the message delivery latency. Extensive simulations are conducted to show the efficiency of our method.

II. Related work

Causal ordering problem has been well discussed in the previous works [6]–[9]. In paper [6], a causal ordering method was proposed and implemented in ISIS system. In this method, if a message should be sent out, this message will be piggybacked with all other messages that causally related with it to a large package to be sent out. In [7], the authors proposed another algorithm to implement the causal ordering. However, this method incurred a large number of communication overhead, since it required each message to attach an extra data

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

structure. To reduce the overhead, the paper [8] presented an adaptive causal ordering algorithm which is suitable for the mobile computing environments. In [9], the author improve the algorithm designed in [8] so that the causal order among can be preserved in a multicast environment.

Some other works studied the total ordering problem which includes the causal ordering. In [6], a protocol ABCAST was original proposed in ISIS system to guarantee the total ordering of messages. In [5], the author proposed a multicast mechanism using propagation trees to guarantee the total ordering. Messages are ordered at each non-leaf node and then are delivered to the child nodes. In paper [10], the authors developed an early-delivery total ordering protocol to reduce the delivery latency. However, all the above works required the communication overhead and incurred the latency for message delivery. No matter how to develop a novel algorithm or improve an existing algorithm, they all required the communication overhead and incurred some message delivery latency.

III. PROBLEM FORMULATION

A. System Model

We consider a CR network that has no fixed infrastructure. Each node has two radios, one for data transmission and the other for control messages. The radio for data transmission is assumed to be capable of sensing the locally available channels for the node by using spectrum sensing technique. Hence, it constantly maintain a set of channels that are available to it. We also assume this radio can do channel switching at packet level. That is, it can transmit a packet on one channel and switch to another channel for transmitting the next packet. All the data channels are assumed to have the same bandwidth. Similar to the previous works [11]–[13], we assume there is a common control channel available to all nodes in the network. The control radio at each node uses the control channel to exchanges messages with others for topology maintenance and other control operations. The locations of the nodes are considered to be static or change slowly. Node mobility is not addressed in the paper.

The network is modeled as an undirected graph $G(V, E)$, where V is the set of nodes and E is the set of links connecting the nodes. A link $(u, v) \in E$, iff nodes u and v have a common data channel available and they are within each other's transmission range. The communication links are bidirectional, following the IEEE 802.11 P2P model that requires each packet to have an acknowledgment. In this paper, we adopt the TDMA scheme as the MAC sub-layer scheme. That is, time is divided into constant size timeslots and a frame consists of K timeslots. In TDMA scheme, the message is transmitted and received in one slot time. For a node, the process of receiving a message and then sending it out is done in one frame time. Thus, the delay of message delivery at each node is constant. In CR network, each node has a set of available channels and each channel is divided into timeslots. Thus, we use a pair (c, t) to denote a slot at a node, where c refers to the channel and t represents the timeslot. A slot (c, t) is said to be free for node u if 1) u does not transmit or receive in timeslot t, because there is only one radio used for data transmission; and 2) none of the nodes that interfered by u occupies slot (c, t). A slot (c, t) is said to be free for link (u, v) if the slot is free for both node u and v.Note that in this new case, the delay of message delivery at each node is still constant which is no large than one frame time.

B. Problem Statement

We first introduce the following definitions.

Definition 1: Group communication: a group of nodes communicate with each other to perform a common task. Each member in the group can send messages to the group and each message from a member must be delivered to all other group members.

During the group communication, one message is delivered to multiple destinations and a node receives messages from different sources. Therefore, the order in which the messages received at a node may be different from the others. However, the messages generated for the group communication have causal dependency relationships between them. The causal dependency relation between messages, called *precede* relation "→", can defined as follows.

Definition 2: Precedence of messages: a message m_1 is said to *precede* another message m_2 (i.e. $m_1 \rightarrow m_2$), iff:

1) m_1 and m_2 are generated by the same node u, and m_1 is sent out before m_2;

2) m_1 and m_2 are generated by the nodes u and v, respectively, and v receives m_1 prior to sending m_2; or

3) $\exists\ m_3$, $m_1 \rightarrow m_3$ and $m_3 \rightarrow m_2$.

