

# Causal Order Protocol Based on Virtual Synchronous Group Membership in Wireless Sensor Networks

Chayoung Kim and Jinho Ahn<sup>1</sup>

*Dept. of Comp. Scie., Kyonggi Univ., Iuidong, Yeongtong, Suwon 443-760  
Gyeonggi, Republic of Korea  
{kimcha0,jhahn}@kgu.ac.kr*

## Abstract

*Many distributed applications require cooperating to fuse their data in those WSNs in real time. Some of these WSNs allow interacting with publishers and subscribers (P/S) sensors. But, there exist little research works on development of causal ordering based on virtual synchronous group membership of these P/S systems, one of mandatory properties for this kind of application fields. In this paper, we propose a novel approach based on gossip protocols and virtual synchronous group membership, guaranteeing the causal message ordering in those WSNs. In the proposed protocol, a broker gossips about FLUSH including the whole set of vectors to subscribers for virtual synchronous group membership. The protocol does not end based on the acknowledgement of FLUSH, but on the maximum gossip rounds of gossip disseminations. Therefore, this protocol is appropriate for sensor networks configured with active and passive sensors in real time synchronization. Its features might result in very low cost communication overhead between brokers and subscribers with flexible consistency by cooperating to fuse their data. Simulation results show that our gossip-style dissemination protocol outperforms the previous one in terms of virtual synchronous group membership.*

**Keywords:** Publish/Subscribe, Virtual Synchronous Group Membership, Reliability, Gossip protocols, Causal Order

## 1. Introduction

In Wireless Sensor Networks (WSNs) configured with passive and active sensors, passive sensors detect naturally radiated energy and active sensors supply, or send out, their own electromagnetic energy. Robotics problems typically require a much higher sensing rate, longer range, and high spatial resolution that cannot be delivered through common conventional means [4]. In robotics applications in WSNs, we can acquire three-dimensional (3-D) information [10]. The use of 3-D information is for detecting and avoiding obstacles in a 3-D workspace, recognizing objects, and mapping environments [10]. So, we need more intelligent means in a 3-D environment in WSNs [11]. For example, a 3-D camera of robotics provides an excellent example of both passive and active WSNs [10]. During a bright sunny day in passive mode, enough sunlight is illuminating the targets and then reflecting toward the camera lens, that the camera simply records the radiation. On a cloudy day or inside a room in active mode, there is often not enough sunlight for the camera to record the targets adequately. Instead, it uses its own energy source to illuminate the targets and record the radiation reflected from them. For another example, a WSN-based conference room configured with active and passive sensors is able to capture the 3D position and pose of users, and enable users to interact with digital media and contents shown on immersive displays. Passive sensors often fail in non-textured, featureless portions of a scene [4]. On the other hand, active sensors are more accurate in these regions and tend to be noisy in highly textured regions. So, the way to synergistically combine the two sensor modes to create a state-of-the-art depth sensing system runs in near real time [11].

---

<sup>1</sup> Corresponding author: Tel.:+82-31-249-9674; Fax:+82-31-249-9673.

These applications require cooperating to fuse their data in WSNs in real time [5]. For significant reliability of fusing sensor data in real time, some WSNs allow a decoupling between senders and receivers to interact with publishers (brokers) and subscribers (P/S) sensors. The sensors designed as brokers, which are fixed and reliable, might aggregate the information of the results based on the subscribers' interests. The receivers designed as subscribers, which can mobile and sense data in real time, receive the results aggregated by their brokers by gossip-style dissemination protocols [7].

Our proposed protocol can be applied to these WANs applications. Also, we consider data fusion based on virtual synchronous group membership [9] for significant reliability of fusing data in real time. For example, there is a mobile sensor for monitoring a moving object in these WSNs. When the sensor can communicate other sensors, it can find new sensor communication groups at that time. The proposed protocol considers data fusion based on virtual synchronous group communication members which can be mobile and join and leave freely. Also, the proposed protocol guarantees a causal ordering [8] based on gossip protocols and virtual synchronous group membership [9] in P/S systems in these WSNs. Gossip protocols have turned out to be adequate for large scale settings by achieving a "high degree of reliability" and strong message delivery ordering guarantees offered by deterministic approaches [2, 3]. Causal ordering is very useful for these distributed applications in which a large number of sensor nodes require cooperating to fuse their data in real time. So, the information of the results by fusing wireless sensed data of an object and monitoring an environment can be delivered to other brokers and subscribers in these WSNs with high degree of reliability [6, 11].

Therefore, we propose a novel approach based on gossip protocols and virtual synchronous group membership, guaranteeing causal ordering in these WSNs. In gossip protocols, when a process sends a multicast message, it randomly selects a small subset of members at random, called gossip targets or fan-out, which relates reliability of gossip protocols [2, 3]. The seminal probabilistic broadcast (pbcast) algorithm of Birman, *et al.*, [2] is originally described as a broadcast presented in the system based on global group membership and Eugster's algorithm (lpbcast) [3] is implemented for P/S systems as a broadcast. Causal ordering ensures that message  $m_1$  is delivered before  $m_2$ , if the logical timestamp of  $m_1$  happened before  $m_2$ . The virtual synchronous group communication [9] ensures that all messages delivered to some processes in a given membership view are delivered to all the correct processes in that view before a new view is installed, guaranteeing that all processes that install two consecutive views deliver totally ordered messages between these views. On the other hand, our proposed protocol based on virtual synchronous group membership guarantees that all processes that install two consecutive views deliver causally related messages.

