Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/092L26

Router Design for DDS: Architecture and Performance Evaluation

Kyu-haeng Lee[1], *Tae Wan Noh*[1], *Kyeong Tae Kim*[2], *Won-tae Kim*[2] *and Chong-kwon Kim*[1,*]

[1] Dept. of CSE, Seoul National University, 151-744, Seoul, Korea
[2] Embedded SW Research Department, Electronics Telecommunications Research Institute, 161 Gajeong-dong, Yuseong-gu, Daejeon, 305-700, Korea

Received: 5 Jun. 2014, Revised: 4 Aug. 2014, Accepted: 6 Aug. 2014

Abstract: The Data Distribution Service (DDS) for real-time systems is a middleware standard for Publish/Subscribe (Pub/Sub) which is an abstraction for one-to-many communication. While DDS is known for supporting many real-time applications successfully, when applications communicate with others in different network domains over a Wide Area Network (WAN), DDS could face severe performance degradation, due to the fact that most ISPs do not allow IP-multicast and UDP flows in WAN. In this paper, we propose a router design for DDS. Keeping DDS semantics, the proposed DDS router provides efficient data distribution over WAN based on two schemes: overlay multicast and local recovery. Through NS-3 simulations, we show that the proposed scheme improves system performance significantly in terms of latency and RTT, compared to the current DDS mechanism.

Keywords: DDS,CPS, multicast, overlay

1 Introduction

Recently, Cyber Physical Systems (CPSs) have been spreading around the world due to the growing needs of real-time applications (such as environment monitoring, air traffic control and management, financial trading, and military communications). DDS is an Object Management Group (OMG) standard for Publish and Subscribe (Pub/Sub), which is an abstraction for one-to-many anonymous, decoupled and asynchronous communication between a publisher and its subscribers [1].

Many researchers have highlighted DDS as a promising approach for next generation CPSs, as it could meet the strict requirements of real-time applications. Due to the nature of data-centric communication, DDS applications generate huge broadcast and multicast traffic while exchanging meta-information and distributing data. Thus, it is very important to handle both types of traffic appropriately in DDS.

Until now, most DDS applications have only been targeted at single, small-scale Local Area Network (LAN) domains, where handling broadcast or multicast traffic does not result in any serious issues. However, it has been predicted that in the near future, applications will need to communicate with other applications in different network domains over a Wide Area Network (WAN), and in such a network DDS may not work well due to the following reasons. First, even though there are plentiful IP-multicast algorithms [2,3,4,5], most of them require routers to support a specific multicast functionality. Unfortunately, most ISPs do not allow multicast due to reasons such as profit issues, management complexity, and deployment issues. Second, ISPs also do not allow UDP flows; so most outbound traffic flows should be forwarded by TCP ports, which means there could be numerous TCP connections for broadcast traffic. It is well understood that TCP is not appropriate for most real-time applications in general [6].

In this paper, we propose a router design for DDS which provides efficient data distribution over WAN. In particular, the proposed DDS router boosts system performance by using two main schemes, overlay multicast and local recovery, while keeping DDS semantics. The overlay multicast scheme works between DDS routers and overcomes the limitation of IP-multicast. Further, to realize local recovery, we propose a virtual subscriber and an aggregate ACKNACK. Both schemes reduce the number of

---

[*] Corresponding author e-mail: ckim@snu.ac.kr

messages which are transmitted over WAN, compared with legacy DDS, and therefore lower the latency and prevent the waste of network resources.

To evaluate the performance of the proposed DDS router, we implement it on the NS-3 simulator. Through the simulations, we show that the proposed scheme improves system performance significantly in terms of latency and RTT, compared with the current DDS mechanism.

We summarize our contributions in this paper as follows. First, we propose a DDS router design, which provides efficient data distribution over a WAN based on overlay multicast and local recovery, while keeping DDS semantics. Second, through the NS-3 simulations, we show that the system with a DDS router outperforms the legacy DDS-based system.

The remainder of the paper is organized as follows. We give the background of this paper in Section 2 and in Section 3 we provide our motivation. We next describe in detail the DDS router design in Section 4. Section 5 shows the performance evaluation via NS-3 simulations. Section 6 provides the related work and we finally conclude our paper in Section 7.