With this *precede* relation definition, we can define the causal ordered group communication.

Definition 3: Causal ordered group communication: if two messages have "→" relation, all the group members which receive both messages should receive the two messages in the same "→" order.

According to above definition, if a node u sends a message m_2 after sending message m_1, and the node v sends a message m_3 after receiving m_2, then the node w which is in the communication group should receive m_3 after receiving m_1. Otherwise, the precede relation between m_1 and m_3 cannot be maintained at node w. On the other words, the causal ordering will be violated at node w. Fig.1 illustrates the case of the violation of the causal ordering.The message m_3 arrives at node w earlier than m_1.In this paper, we consider how to preserve the causal ordering.

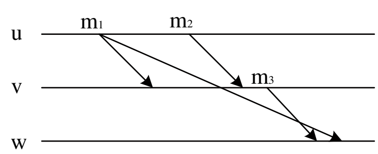

Fig. 1. An example of the violation of causal ordering.

In group communication, each member can send messages and the message sent by it should be delivered to all other group members. Therefore, the bandwidth consumption of a group communication includes the bandwidth consumed by the group members and the relay nodes for delivering the messages. Furthermore, the bandwidth consumption of a node can be measured by the number of transmission slots required by the node to deliver messages for the group communication. Thus, we define the bandwidth consumption of a group communication as follows.

Definition 4: Supposing each member multicasts one message to the group, the bandwidth consumption of a group communication is measured as the total number of transmission slots required for delivering all the multicast messages to all group members.

Now, we define our problem as follows. Suppose we are given a group communication request and the bandwidth requirement is 1 (in terms of number of timeslots in a frame). The problem is how to set up a group communication connection, which connects all the group members, so that the causal ordering of the group communication is preserved and the bandwidth consumption of the group communication is minimal.

In this paper, we assume that the bandwidth requirement of the group communication is 1. That means each group member requires one slot for transmission. Our method can be easily extended to the case where the bandwidth requirement is more than one.

IV. The Proposed Solution

In this section, we present a two-phase method to solve the problem. The first phase is to find a multicast tree for each group member so that the messages sent by the member can reach all other group members. The second phase is to assign transmission slots for each node that delivers messages for the communication so that the total bandwidth consumption of the communication is minimal.We can prove that this method can guarantee the causal ordering (see section V for details).

A. MST based algorithm

We consider a group consisting of M group members. In order to make sure each message can be delivered from the source to all the destinations, M multicast trees are required. However, if we consider the group communication as M multicast requests and find a multicast tree for each request, we will get a very complicated routing graph for the group communication and the causal ordering of messages is hard to guarantee.In this paper, we introduce a minimal spanning tree (MST) based routing algorithm to find M multicast trees together.

As we know, MST is a tree that spans all vertices of the graph and has the minimal total weight. In our model, each link has the same weight which is supposed to be 1. Therefore, MST method constructs a tree which has minimal number of tree links. Less number of tree links will bring less bandwidth consumption. However, if the MST method is operated on the original graph G, it may get a very long tree, which causes the long delay of the communication. In order to constrain the length of the MST, we introduce a MST based algorithm.

First, we generate a virtual complete sub-graph $G_c(V', E')$, where V' consists of all the group members and E' are the edges between any two vertices in G_c. The edges of this sub-graph are virtual. In order to make this virtual sub-graph G_c make sense, we replace each virtual edge in this graph with the corresponding shortest path in the original graph G. Then, we get a replaced graph G_p. Fig.2 shows an example. In Fig.2(a), the graph in red refers to the virtual complete graph which consists of 4 vertices and 6 edges. The red dashed lines are the edges of this complete graph. And Fig.2(b) shows the replaced graph of the virtual complete graph.

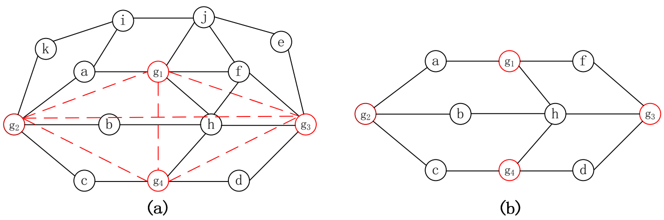

Fig. 2. (a) The virtual complete graph G_c, (b) The replaced graph G_p.