In this protocol, because every broker knows every other brokers, it manages a vector per group, that represents its knowledge for each member of the group, of the number of message multicast by this member within this group like in the protocol of Birman, *et al.*, [1]. In the protocol [1], each member has to manage a vector per group and the whole set of these vectors should be piggybacked on each message. On the other hand, in our proposed protocol, every broker gossips about the multicast message including the whole set of vectors based on virtual synchronous group membership in P/S systems. When a broker synchronizes the group membership, it sends FLUSH including the whole set of vectors to other brokers. Between brokers, the virtual synchronous membership ends when the acknowledgments (ACKs) of FLUSH are received from all brokers, as same as a member in the protocol of Birman, *et al.*, [1]. On the other hand, between brokers and subscribers, a broker gossips about FLUSH including the whole set of vectors to subscribers, but the virtual synchronous membership ends when the maximum number of gossip rounds (MGRs) is reached based on gossip protocols. That is, the end of virtual synchronous group membership is not based on ACKs of FLUSH, but based on MGRs of gossip dissemination protocols.

Therefore, this protocol is appropriate for sensor networks configured with active and passive sensors in real time synchronization because virtual synchrony is not based on ACKs of all members, which might incur network congestion or explosion. Its features might be highly suitable for the area of the applications requiring only the small causal information overhead based on virtual synchronous group membership with flexible consistency by cooperating to fuse their data.

## 2. The Proposed Protocol

### 2-1. Algorithm Description

The proposed protocol respects deadline-constraints causal order using  $2DV_i$  ( $2DV$  stands for 2-Dimensional Vector) between brokers, like in the protocols of Rodrigues, *et al.*, [8]. As an example, in Figures 1 and 2,  $2DV_i$  describes the causal past of the message each broker  $i$  sends and receives for causal order and  $2DV_i[x,y] \leftarrow \text{send\_time}_m$  describes broker  $i$  knows the last message sent by broker  $x$  to broker  $y$  has been sent at time  $\text{send\_time}$ .

Figures 3 and 4 show the proposed protocol respecting causal order based on virtual synchronous group communications. In this proposed protocol, when broker  $i$  sends a message, it updates the entries of the array  $2DV_i[i,j]$  corresponding to all the destination brokers. Let us first a simple description of the protocol shown in Figure 3 (lines MB1-MB2).  $\forall (i,j)$ ,  $2DV_i[i,j]$  is stored in  $2DV_m$  and the multicast message  $m$  with  $2DV_m$  is sent for all destination brokers. So, we use  $m \rightarrow m' \Leftrightarrow 2DV_m < 2DV_{m'}$  according to this approach, like in the protocol Rodrigues, *et al.*, [8].

This example of Figure 1 shows how each broker = {A, B, C, D} participating in BrokerGroup1 = {A, B, C} and BrokerGroup2 = {A, B, D} instances FLUSH implementations between brokers. In Figure 1, broker A crashes. Broker B multicasts a FLUSH message including each group member list and 2-D vector for causal ordering, as a coordinator. For example, for BrokerGroup1, FLUSH is  $\langle \{B,C\}, (1,1,0,*) \rangle$  and for BrokerGroup2, FLUSH is  $\langle \{B,D\}, (1,1,*,0) \rangle$ . Brokers C and D acknowledge the FLUSH with a FLUSH-OK and request sending a lost message,  $A_1$ . The protocol is terminated by the coordinator which sends a VIEW message as soon as a FLUSH-OK is received from every broker. For example, for BrokerGroup1, VIEW is {B,C} and for BrokerGroup2, VIEW is {B,D}.