2 Preliminaries

In this section, we briefly review how DDS works.

2.1 Topic

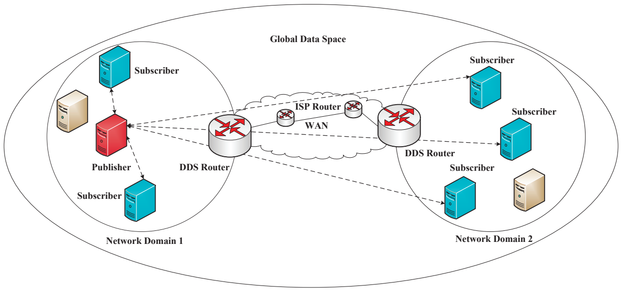


Fig. 1: Network model. The current version of DDS cannot support efficient transmission over WAN because (1) IP multicast has not been deployed yet due to a variety of factors such as cost of deployment and the need for pricing models, and (2) most ISPs do not allow UDP flows, and therefore outbound traffic flows should be forwarded by TCP ports.

DDS is based on the concept of a fully distributed Global Data Space (GDS), as shown in Fig. 1. Applications can join and leave the GDS at any time to write and read data autonomously and asynchronously. These applications and data are called publishers, subscribers, and 'topics' - the unit of information which can be produced and consumed, respectively. Further, a topic is defined as a triad composed of a type, a name, and QoS policies. In particular, each time a publisher sends instances of a particular topic, DDS distributes them to all subscribers who want to receive the topic. It all starts with the discovery process.

2.2 Discovery

Before the actual topic data transmission, all participants, publishers and subscribers, should exchange advertisement messages to express their intent for a specific topic. Then, subscriptions are dynamically matched by taking into account the exchanged topic information. For this purpose, the DDS defines a protocol called the Endpoint Discovery Protocol (EDP). In EDP, every participant periodically broadcasts its topic meta-information to the whole network. The EDP requires reliable communication, so after receiving an EDP message, a participant should send an ACKNACK message back to the participant that has sent the EDP message.

2.3 Multicast

A DDS publisher can be configured to send data on multicast [1]: a single message sent to a multicast address will be received by multiple nodes on the network. Though the multicast functionality can improve the efficiency of the DDS and related APIs are already specified in DDS, it has not been used in practice, because most routers are not capable of UDP/IP multicast algorithms, as mentioned earlier. Thus, the data should be unicast in the case of WAN transmission.

2.4 Reliability

22 QoS policies are used to control relevant properties of DDS entities [1]. One of the key policies is reliability: applications require their instances of topics to be delivered in order. ACKNACK and HEARTBEAT (HB) messages are used to support the reliability QoS. Upon receiving an HB from a publisher, subscribers should send ACKNACKs of previously received data back to the publisher, so that the publisher can retransmit some missing data to subscribers, if needed.

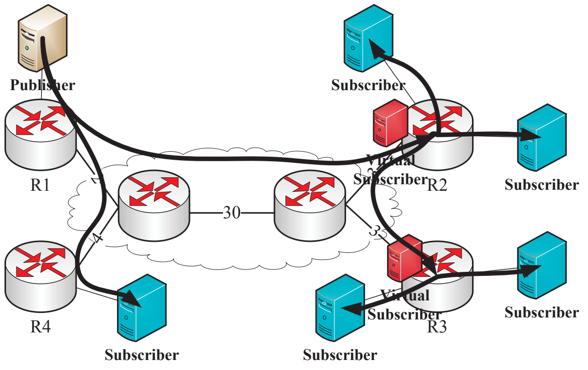
3 Motivation

The majority of traffic in DDSs is just right for being broadcast or multicast. However, in the case of WAN transmission, we should send traffic on TCP unicast because most ISPs do not allow UDP flows and IP-based multicast. Additionally, it is non-trivial for DDS applications to achieve the reliability QoS in such a network due to the following reasons. First, the ACKNACK implosion problem can occur [7,8]. When many subscribers simultaneously send their ACKNACKs to their publisher, the publisher cannot handle all the incoming messages due to the queue limit. Second, erroneous links may generate tremendous TCP flows for retransmission, which incurs high latency, and therefore hinders real-time communication.