After obtaining the replaced graph G_p, we apply the MST algorithm over G_p. The algorithm can start from any member chosen from the group. Initially, the MST contains only the chosen member. We denote N to be the candidate set which consists of the neighbors of all nodes in the MST. Each time, we randomly select a node from N to join the MST. The process is repeated until all the group members join the tree. Then we can get a MST. If some leaf nodes are not the group members, we just remove such nodes and their adjacent links from the MST. Then we can get a multicast tree, where the root is the chosen member and the leaf nodes are all other group members.

However, starting from different members, the algorithm will find different multicast trees. In our method, we choose the shortest multicast tree as the final tree. Consider the tree as a new graph. For each group member, there is a tree transformed from the graph by choosing the member as the root and all other members as the leaf nodes. In this way, we can get M multicast trees from the tree generated by the MST based algorithm. Therefore, for a group member, it sends message to all other group members through its corresponding tree where the root is this member and leaves are all other group members.

B. Slot Assignment

After multicast trees are constructed, slot assignment should be done for each tree so that the total bandwidth consumption of the group communication is minimized. When all the trees have been assigned the proper slots for transmission, the group members can start to send messages using the slots assigned to them. However, if the slot assignment of any multicast tree is failure, the group communication request will be rejected.

For each multicast tree, the slot assignment process starts from the root, going down along the multicast tree hop by hop until reaching the leaf nodes. The details of the slot assignment algorithm will be discussed later. If any node in the tree fails to be assigned proper transmission slots, the slot assignment of tree will be failure.

In order to minimize the bandwidth consumption of the group communication, we should minimize the bandwidth consumption of each multicast tree. Thus, we should minimize the bandwidth consumption of each non-leaf node in the multicast tree. Since the bandwidth requirement is 1, each non-leaf node in a multicast tree just needs to transmit one packet to all its child nodes. However, that node may need more than one transmission slots in CR networks. For example, the slots (1,4), (1,5), (2,5) are free for node a, the slot (1,4) is free for b and the slot (1,5), (2,5) are free for c. Note that b and c have no common free slots. In this case, if a needs to transmit a packet to b and c, it has to transmit twice and consume two transmission slots. However, transmitting more times leads to consuming more bandwidth. Hence, in order to minimize the bandwidth consumption of a node in a multicast tree, we should choose the least number of free slots for the node to transmit one packet to all its child nodes.

Now, we present a slot assignment algorithm to allocate least number of transmission slots for a non-leaf node u. In order to avoid interference, nodes always use the slots that are free for them to transmit or receive packets. Thus, to assign transmission slots for u, we first calculate the free slot sets of u and all its child nodes according to the rules discussed in section III-A. We use F_u to denote the set of slots that are free for u, and C to denote the set of the child nodes that haven't allocated the reception slots for receive the packet from u. Initially, C contains all the child nodes of u. If a slot in F_u is free for the child node v, we say the slot covers v. It means that the node u can use this slot to transmit a packet to its child node v. Thus, in order to assign the least number of transmission slots to u, we just need to pick the least number of slots in F_u to cover all the child nodes in C. This is a set cover problem, which is NP hard. We take a simple (and effective) greedy algorithm to solve the problem. Each time, we pick the slot that covers the most number of nodes in C as one transmission slot of u. Remove the nodes that covered by the slot from C and update F_u. This procedure continues until C is empty, which means all the child nodes can receive one packet from u.

V. Correctness Analysis

In this section, we use three lemmas and one theorem to prove that our method can guarantee the causal ordering of messages. Let $R_i(m)$ denote the event that node i receives the message m, and $S_i(m)$ denote the event that node i sends the message m. $R_i(m) \prec S_i(m)$ refers to the event $R_i(m)$ happens before $S_i(m)$. Let U denote the set that consists of all group members.

Lemma 1: Suppose messages m_1, and m_2 are generated by the same node u, and u sends m_1 prior to m_2. Our method ensures that for any group member z receiving both m_1 and m_n, $R_z(m_1) \prec R_z(m_2)$ is correct.