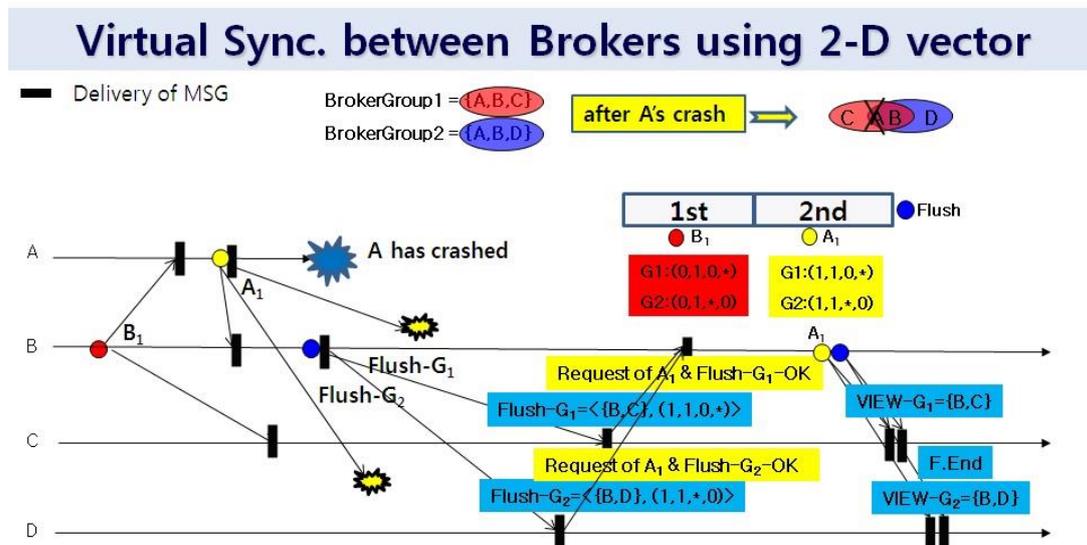
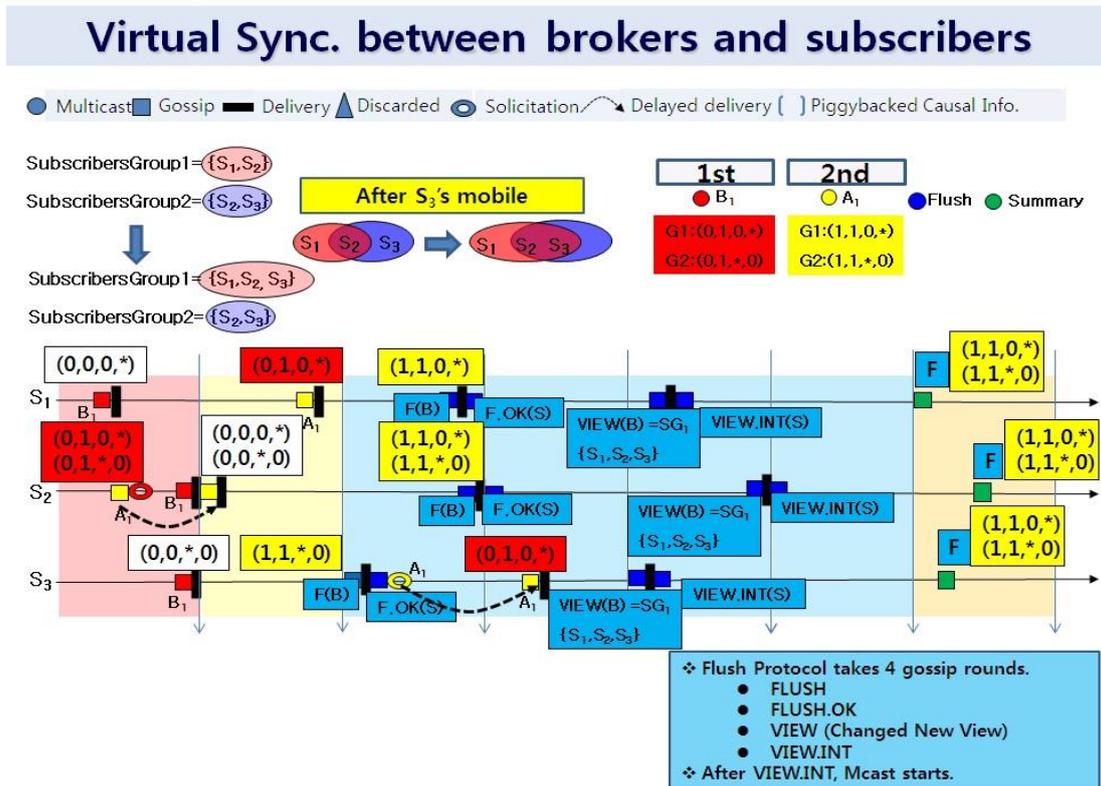


Figure 1. Virtual Synchronous Group Membership between Brokers

The example of Figure 2 shows FLUSH messages transmitted between brokers and subscribers based on gossip disseminations. In Figure 2, subscriber S3 is mobile. After that, SubscriberGroup1 is {S1, S2, S3} and SubscriberGroup2 is {S2, S3}. Between brokers, FLUSH is terminated when the coordinating broker receives FLUSH-OK from every broker. But, in the proposed protocol, between brokers and subscribers, FLUSH is processed based on gossip rounds. The protocol is terminated in 4 gossip rounds because it uses FLUSH, FLUSH-OK, VIEW, VIEW.INT. Each of them is proceeding based on the deadline of the gossip round that the system designer decides. Every broker gossips a FLUSH message including 2-D vector for causal ordering. For example, for SubscriberGroup1, FLUSH is including  $\langle(1,1,0,*)\rangle$  and for SubscriberGroup2, FLUSH is including  $\langle(1,1,*,0)\rangle$ . Because S2 is in both SubscriberGroup1 and SubscriberGroup2, it receives  $\langle(1,1,0,*)\rangle$ ,  $\langle(1,1,*,0)\rangle$ . Every subscriber gossips about FLUSH-OK to acknowledge FLUSH and solicits a lost message. In this figure 3, S3 solicits A1. The protocol is terminated not by the coordinator, but based on gossip dissemination protocols. Every broker gossips about VIEW as soon as a FLUSH-OK is received from every subscriber. But, the termination of the protocol does not depend on FLUSH-OK of all subscribers. After that, for SubscriberGroup1, VIEW is {S1, S2, S3} and for SubscriberGroup2, VIEW is {S2, S3}.



