

Gossip-Pull System-based Dissemination Protocol Satisfying Message Causality Condition

Jin-Ho Ahn, Cha-Young Kim
Department C.S., Kyonggi Univ., South Korea
e-mail: jinhoahn@hanmail.net

Abstract—This paper introduces a group communication protocol ensuring message causality condition based on gossip-pull systems. The message causality condition can be satisfied by making all multicast messages transferred to group recipients through redundant message sending by sensor brokers, especially based on gossip-pull systems, which are only used when recipients request their messages to sensor brokers. These advantageous aspects enable the proposed protocol to enormously lower the redundancy of gossip messages compared with the gossip-push systems.

Keywords—distributed algorithms; gossip-pull; message redundancy; message causality

I. BEGINNING

There are large number of important applications in wireless sensor networks (WSN) [1,2]. The proposed protocol in this paper can be applied to above mentioned applications in wireless sensor nodes, deployed in an unstructured WSN [3]. Our proposed protocol is suitable to unstructured WSN applications like as a temporal protocol [4] based on logical time not physical time with ensuring that message m_1 is delivered before m_2 , if m_1 happened before m_2 , such as causally ordered delivery. In [4], the protocol is based on gossip protocols for the management policy distribution. So, our proposed protocol ensures the causally ordered delivery based on gossip protocol, especially gossip-pull systems. For reliable communication for this sensor networks, services such as congestion control, acknowledgements, and packet-loss recovery are necessary to guarantee reliable packet delivery and delivery order consistency [11]. Also, there is no causal order guaranteeing multicast protocol supporting many distributed and unstructured WSN applications with a variety of collaboration features, such as chat windows, white boards, p2p video and other media streams, based on publisher(sensor-broker) in the previously developed protocols. Especially, these above mentioned applications need a variety of collaboration features and coordination mechanisms [11].

Fan-out and randomness relates reliability and robustness of gossip-based protocols [2,3,4]. This paper introduces a group communication protocol for overlapping multicast groups ensuring message causality condition in an unstructured WSN. Our proposed protocol uses gossip-pull systems, which are only used when subscribers request their messages to sensor brokers. Also, the message causality condition can be satisfied by making all multicast messages transferred to overlapping subscribers using gossip-based

protocols by sensor brokers, especially based on gossip-pull systems, which are only used when subscribers request their messages to sensor brokers. These advantageous aspects enable the proposed protocol to enormously lower the redundancy of gossip messages compared with the gossip-push systems and be highly scalable for the area of the applications requiring only the minimum causal information. So, our proposed protocol uses a vector per group.

II. NEW MULTICAST PROTOCOL

A. Informal Descriptions

In this proposed protocol, sensor brokers selected from all of the sensor nodes which are based on time-stamped gossip-pull protocols [6]. In gossip-pull protocol, if some subscribers in a grid lose their interests in information published on a particular topic, the brokers representing the sensor grid update properly leaving subscribers to acquire significantly less redundancy.

On the other hand, in the proposed protocol, every broker disseminates the multicast message piggybacked with the digest information of vectors using gossip-pull systems.

P/S some sensors designed as brokers might aggregate the information of the results based on the subscribers' interests, while guaranteeing the causally ordered delivery of messages. The subscribers receive the results aggregated by their brokers by gossip-pull protocols. Recently, research on P/S in wireless sensor networks [5] has mainly focused on mobile subscribers and publishers relying on a fixed broker infrastructure to support them. We can see that each sensor broker representing a sensor grid and subscribers requesting messages marked in blue in figure 1. Proposed protocol needs a vector, which size is a number of brokers because of only one color per sensor broker. Therefore, this protocol is appropriate for sensor networks based on sensor brokers because every sensor broker knows every other sensor brokers.