Proof: In our model, the FIFO protocol is employed as the transport protocol. That means the messages received first will be sent out first. Thus, m_1 will be received earlier than m_2 by the first relay node, since m_1 is sent out before m_2. Then, the first relay node will sent m_1 before m_2. Similarly, for all the nodes along the multicast tree which is generated by the method, they always receive m_1 before m_2 and then forward m_1 before m_2. Hence, $R_z(m_1) \prec R_z(m_2)$. ■

Lemma 2: Suppose messages m_1, and m_2 are generated by the nodes u and v respectively, and v sends m_2 after receiving m_1. Our method ensures that for any group member z receiving both m_1 and m_2, $R_z(m_1) \prec R_z(m_2)$ is correct.

Proof: Two situations should be considered.

1. Nodes u, v and z are on the same path.

There are three cases in this situation. Case 1: node z is between u and v on the path (see fig.3(1)). In this case, $R_z(m_1) \prec R_v(m_1)$ is correct. Since $R_v(m_1) \prec S_v(m_2) \prec R_z(m_2)$, we can get that $R_z(m_1) \prec R_z(m_2)$. Case 2: node v is between u and w (see fig.3(2)). Because of the FIFO protocol, we know $S_v(m_1) \prec S_v(m_2)$. Hence, $R_z(m_1) \prec R_z(m_2)$. Case 3: node u is between v and z (see fig.3(3)). It is obvious to know that $S_u(m_1) \prec S_u(m_2)$. Hence, $R_z(m_1) \prec R_z(m_2)$.

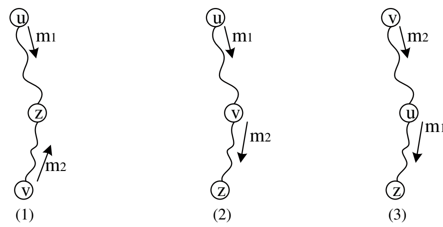

Fig. 3. The three nodes on the same path.

2. Nodes u, v and z are not on the same path.

According to the algorithm, we know that there must exist at least one node q which relays messages for any two of u, v and z (see Fig.4(1)). Otherwise, there will exist three relay nodes a, b and c so that a and b relay messages for u and v, b and c relay messages for v and z, and a and c relay messages for u and z (shown in fig.4(2)). So there is a circuit (formed by a, b and c) on the route which contradicts with the fact that there is no circuit in a MST. Hence, with the existence of q, we can get that $S_q(m_1) \prec S_q(m_2)$. Hence, $R_z(m_1) \prec R_z(m_2)$. ■

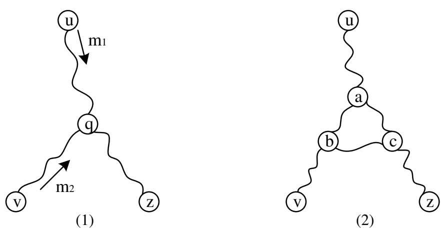

Fig. 4. The three nodes not on the same path.

Lemma 3: Suppose messages m_1, m_2, ... and m_n have the *precede* relation: $m_1 \to m_2 \to ... \to m_n$. Our method ensures that for any group member z receiving both m_1 and m_n, $R_z(m_1) \prec R_z(m_n)$ is correct.

Proof: This lemma can be proved by the method of induction.

First, we consider that there are only two messages, i.e, $n = 2$. According to Lemma 1 and 2, we know that the algorithm guarantees that for any destination z that should receive m_1 and m_2, $R_z(m_1) \prec R_z(m_n)$.

Then, suppose this lemma is correct when $n = k$. That means if $m_1 \rightarrow m_2 \rightarrow ... \rightarrow m_k$, the algorithm can ensure that for any destination z that should receive m_1 and m_k, $R_z(m_1) \prec R_z(m_k)$. Now, we should prove that when $n = k + 1$, the lemma is still correct. In order to prove it, two situations should be considered.

1. m_1 and m_k are generated by the same node u.