**Figure 2. An Example of Virtual Synchronous Communication Between Brokers**

In this proposed protocol, when a broker sends and receives multicast messages, there are two cases, i) the messages might be delivered to the application layer or ii) the messages might have to be discarded. In the proposed protocol shown in Figure 3, if the delivery of  $m$ , which is sent by  $j$  and received by  $i$ , violates causal order, then it should be discarded. Let us a description of the protocol shown in Figure 4 (lines RMB2) for 1) delivering a message. If the predicate  $Del\_OK \equiv ((2DV_{i[j,i]} < 2DV_{m[j,i]}) \wedge (\forall i \neq j : (2DV_{m[j,i]} \leq 2DV_{i[j,i]}))$  is true, that means its delivery does not violate causal order, then  $m$  can be delivered, like in the protocol Rodrigues, *et al.*, [8]. Let us a description of the

protocol shown in figure 4 (lines RMB1) for 2) discarding a message. If the predicate Del-Violation  $\equiv (2DV_m[j,i] \leq 2DV_i[j,i])$  is true, then  $m$  is discarded.

```

Procedure multicast( $m, 2DV_m, Bkers_m$ ) from brokers to brokers
  (MB1)  $send\_time_m \leftarrow current\_time$ ;
  (MB2)  $\forall j \in Bkers_m$  do  $2DV_i[i,j] \leftarrow send\_time_m$  od;
  (MB3) let  $2DV_m = 2DV_i$ ;
  (MB4)  $\forall j \in Bkers_m$  do  $send(m, 2DV_m, Bkers_m)$  od;

Procedure multicast( $m, 2DV_m, Subs_m$ ) from brokers to subscribers
  (MS1)  $send\_round_m \leftarrow current\_round$ ;
  (MS2)  $\forall j \in Subs_m$  do  $2DV_i[i,j] \leftarrow send\_round_m$  od;
  (MS3) let  $2DV_m = 2DV_i$ ;
  (MS4)  $\forall s \in Subs_m$  do  $send(m, 2DV_m, Subs_m)$  od;

Procedure gossip( $m, Partial\_Bkers_m$ ) from brokers to brokers
  (GB1) select partial brokers( $Partial\_Bkers_m$ ) from local_view;
  (GB2)  $\forall j \in Partial\_Bkers_m$  do gossip( $m, summarize(2DV_i)$ ) od;

Procedure gossip( $m, Partial\_Subs_m$ ) from brokers to subscribers
  (GS1) select partial subscribers( $Partial\_Subs_m$ ) from local_view;
  (GS2)  $\forall j \in Partial\_Subs_m$  do gossip( $m, summarize(2DV_i)$ ) od;

let each Deadline of FLUSH, FLUSH-OK, VIEW, VIEW-INT based on gossip disseminations.

Procedure flush(FLUSH,  $2DV_i$ ) between brokers and subscribers
  (FBS1) select partial brokers( $Partial\_Bkers_m$ ) from local_view;
  (FBS2) select partial subscribers( $Partial\_Subs_m$ ) from local_view;
  (FBS3)  $\forall j \in Partial\_Subs_m$  do gossip(FLUSH,  $summarize(2DV_i)$ ) od;
  (FBS4)  $\forall j \in Partial\_Bkers_m$  do gossip(FLUSH-OK) od;
  (FBS5) if Deadline of FLUSH-OK becomes then
     $\forall j \in Partial\_Subs_m$  do gossip (VIEW) od;
  (FBS6)  $\forall j \in Partial\_Bkers_m$  do gossip(VIEW-INT) od;

```

**Figure 3. The Proposed Protocol**

In the proposed protocol of Figure 3, there are the procedures for sending a multicast message from brokers to brokers and subscribers. In the procedure of **multicast** for brokers, **current\_time** is set as **send\_time<sub>m</sub>** and assigned to **2DV<sub>i</sub>[i,j]** for a causal ordering. In the procedure of **multicast** for subscribers, **current\_round** is set as **send\_round<sub>m</sub>** and assigned to **2DV<sub>i</sub>[i,j]** for a causal ordering. In the procedure of **gossip** for brokers and subscribers, a broker periodically gossips about the summary of a causal ordering to randomly selected gossip target, brokers (**Partial\_Bkers<sub>m</sub>**) and subscribers (**Partial\_Subs<sub>m</sub>**). In the proposed protocol of Figure 4, there are the procedures for receiving a multicast message from brokers and subscribers. In the procedure of **multicast** for brokers and subscribers, after delivering multicast messages satisfied with the delivery condition, they compare **2DV<sub>i</sub>[x,y]** and **2DV<sub>m</sub>[x,y]** and the bigger is assigned to **2DV<sub>i</sub>[x,y]**. In the procedure