Our proposed protocol uses gossip-pull systems. So, there is no subscriber to discard them and it acquires significantly less redundancy. In the figure 2, there are VT clocks for each group, Group1 and Group2. For each message generated by a member i , each $VTs[i]$ is incremented by 1. Figures 2 shows an example of the protocol based on sensor brokers with causal ordering VT clocks in what order is $B1 \rightarrow A1 \rightarrow B2 \rightarrow B3$. This example in figure 3 illustrates $Group1 = \{A, B, C\}$, $Group2 = \{A, B, D\}$, $Subscribers = \{S1, S2, S3\}$.

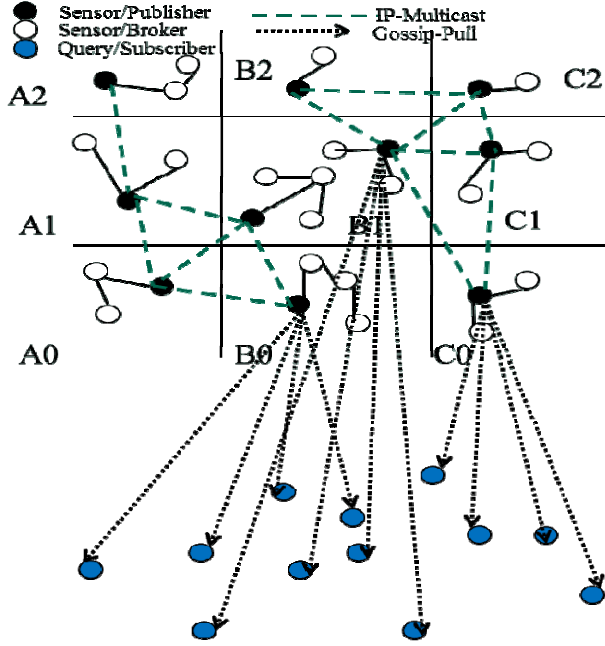


Figure 1. Publishers vs. subscribers in gossip-pull systems.

Subscriber S1 receives all messages from Broker A and all messages are delivered in accordance with VT clocks piggybacked by multicast messages. Subscriber S2 requests message B1 to the latest gossip-sender after receiving message A1 because it knows that there is one message, which is not received by comparing piggybacked information about causally ordered delivery. Subscriber S3 does the exactly same thing as S2 did because it knows that there is one message, which is not received after all of messages, A1, B2 and B3.

B. Algorithmic explanations

Procedure SEND_MULTICAST updates **Overlapped_Multicast_Vectors** by incrementing by one and sends the multicast message piggybacked with all vectors of all groups **Overlapped_Multicast_Vectors**. In **SEND_GOSSIP** procedure, a broker digests all vectors of all groups **Overlapped_Multicast_Vectors** and gossips about them to randomly selected subscribers requesting messages from its local view. In **RECEIVE_MULTICAST** procedure, a broker receive the multicast message from other broker, it compares it's all vectors of all groups with the all vectors piggybacked on the multicast message **Overlapped_Multicast_Vectors(msg) ≤ id.Overlapped_Multicast_Vectors** to verify causally related all message. In **RECEIVE_GOSSIP** procedure, a subscriber requests some messages not to be received to the latest gossip sender by comparing it's all vectors of all groups with the piggybacked all vectors of all groups on the multicast message **Overlapped_Multicast_Vectors(msg) ≤ id.Overlapped_Multicast_Vectors**.

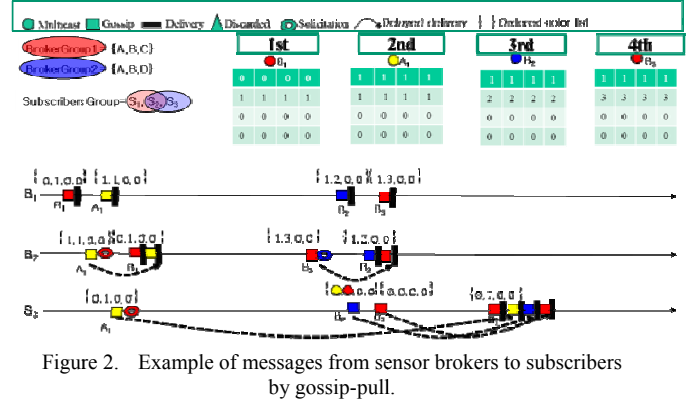


Figure 2. Example of messages from sensor brokers to subscribers by gossip-pull.