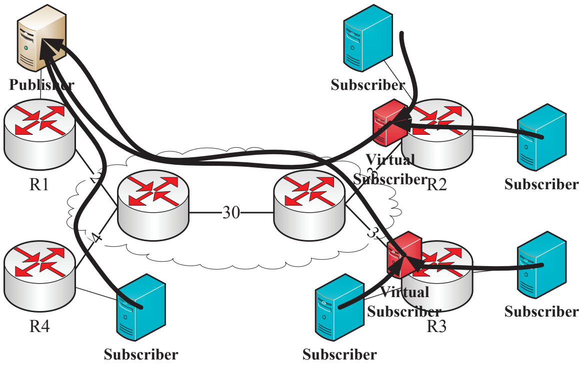
Example

Let P and S be the number of publishers and the number of remote subscribers for each publisher, respectively. Therefore, the total number of DDS participants is PS.

First, for the discovery, all participants should broadcast their topic information to each other. Thus, the number of required messages is $O(P^2S^2)$. Recall that these messages are transmitted through TCP. Second, in the data transmission phase, many copies of data could be generated because we cannot use any IP-level multicast algorithms. Especially in the case of copies being frequently transmitted over a high cost link (e.g., the link in the middle of Fig. 2), the overall latency may increase significantly. The number of transmitted messages is $O(PS)$. Further, if we need reliable communication, DDS participants may generate additional messages such as HB and ACKNACK for every topic instance.



(a) Overlay multicast



(b) Local recovery

Fig. 2: An example of a DDS router operation. The proposed DDS router can provide efficient data distribution over WAN by using two main mechanisms: overlay multicast (a) and local recovery (b).

4 DDS Router Design

4.1 Overview

As discussed above, the proposed DDS router can provide an efficient data distribution over WAN by using two main mechanisms: overlay multicast and local recovery. We illustrate Fig. 2 to show how the proposed schemes work.

First, the overlay multicast overcomes the limitations of IP-multicast in WAN. In particular, as shown in Fig. 2 (a), we regard each DDS router as an end-host. This is different from most overlay multicast proposals [9,10], because in our scheme, any actual end-level hosts which are behind routers (i.e., publisher or subscriber) need not do the overlay multicast. Instead, the overlay multicast is only adopted in DDS routers. After building a multicast tree, topic data are forwarded along the tree. In our example, R3 receives it from R2, not from R1. In this paper, we assume that a publisher or a subscriber is associated with only one topic, and overlay multicast trees are generated per topic.

Second, local recovery is performed by a virtual subscriber, as shown in Fig. 2 (b). The main idea is that after a virtual subscriber (in actuality, this is a router) receives ACKNACKs from local subscribers, it makes an aggregate ACKNACK and sends it back to the publisher. Further, it performs retransmission if needed. The local recovery not only provides fast retransmission but reduces the number of messages sent over WAN.

Now, we elaborate each scheme in more detail in the next section.

4.2 Overlay Multicast

4.2.1 Gathering link costs via EDP

To construct an overlay multicast tree, routers should gather the cost of the link (e.g., RTT) between them. The best way is to exploit Link State Advertisement (LSA) information in OSPF-based routing protocols which are widely used in current network systems [6]. From this, each router can estimate the entire network topology as well as the link cost. To implement this in real systems, however, the DDS needs to access network layer information, which may violate the conceptual model of the communication system and even ruin the legacy DDS semantics. Recall that the layer of the DDS lies between the application layer and the transport layer, not the network layer.

Instead, we use an approximate approach; we estimate link cost only between DDS routers during the EDP. Fortunately, DDS routers exchange messages periodically in the discovery phase, so it is very simple to measure the cost of each neighboring link between DDS routers. After measuring the cost, DDS routers exchange it with each other by appending it to EDP ACKNACK messages.

4.2.2 Tree construction

The part of constructing a tree is based on the well-known shortest path algorithms [11,12,13]. Recall that a multicast tree is generated per topic. Let N and E be a set of DDS routers and edges. Each link $(u,v) \in E$ is associated with cost c_{uv}, and p denotes the publisher of a topic. Then, by introducing a binary decision variable x_{uv} and a non-negative auxiliary variable f_{uv}, we can formulate an optimization problem as follows.