• Message m_{k+1} is generated by node u. In this case, any node $z \in U/\{u\}$ should receive both m_1 and m_{k+1}. Because the lemma is correct when $n = k$, we know that $R_z(m_1) \prec R_z(m_k)$. Besides, we have proved the lemma is correct when $n = 2$ and we are given $m_k \rightarrow m_{k+1}$, so we know that $R_z(m_k) \prec R_z(m_{k+1})$. Hence, $R_z(m_1) \prec R_z(m_{k+1})$.

• Message m_{k+1} is generated by another node v. In this case, any node $z \in U/\{u, v\}$ should receive both m_1 and m_{k+1}. Since the lemma is correct when $n = 2$ and we are given $m_k \rightarrow m_{k+1}$, we can get that for any member $z \in U/\{u, v\}$, $R_z(m_1) \prec R_z(m_k) \prec R_z(m_{k+1})$. Thus, in this case, the lemma is correct.

2. m_1 and m_k are generated by u and v, respectively.

• Message m_{k+1} is generated by node u. In this case, any node $z \in U/\{u\}$ should receive both m_1 and m_{k+1}. Since the lemma is correct when $n = 2$ and we are given $m_k \rightarrow m_{k+1}$, we can get that for any node $z' \in U/\{u, v\}$, $R_{z'}(m_1) \prec R_{z'}(m_k) \prec R_{z'}(m_{k+1})$ is true. Thus, we just need to consider the node v. We know that $S_u(m_1) \prec S_u(m_{k+1})$, because m_1 and m_{k+1} are sent by the same node and they have the *precede* relation. In our method, there is only one path for two node to communicate each other. Therefore, for node v, $R_v(m_1) \prec R_v(m_{k+1})$ is also true.Hence, for any node $z \in U/\{u\}$, $R_z(m_1) \prec R_z(m_{k+1})$ is correct.

• Message m_{k+1} is generated by node v. In this case, any node $z \in U/\{u, v\}$ should receive both m_1 and m_{k+1}. Since the lemma is correct when $n = 2$ and we are given $m_k \rightarrow m_{k+1}$, we can get that for any node $z \in U/\{u, v\}$, $R_z(m_1) \prec R_z(m_k) \prec R_z(m_{k+1})$.

• Message m_{k+1} is generated by node w that is different from u and v. In this case, any node $z \in U/\{u, w\}$ should receive both m_1 and m_{k+1}. Since the lemma is correct when $n = 2$ and we are given $m_k \rightarrow m_{k+1}$, we can get that for any node $z^{'} \in U/\{u, v, w\}$, $R_{z'}(m_1) \prec R_{z'}(m_k) \prec R_{z'}(m_{k+1})$ is true. Thus, we just need to consider the node v. Because m_1 and m_k are sent by different nodes and they have the *precede* relation, we can know that $R_v(m_1) \prec S_v(m_k)$. As we are given $m_k \rightarrow m_{k+1}$, we know that $R_w(m_k) \prec S_w(m_{k+1})$. What's more, we know that $S_v(m_k) \prec R_w(m_k)$ and $S_w(m_{k+1}) \prec R_v(m_{k+1})$. Thus, we can deduce that for v, $R_v(m_1) \prec R_v(m_{k+1})$ is true. Therefore, for any node $z \in U/\{u, w\}$, $R_z(m_1) \prec R_z(m_{k+1})$ is correct. ■

Theorem 1: The proposed method preserves the causal ordering for group communication.

Proof: In order to preserve the causal ordering, the method should ensure that the messages that have the *precede* relation should be delivered to the recipients in the same correct order. According to definition 2, there are three kinds of *precede* relations. According to the three lemmas, we know that each kind of *precede* relations is ensured by the method. Hence, our method guarantees the causal ordering of messages. ■

VI. SIMULATIONS

In this section, we compare our MST based two-phase method (MSTTP) with the Core Based Tree (CBT) based method. CBT [14] method is a simple but well-known method for group communication routing. In our compared CBT based method, it first finds a CBT for the group communication and then assign slots for the tree by using the slot assignment algorithm proposed in this paper. In the method, the node which lies at the center of the group members is chosen as the core node. Then, we find a shortest path tree, which is rooted from the core node and spans all the group members, as the CBT. Messages from each group member are first sent to the core node through the shortest path from the member to the core node. Then, the core node propagates the messages further to other group members through corresponding shortest paths. The CBT based method can guarantee the causal ordering, since all the messages should arrive at the core node.