// **Select gossip-target from Publishers**

Procedure SEND_MULTICAST

// **Sending Seq.=Sending Seq.+1**

for All_Interest_Groups do
increment **Overlapped_Multicast_Vectors(id)** by one ;

// **M piggybacked with Vectors**

m = (**All_Interest_Groups**,
Overlapped_Multicast_Vectors) ;
Unreliable_Multicast(m) ;

// **Select gossip-target from Subscribers**

Procedure SEND_GOSSIP

// **Randomly Selecting Small Set**

for each **p** in **All_Subscribers**, requesting **m**
with probability rate
send **m** to **p** ;
do **Garbage_Collection** ;

// **From Publishers to Publishers**

Procedure RECEIVE_MULTICAST

if **m** not in **Waiting_List** then
put **m** into **Waiting_List** ;
call **CAUSAL_DELIVERY** ;

// **From Publishers to Subscribers**

Procedure RECEIVE_GOSSIP

// **Check the Vector Digest for identifying whether MSG is not received**

If **m** is not in **Waiting_List** or not delivered then
send solicit_Retransmission(**r**) to
Latest_Gossip_Sender ;
call **CAUSAL_DELIVERY** ;

Procedure CAUSAL_DELIVERY

delivery = TRUE ;

// **Check VT of MSG for causal order**

for All_Interest_Groups do
if(**Overlapped_Multicast_Vectors** >
id.Overlapped_Multicast_Vectors) then
delivery = FALSE ;
break ;

// **If causal order is satisfied**

if(delivery = TRUE) then
remove **m** from **Waiting_List** ;
deliver **m** to APPLICATION ;
MAX(Overlapped_Multicast_Vectors) ;

Figure 3. Algorithmic description.

III. CONCLUSIONS

This paper introduces a dissemination protocol for redundant groups satisfying message causality condition based on gossip-pull systems. The message causality condition can be satisfied by making all multicast messages transferred to group recipients through redundant message sending by sensor brokers, especially based on gossip-pull systems, which are only used when recipients request their messages to sensor brokers. These advantageous aspects enable the proposed protocol to enormously lower the redundancy of gossip messages compared with the gossip-push systems and be highly scalable for the area of the applications requiring only the minimum causal information.

FUNDING SUPPORT

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(Project No.: 2012R1A1A2044660).

REFERENCES

- [1] A. Attarwala, D. Jagdish, and U. Fischer, "Realtime collaborative video annotation using GAE and XMPP", *Cloud Computing (CLOUD)*, 738 – 739, 2011.
- [2] C. Esposito, D. Controneo, and S. Russo, "On reliability in publish/subscribe services", *Computer Networks*, Vol(57), Issue(5), April 2013, 1318–1343.
- [3] D. Eysers, T. Freudenreich, A. Margara, S. Frischbier, P. Pietzuch, and P. Eugster. "Living in the present: on-the-fly information processing in scalable web architectures" In *CloudCP*, 2012.
- [4] Margara, A. Frischbier, S. T. Freudenreich, Eugster, P. Eysers, D. and Pietzuch. P. "ASIA: Application-specific integrated aggregation for publish/subscribe systems. Technical report, 2012.<http://www.cs.otago.ac.nz/staffpriv/dme/asia/ASIA2011.pdf>.
- [5] K. Mitchell, et al. 2010. "Social TV: Toward Content Navigation Using Social Awareness", In *Proceedings of EuroITV2010*, 283--291.
- [6] 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.