Objective:

$$\text{minimize} \sum_{(u,v)\in E} c_{uv} x_{uv} \qquad (1)$$

Constraints:

$$\sum_{u \neq v} x_{uv} = 1 \qquad \forall v \neq p \qquad (2)$$

$$f_{uv} \leq \begin{cases} (|N|-1)x_{uv} & \text{if } u = p \\ (|N|-2)x_{uv} & \text{otherwise} \end{cases} \qquad \forall u, \forall v \neq p \qquad (3)$$

$$\sum_{u \neq v} f_{uv} - \sum_{u \neq v} f_{vu} = 1 \qquad \forall v \neq p \qquad (4)$$

The main goal of this problem is to minimize the total cost of flows that traverse the overlay network, as shown in Eq. (1). Eq. (2) states that every overlay node v (except the publisher) has exactly one parent node. Eq. (3) and (4) show that the number of unit flows is limited by the number of arcs in the tree.

Unfortunately, this problem is known to be NP-hard [13]. Instead, we can employ an LP-relaxation or solve the dual program [12,14]. A detailed description of those techniques is out of the scope of this paper.

Since all DDS routers share the same link cost information, trees made by each DDS router would be same (i.e., a shared tree). After building the tree, the publishers DDS router sends topic data to its children DDS routers along the tree. Upon receiving the data, DDS routers broadcast it into their networks and forward it to their children routers.

4.3 Local Recovery

Local recovery, where retransmission is performed not by a source but by an intermittent node, enhances the performance of the reliable multicast. It provides fast retransmission and recovery, and therefore improves network efficiency. However, it is very difficult to realize the local recovery mechanism in DDS due to the following reasons.

First, all DDS participants, including DDS routers, should follow the DDS semantics during local recovery. Though tree-based architectures [7,8] have been known to provide the best performance, we cannot directly adopt it into our system because additional modifications are required to build such logical architectures. Second, since CPS nodes are basically low-power, low-computing sensors, they may suffer from resource limitations, such as memory and CPU limitations. Nodes taking charge of the recovery should keep the samples of previously sent data. In the worst case scenario, such nodes should maintain all previous data, which requires a significantly large buffer size.

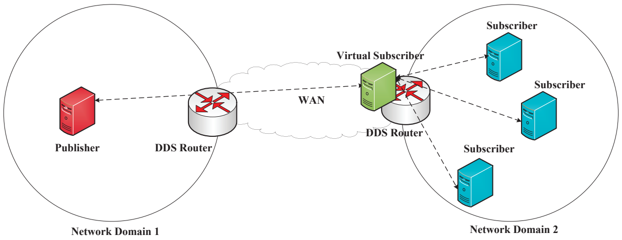


Fig. 3: Virtual subscriber. A virtual subscriber is a representative of certain subscribers. According to QoS settings and memory status, each DDS router can create more than one virtual subscriber.

To overcome the above issues, we introduce the concept of a virtual subscriber. A virtual subscriber is defined as a set of subscribers for the local recovery. DDS participants in other network domains do not need to know which subscribers are associated with a certain virtual subscriber. They cannot see the actual configuration of the virtual subscriber, as shown in Fig. 3.

4.3.1 Configuring a virtual subscriber

Basically subscribers are regarded as candidates to be grouped, if they have the same topic type and name. And each DDS router can make more than one virtual subscriber according to the QoS settings of the subscribers it has. We should carefully consider two QoS policies for configuring virtual subscribers: memory-related and timing-related QoS.

Memory-related QoS

Two QoS policies, ‘History’ and ‘ResourceLimits’, are related to system memory; the ‘History’ QoS policy specifies how much data must be stored by DDS for datawriter or datareader; the ‘ResourceLimits’ controls the amount of physical memory allocated for DDS entities. Assume that a DDS router with memory of R wants to make a virtual subscriber from S subscribers. If the required memory for each subscriber is K, then the following condition should be satisfied, $KS \leq R$.