of **gossip** for brokers and subscribers, on receiving the summary of a causal ordering, they compare  $2DV_i[x,y]$  and  $2DV_m[x,y]$  and request all message not to be received to the latest gossip sender.

```

When one of brokers,  $P_i$  receives multicast( $m, 2DV_m$ )
  let Del-Violation  $\equiv (2DV_m[j,i] \leq 2DV_i[j,i]);$ 
  let Del_OK  $\equiv ((2DV_m[j,i] > 2DV_i[j,i]) \wedge (\forall x \neq j : (2DV_m[x,i] \leq 2DV_i[x,i]));$ 
  (RMB1) if Del_Violation then discard( $m$ )
  (RMB2) else wait Del_OK Delivery Condition: DC( $m$ )
  (RMB3)  $\forall (x,y) : 2DV_i[x,y] \leftarrow \max(2DV_i[x,y] \leq 2DV_m[x,y]);$ 
  (RMB4) Delivery of  $m$  to  $P_i$ 
  (RMB5) endif

When one of subscribers,  $S_i$  receives multicast( $m, 2DV_m$ )
  let Del-Violation  $\equiv (2DV_m[j,i] \leq 2DV_i[j,i])$  or send_round $_m$  out of MAX_ROUND;
  let Del_OK  $\equiv ((2DV_m[j,i] > 2DV_i[j,i]) \wedge (\forall x \neq j : (2DV_m[x,i] \leq 2DV_i[x,i]))$ 
    or send_round $_m$  in MAX_ROUND;
  (RGS1) if Del_Violation then discard( $m$ )
  (RGS2) else wait Del_OK Delivery Condition: DC( $m$ )
  (RGS3)  $\forall (x,y) : 2DV_i[x,y] \leftarrow \max(2DV_i[x,y] \leq 2DV_m[x,y]);$ 
  (RGS4) Delivery of  $m$  to  $S_i$ 
  (RGS5) endif

```

**Figure 4. The Proposed Protocol (Continued)**

## 2-2. Correctness Proof

### 1) Causality is Never Violated (Safety)

**Proof:** Consider the delivery of a subscriber  $s_j$  that receives two messages  $m_1$  and  $m_2$  such that  $m_1 \rightarrow m_2$  in a given view.

Case 1.  $m_1$  and  $m_2$  are both transmitted by the same process  $B_i$  in a given view.

When we assumed gossip based group communication system, hence both  $m_1$  and  $m_2$  are gossiped to a subscriber  $s_j$  and  $2DV(m_1) < 2DV(m_2)$ . Under the protocol,  $m_2$  can only be delivered after  $m_1$  has been delivered.

Case 2.  $m_1$  and  $m_2$  are transmitted by two distinct processes  $B_i$  and  $B_i'$  in a given view.

By induction on the messages received by a subscriber  $s_j$  that  $m_2$  cannot be delivered before  $m_1$  in a given view. Assume that  $m_1$  has not been delivered and that  $s_j$  has received  $k$  messages. Since  $m_1 \rightarrow m_2$ , hence  $2DV(m_1) < 2DV(m_2)$ , we have  $2DV_{m_1}[i] \leq 2DV_{m_2}[i]$  in a given view. -----(1)

Base step: The first message delivered by  $s_j$  cannot be  $m_2$ . Recall that if no messages have been delivered to  $s_j$ ,  $2DV_{s_j}[i]=0$ . However,  $2DV_{m_1}[i]>0$  (because  $m_1$  is sent by  $B_i$ ), hence  $2DV_{m_2}[i]>0$  in a given view. By the protocol,  $m_2$  cannot be delivered by  $s_j$ .

Inductive step: Suppose  $s_j$  has received  $k$  messages, none of which is a message  $m$  such that  $m_1 \rightarrow m$  in a given view. If  $m_1$  has not yet been delivered, then  $2DV_{s_j}[i] < 2DV_{m_1}[i]$  in a given view. -----(2) Because the only way to assign a value to  $2DV_{s_j}[i]$  greater than  $2DV_{m_1}[i]$  is to deliver a message from  $B_i$  that was sent subsequent to  $m_1$ , and such a message would be causally dependent on  $m_1$  in a given view. Also, it starts to assign a

value to  $2DV_{sj}[i]$  greater than  $2DV_{mi}[i]$  after a new view. From relations (1) and (2) it follows that by the rules of our proposed protocol, The  $k+1$ st message delivered by  $s_j$  cannot be  $m_2$ , remember that  $m_2$  is sent by  $B_i$ .