Simulations are conducted to assess the effectiveness of our method by considering two performance metrics. The first one is the bandwidth consumption of the group communication, i.e. the number of transmission slots assigned for the request. The second one is the success rate of the call, which equals to the number of successful communication requests over the number of total requests. We vary two parameters over a wide range: the number of members in a group (i.e. group size) and the network load. In this paper, the network load is defined as the average percentage of occupied slots in all nodes in the network [15]. In the simulations, we randomly place 100 nodes and 4 primary users in a $1000 * 1000$ 2-D free-space. Each of the primary users randomly chooses a data channel to use, which will be unusable for the nodes that are in the corresponding interference range. The radius of transmission range and interference range for all nodes are fixed to be 250 and 500, respectively. The total number of data channels is set to 6, and the number of timeslots in a frame set to 20. For the simulations, we perform 1000 times tests to evaluate the success rate of the two methods, and pick the mean value of results produced by 100 successful requests as the result of the number of transmission slots.

Fig.5 and Fig.6 show the number of transmission slots and the success rate vs the group size, where the network load is fixed at 0.3. From Fig.5, we can see that MSTTP method requires less transmission slots than CBT under the same condition. This is because in MST based method, we always find the tree which has the minimal tree links as the routing tree. Less tree links will bring to less bandwidth consumption. From Fig.6, we can find that the success rate of CBT is much less than the MSTTP method under the same condition. There are two reasons. One is that MSTTP method requires less number of transmission slots than CBT. The other is that the CBT method has a core node which is the performance bottleneck. When the group size is larger, the core node needs a large number of transmission slots to propagate messages for group members, which makes the call unsuccessful.

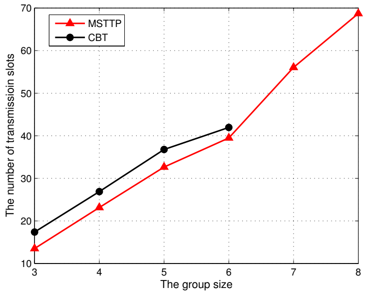

Fig. 5. The number of transmission slots versus the group size.

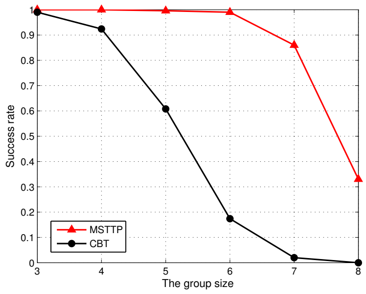

Fig. 6. Success rate versus the group size.

Fig.7 and Fig.8 show the number of transmission slots and the success rate vs the group size, respectively, where the group size is fixed at 5. From Fig.7, we can see that the number of transmission slots required by MSTTP method is less than by CBT. However, the number of transmission slots do not change with the increase of the network load. This is because the group size is fixed. In Fig.8, we can find that the success rate of CBT is much less than the MSTTP method under the same condition. The reason is that when the network load it lager, the available slots is less. Thus, it's harder to assign enough transmission to the core node which is the bottleneck. When the network load is 0.5, the success rate by using CBT method is 0.07 while it is 0.86 by using MSTTP method.

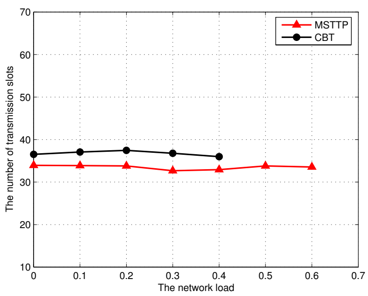

Fig. 7. The number of transmission slots versus the network load.