Timing-related QoS

Another possible concern arises from the need to assign the QoS policies of a virtual subscriber. One easy and efficient way is to group subscribers which have the same QoS settings. In this case, we just set the QoS values of the virtual subscriber to those of its subscribers. However, in most cases subscribers have different QoS policies, even for the same type of topic. In this case, we take the strictest QoS value. For example, suppose 'Deadline' QoS values of three subscribers (Sub1, Sub2 and Sub3) are 0.5, 0.7 and 1, respectively. If we want to group all subscribers as one, then the 'Deadline' value will be 0.5. If we group Sub2 and Sub3 as one virtual subscriber, then the value will be set at 0.7.

4.3.2 Retransmission and aggregate ACKNACK

Basically, a virtual subscriber works exactly the same as a legacy subscriber: it broadcasts its topic information (type, name, and QoS), makes a connection with a publisher in the discovery phase, and waits to receive topic data from the publisher. One different behavior from the legacy subscriber is that the virtual subscriber sends an aggregate ACKNACK instead of the legacy ACKNACK, as shown in Fig. 2 (b). The aggregate ACKNACK represents the receiving status of all subscribers of a virtual subscriber. More specifically, after receiving data from a publisher, a DDS router first forwards it to the corresponding subscribers and waits for ACKNACKs from the subscribers. If the received ACKNACK comes from the subscriber which does not belong to any virtual subscriber, then the DDS router just forwards it to the publisher. Otherwise, the virtual subscriber starts retransmission until its subscribers receive all messages completely. After the recovery, it generates an ACKNACK and sends it back to the publisher. Since the virtual subscriber only sends an ACKNACK when all its subscribers have received the data correctly, the publisher does not need to perform the recovery for the virtual subscriber and its actual subscribers.

While the aggregate ACKNACK scheme reduces the number of WAN messages significantly, as expected, it may result in a slight increase of delay because it takes time to wait for responses from subscribers of a virtual subscriber. Thus, to minimize the effect of the delay, we propose that non-time-sensitive subscribers are grouped first as a virtual subscriber.

5 Performance Evaluation

5.1 Setting

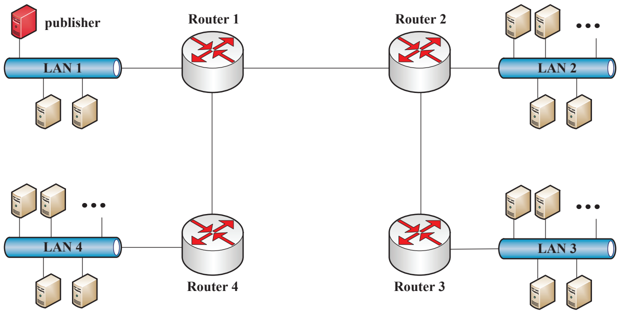


Fig. 4: Network topology used in the simulation. There are four network domains with routers, and one publisher in the network domain of Router1 publishes a topic to all other nodes.

We implement our DDS router in the NS-3 simulator platform to analyze and compare performance. Fig. 4 shows the network topology used in the simulation. We set up four network domains with routers and assume that one publisher in the network of Router1 publishes topic data to all other subscribers in the network. We evaluate the performance of our scheme by comparing it with two others as the following.

- Legacy: consists of ordinary routers with basic features.
- Model 1: exploits the overlay multicast only.
- Model 2: the one which we suggest exploits both overlay multicast and local recovery.

We measure the following three metrics and summarize the parameters used in the simulation in Table 1.

- Average latency: average time from the publisher sending a topic to the subscribers successfully receiving it.
- Average RTT: average time between the publisher sending a topic to subscribers and receiving ACKNACKs from all subscribers.
- The number of packets in WAN: the number of packets transmitted over WAN during the simulation.

5.2 Overall performance

5.2.1 Latency

In the first simulation, we vary the size of subscribers per network domain from 50 to 200 and we conduct a

Table 1: NS-3 simulation parameters

Parameter	Value
Link bandwidth and delay of LAN	1Gbps, 2ms
Link bandwidth and delay of WAN	100bps, 6560ns
Traffic type	CBR
Topic data size	1024 (B)
ACKNACK size	40 (B)
HeartBeat size	50 (B)
Topic data interval	1 (s)
Simulation duration	100 (s)
Topic data generation	from 1 to 100 (s)