## 2) Liveness

**Proof:** Suppose there exists a broadcast message  $m$  sent by process  $B_i$  that can never be delivered to a subscriber  $s_j$ . The rule implies that either  $\exists k$  s.t.  $2DV_m[k] \neq 2DV_{sj}[k]+1$  for  $k=i$ , or  $2DV_m[k] > 2DV_{sj}[k]$ , for  $k \neq i$ . We consider these cases in turn.  $2DV_m[i] \neq 2DV_{sj}[i]+1$ . That is,  $m$  is not the next message to be delivered from  $B_i$  to  $s_j$ . Notice that only a finite number of messages can precede  $m$ . Since all message are gossiped about to all processes and gossip communication channels are redundant and managed based on the maximum gossip round, it follows that there must be some message  $m'$  sent by  $B_i$  that  $s_j$  received previously, has not yet delivered, and this is the next message from  $B_i$ , *i.e.*,  $2DV_m[k] = 2DV_{sj}[k]+1$ . If  $m'$  is also delayed, it must be under the other case.  $\exists k \neq i$ , s.t.  $2DV_m[k] > 2DV_{sj}[k]$ . Let  $n=2DV_m[k]$ . The  $n$ th transmission of process  $p_k$ , must be some message  $m' \rightarrow m$  that has either not been received at  $s_j$ , or was received and is delayed. Under the hypothesis that all messages are sent to all processes,  $m'$  was already gossiped about to and periodically sent to  $s_j$ . Since the communication system eventually ends all messages based on the maximum gossip round, we may assume that  $m'$  has been received by  $p_j$ . The same reasoning that was applied to  $m$  can now be applied to  $m'$ . The number of messages that must be delivered before  $m$  is finite and  $>$  is acyclic, hence this leads to a contradiction.

## 4. Performance Evaluation

We compare average throughput of our proposed causal order protocol based on virtual synchronous group membership [1] using gossip dissemination protocols between brokers and subscribers with that of a previous one [9] based on virtual synchronous group membership [1] using traditional reliable group communications. We consider a system composed of a finite set of brokers (publishers) that communicate by message passing. There is also a set of subscribers. In this comparison, we rely on a set of parameters referred to Bimodal Multicast [2] and LPBCast [3] for gossiping parameters. Brokers can only fail by crashing (*i.e.*, we do not consider Byzantine failures). A broker that never fails is correct. For simplicity, we do not include process recovery in the model. We assume further that failures are independent. The probability of a message loss does not exceed a  $\varepsilon > 0$ ,  $\varepsilon=0.05$ . The number of broker crashes does not exceed  $f < n$ . The probability of a broker crash during a run is bounded by  $\tau = f / n$ ,  $\tau=0.01$ . At every round, each broker has an independent uniformly distributed random view of size  $l$  of subscribers like as [3]. These views called as uniform views are not constant, but continue evolving. The subset of subscribers in the uniform view to which a broker gossips a message, are chosen randomly according to a uniform distribution. And we assume that processes gossip in periodic rounds, the gossip round is constant and identical for each process and the maximum gossip round is  $\log N$ .

We are interested in how our proposed protocol, based on P/S systems with difference communication groups, from brokers and to brokers and from brokers and subscribers, has low cost overhead than a traditional one without distinguishing P/S ones like as [1, 9]. Specifically, the proposed one and the traditional one are compared to each other in terms of the stability of average throughput by showing how many subscribers receives messages within a deadline. Figure 5 shows the average throughput as a function of perturb rate for various group sizes. The x-axis is the group size and the y-axis is the number of messages processed in the perturb rate, (a)20%, (b)30%, (c)40% and (d)50%. In the four subfigures from 5(a) to 5(d), the average throughput of causally ordered delivery protocol based on virtual synchronous group communication and gossip

protocols from brokers to subscribers is not a rapid change. In perturbed networks with subscribers join and leave, traditional reliable causal order based on virtual synchronous group membership [1] are very expensive because events of sending and receiving messages are governed by all members without distinguishing between brokers and subscribers and ACK mechanism to ensure reliability. On the other hands, the proposed protocol based on P/S has less overhead because communications between brokers and subscribers are based on gossip-style disseminations, the information are managed only by brokers and the deadline of virtual synchronous group membership is not based on ACK mechanism, but based on the maximum gossip round. So, the management cost of virtual synchronous group membership is cheaper than that of the previous protocol [1].

We know which approach is more preferable depends on the user applications. In the network layers using some sort of ACK mechanism to ensure virtual synchronous group membership, such a use of more information for causal order is very expensive. On the other hand, in gossip-style dissemination networks, there is no ACK mechanism because members periodically gossip about the summary of received messages. Although gossip protocols might not use ACK mechanism, the protocols guarantee high reliability [2, 3]. So, our proposed protocol can use this periodic mechanism for causal order protocol based on virtual synchronous group membership without further mechanisms. So, we argue that our gossip-style dissemination protocol outperforms the previous one in terms of virtual synchronous group membership. Therefore, the protocol may be more attractive for mobility of subscribers because the message overhead resulting from the mobility depends on the number of brokers and brokers are more stable than subscribers.

## 5. Related Works

There are researches based on the P(publish)/S(subscribe) paradigm in the area of sensor network communications to approach the problem of querying sensors from mobile nodes [7], [17]. Directed Diffusion [7] can be seen as publish-subscribe mechanism, which is implemented using the tree-based architecture rooted at the publisher. SENSTRACT [17] is mapping from queries to topics and the corresponding underlying sensor network structure. SENSTRACT [17] is a tree-based P/S system structured by service providers as roots, representing one of the data-centric routing protocols for data dissemination of sensor networks. Cross Reality is about connecting "location-specific 3D animated constructs" in virtual worlds to in-building sensors [8].