VII. CONCLUSION

In this paper, we investigate how to preserve the causal ordering of messages for group communication without a large amount of communication overhead nor a long latency for delivering message in CR networks. We considered the problem at the network layer. Our goal is to make a group communication connection so that the causal ordering is preserved and the bandwidth consumption of the communication is minimized. We proposed a MST based two-phase method which involves routing and transmission scheduling to solve it. Simulations have been conducted to compare our method with the CBT based method, and the results have shown that our method performs better than CBT in reducing the bandwidth consumption of the group communication. Our method also has higher success rate than CBT method, since there is no bottleneck node in our method.

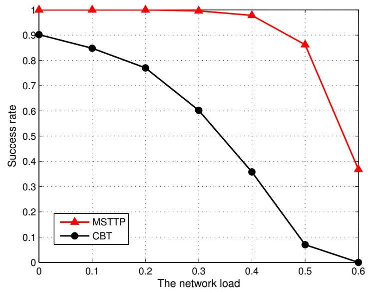

Fig. 8. Success rate versus the network load.

REFERENCES

[1] I. F. Akyildiz, W.-Y. Lee, M. C. Vuran, S. Mohanty, "NeXt generation/dynamic spectrum access/cognitive radio wireless networks: A survey", *Computer Networks*, Vol. 50, No. 13, 2006, pp.2127-2159.

[2] V.C. Gregory, K. Idit, V. Roman, "Group communication specification: a comprehesive study", *ACM Comptuing Surveys*, Vol. 33, No. 4, Dec. 2001, pp. 427-469.

[3] X. Jia, L. Wang, "A group multicast routing algorithm by using multiple minimum Steiner trees", *Computer Communications(Elsevier)*, Vol. 20, No. 9, Sep. 1997, pp.750-758.

[4] H. Liu, P.-J. Wan, X. Jia, "Fault-Tolerant Relay Node Placement in Wireless Sensor Networks, *Lecture Notes in Computer Science*, Vol. 3595, 2005, pp.230-239.

[5] X. Jia, "A total ordering multicast protocol using propagation trees", *IEEE transactions on parallel and distributed systems*, Vol. 6, No. 6, June 1995, pp. 617-627.

[6] K. Birman, T. A. Joseph, "Reliable communication in the presence of failures", *ACM Transactions on Computer Systems (TOCS)*, Vol. 5, No. 1, Feb. 1987, pp. 47-76.

[7] A. Schiper, J. Eggli, A. Sandoz, "A new algorithm to implement causal ordering", *Proceedings of the Third International Workshop on Distributed Algorithms*, Sep. 1989, pp.219-232.

[8] R. Prakash, M. Raynal, M. Singhal, "An efficient causal ordering algorithm for mobile computing environments", *Proceedings of 16th International Conference on Distributed Computing Systems*, 1996.

[9] W. Cai, B.-S. Lee, J. Zhou, "Causal order delivery in a multicast environment: an improved algorithm", *Journal of Parallel and Distributed Computing*, Vol. 62, No. 1, Jan. 2002, pp. 111-131.

[10] D. Dolev, S. Kramer, D.Malki, "Early delivery totally ordered multicast in asynchronous environments", *Proceeding of the 23th International Symposium on Fault-Tolerant Computing*, Jun 1993, pp. 544 - 553.

[11] R. Hincapie, J. Tang, G. Xue and R. Bustamante, "QoS routing in wireless mesh networks with cognitive radios", *Proceedings of IEEE Globecom*, Dec. 2008, pp. 1-5.

[12] J. Jia, J. Zhang, Q. Zhang, "Relay-assisted routing in cognitive radio networks", *Proceedings of IEEE ICC*, 2009, pp. 1-5.

[13] L. Xie, X. Jia, "QoS Multicast Routing and Transmission Scheduling in Multi-hop Cognitive Radio Networks", *Proceedings of Globecom 2010 Workshop on Pervasive Group Communication*, 2010, pp. 1487-1491.

[14] T. Ballardie, P. Francis, J. Crowcroft, "Core based tree", *Proceedings of ACM SIGCOMM' 93*, Sail Francisco, CA, Oct. 1993, pp. 85-95..

[15] H. Wu, X. Jia, "QoS multicast routing by using multiple paths/trees in wireless ad hoc networks", *Ad Hoc Networks(Elsevier)*, Vol. 5, No. 5, July 2007, pp.600-612.