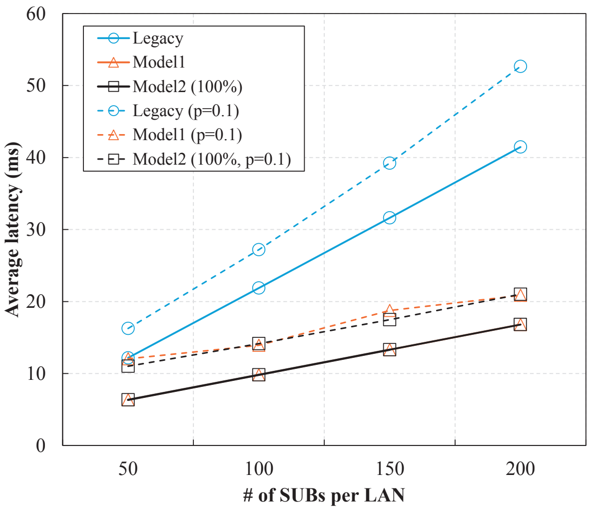


Fig. 5: Latency result according to the number of subscribers per network domains. The proposed system obtains the latency gain significantly via overlay multicast.

simulation with different link failure probabilities to evaluate how well local recovery of the DDS router can handle the link failure.

Fig. 5 shows the average latency result. First, we can observe that the latency increases with the number of subscribers. Additionally, the average latency in Model 1 and Model 2 decreases more than in Legacy thanks to the overlay multicast. When there is no error in the network, Model 1 and Model 2 show the same performance because retransmission does not happen in both cases. Recall that the only difference between both schemes is local recovery. In the case that link errors exist, latency increases. In the case of 200 LAN subscribers, latency increases by 11ms and 5ms for Legacy and Model1, 2, respectively. Most importantly, the latency of Model 2 has better performance than that of Model 1, because Model 2 provides a fast retransmission and recovery to subscribers.

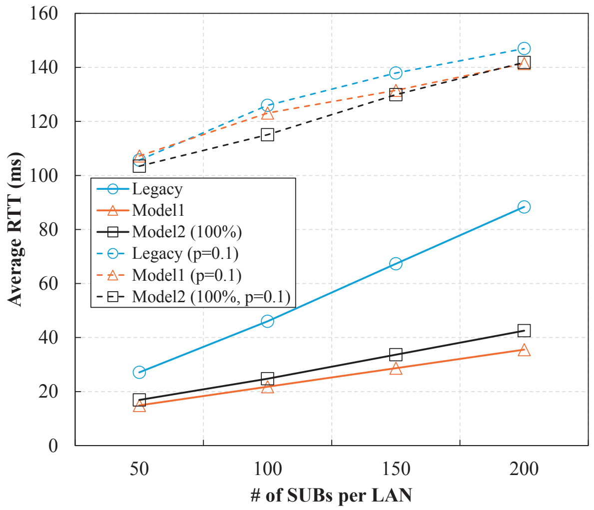


Fig. 6: RTT results according to the number of subscribers per network domains. Overlay multicast and local recovery in our proposed model can reduce RTT. There is a delay caused by aggregate ACKNACK, but its effects are negligible.

5.2.2 RTT

Next, we illustrate the average RTT result in Fig. 6. In the case of no link error, performances of the RTT are better in Model 1 and Model 2 than in Legacy. In particular, Model 1 shows better performance than Model 2, because the aggregate ACKNACK in Model 2 incurs a slight delay (maximum 4ms).

Furthermore, we can see that RTT is more influenced by link error, compared with latency. Model 1, where only the overlay multicast is used, shows better performance than Legacy, but it is worse than Model 2 (maximum 7ms). The dominant factor which determines RTT is the delay caused by the retransmitted packet. In Model 1, every packet associated with retransmission should be unicast through WAN and this causes a delay. On the other hand, in Model 2 the number of WAN unicast messages for retransmission can be reduced by executing local recovery. Although there is a delay caused by the aggregate ACKNACK, it has negligible effects.

5.2.3 The number of packets in WAN

Fig. 7 shows the measured results of the number of packets which passed WAN; this is the main cause for the increase in latency and RTT. Both an overlay multicast and a virtual subscriber can reduce the number of WAN messages. From the result, we can see that overlay multicast is more effective than virtual subscriber for reducing the number of WAN messages.