Early work in gossip-style protocol, Birman et al. [3] proposes bimodal multicast thanks to its two phases: a "classic" best-effort multicast such as IP-Multicast is used for the first rough dissemination of messages. The second phase assures reliability with a certain probability by using gossip-based retransmissions. But, Lpbcast [4] proposes gossip-style broadcast mechanisms based on a local view instead of a global view. Lpbcast [4] is a completely decentralized protocol because of no dedicated brokers for membership management. To ensure causal message ordering, [9] proposes a novel causal ordering abstraction that takes messages deadlines into consideration for distributed soft real-time applications. In deadline-constrained causal order, each message has its own deadline and, if it arrives on time, never misses its deadline due to preceding messages

To ensure causal message ordering, [9] proposes a novel causal ordering abstraction that takes messages deadlines into consideration for distributed soft real-time applications. In deadline-constrained causal order, each message has its own deadline and, if it arrives on time, never misses its deadline due to preceding messages.

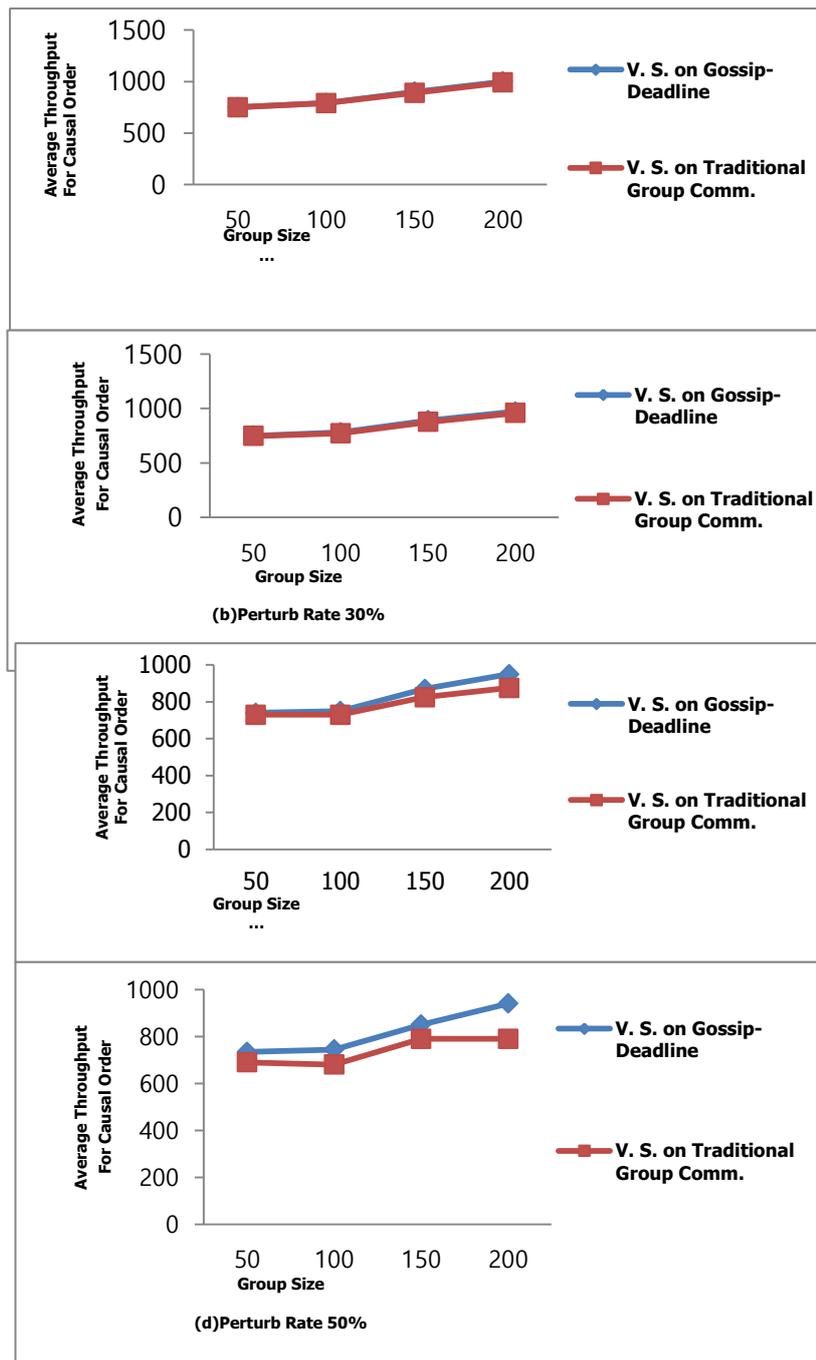


Figure 5. The Average Throughput as a Function of Perturb Rate

## 6. Conclusions

This paper proposes a novel approach based on gossip protocols and virtual synchronous group membership, guaranteeing causal ordering in wireless sensor networks (WSNs) configured with active and passive sensors. The way to synergistically combine the two sensor modes to create a state-of-the-art depth sensing system runs in near real time. In the proposed protocol, because every broker knows every other broker, it manages a vector per group, which represents its knowledge for each member of the group. When brokers synchronize about the group membership, it sends FLUSH including the whole set of vectors to other brokers. The virtual synchronous membership

ends when acknowledgments of FLUSH are received from all brokers. On the other hand, between brokers and subscribers, every broker gossips about FLUSH including the whole set of vectors to subscribers. However, the virtual synchronous membership ends when the maximum gossip rounds (MGRs) is reached based on gossip protocols. Simulation results show that our gossip-style dissemination protocol outperforms the previous one in terms of virtual synchronous group membership. Therefore, we believe the protocol is more attractive for mobility of subscribers because the message overhead resulting from the mobility depends on the number of brokers and brokers are more stable than subscribers.

## Acknowledgement

This paper is a revised and expanded version of a paper entitled “Virtual Synchronous Group Membership-based Causal Order Protocol for Wireless Sensor Networks” presented at Int’l Conference on Advanced Software Engineering & Its Applications, held at HNA Grand Hotel Mingguang Haikou, Hainan China, in December 20-23, 2014.

## References

- [1]. K. Birman, A. Schiper and P. Stephenson, “Lightweight Causal and Atomic Group Multicast”, ACM Transactions on Computer Systems, vol. 9, no. 3, (1991), pp. 272-314.
- [2]. K. Birman, M. Hayden, O. Ozkasap, Z. Xiao, M. Budiu, and Y. Minsky, “Bimodal Multicast”, ACM Transactions on Computer Systems, vol. 17, no. 2, (1991), pp. 41-88.
- [3]. P. Eugster, R. Guerraoui, S. Handurukande, P. Kouznetsov, and A.-M. Kermarrec, “Lightweight probabilistic broadcast”, ACM Transactions on Computer Systems, vol. 21, no. 4, (2003), pp. 341-374.
- [4]. M. Hebert, “Active and Passive Range Sensing for Robotics” In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00), (2000) April, pp. 102-110, San Francisco, CA.
- [5]. C. Intanagonwiwat, R. Govindan and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (MobiCOM '00), (2000) August, pp. 56-67, Boston, MA.
- [6]. J. Lipton, M. Laibowitz, D. Harry, N.-W. Gong, M. Mittal and A. Paradiso, “Metaphor and Manifestation Cross-Reality with Ubiquitous Sensor/Actuator Networks”, IEEE Pervasive Computing, vol. 8, no. 3, (2009), pp. 24-33.
- [7]. S. Pleisch and K. Birman, “SENSTRAC: Scalable Querying of SENSOR Networks from Mobile Platforms Using TRACKING-Style Queries”, International Journal of Sensor Networks, vol. 3, Issue 4, (2008), pp. 266-280.
- [8]. L. Rodrigues, R. Baldoni, E. Anceaume and M. Raynal, “Deadline-Constrained Causal Order”, 3rd IEEE International Symposium on Object-oriented Real-time distributed Computing, (2000).
- [9]. L. Rodrigues, K. Guo, A. Sargento, R. van Renesse, B. Grade, P. Ver´issimo and K. Birman, “A transparent light-weight group service”, In Proceedings of the 15th IEEE Symposium on Reliable Distributed Systems, (2009) October, pp. 130-139, Niagara-on-the-Lake.
- [10]. Q. Yang, K.-H. Tan, B. Culbertson and J. Apostolopoulos, “Fusion of active and passive sensors for fast 3d capture”, 2010 IEEE International Workshop on Multimedia Signal Processing (MMSP), (2010) October, pp. 69-74.
- [11]. J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey”, Computer Networks, vol. 52, Issue 22, (2008) August, pp. 2292-2330.

## Authors



**Chayoung Kim** received B.S. and M.S. degrees from the Sookmyung Women's University, Seoul, Korea, in 1996 and 1998, respectively and Ph.D. degree from the Korea University in 2006. From 2005 to 2008, she was a senior researcher in Korea Institute of Science and Technology Information, Korea, where she has been engaged in National e-Science of Supercomputing Center. From 2009 to 2012, she was a researcher at Contents Convergence Software Research Center

in Kyonggi University, Korea. Since 2012, she has been an adjunct professor in Department of Computer Science, Kyonggi University, Korea. Her research interests include distributed computing, group communications and peer-to-peer computing.



**Jinho Ahn**, received his B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Korea University, Korea, in 1997, 1999 and 2003, respectively. Since 2003, he has been a professor in Department of Computer Science, Kyonggi University. He has published more than 80 papers in refereed journals and conference proceedings and served as program or organizing committee member or session chair in several domestic/international conferences and editor-in-chief of journal of Korean Institute of Information Technology and editorial board member of journal of Korean Society for Internet Information. His research interests include distributed computing, fault-tolerance, sensor networks and mobile agent systems.