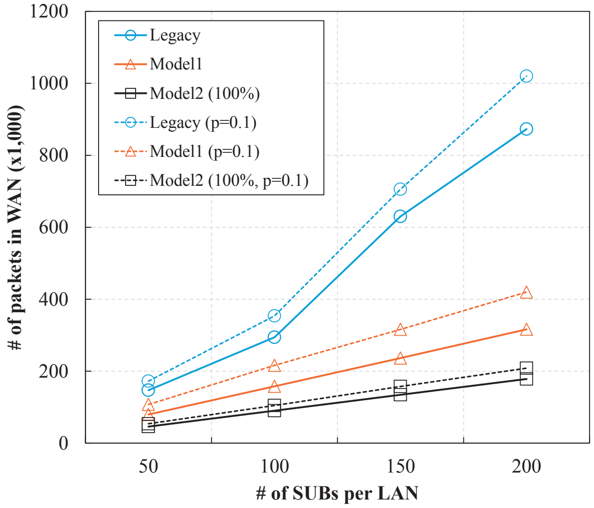


Fig. 7: The number of packets with respect to the different number of subscribers per network domain. Both overlay multicast and virtual subscriber can reduce the number of WAN messages. Notably, overlay multicast has more influence on reducing the number of messages.

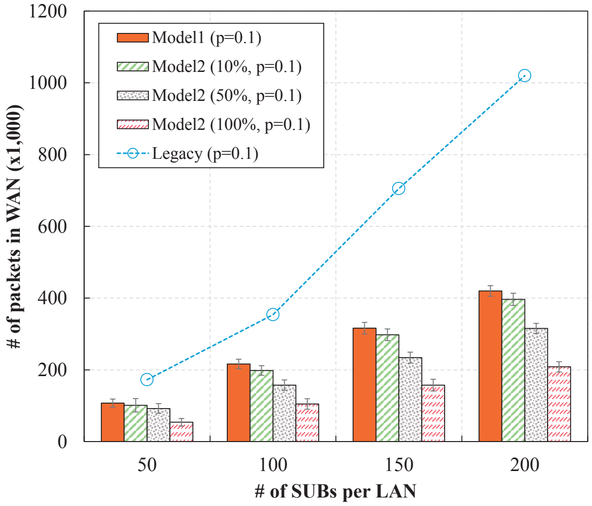


Fig. 9: The number of messages with respect to different virtual subscriber configurations. As the number of subscribers that a virtual subscriber covers increases, the number of messages to be sent over WAN decreases. This affects latency and RTT.

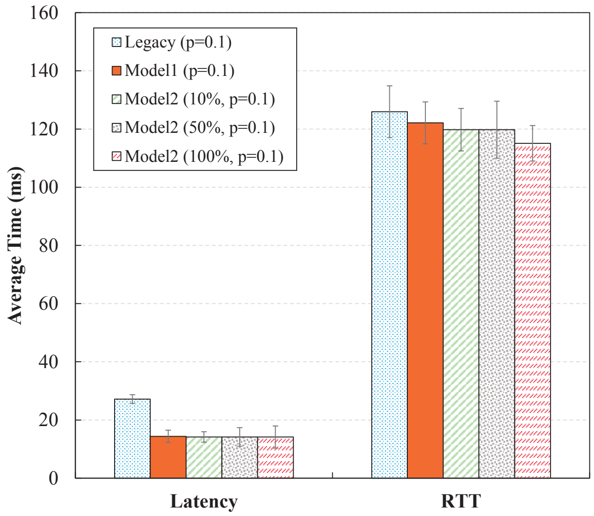


Fig. 8: Latency and RTT results according to different virtual subscriber configurations. The virtual subscriber has a greater effect on RTT than on latency.

5.3 Effect of virtual subscriber configuration

In this section we study the effects of the virtual subscriber configuration. First we fix the number of virtual subscribers per router as one, and vary the size of the subscribers which the virtual subscriber handles. For example, in the case of 50%, the virtual subscriber only covers half of the total subscribers. After that, we measure the same metrics with the previous set of simulations and illustrate the results in Fig. 8 and Fig. 9.

Fig. 8 shows latency performance when the number of LAN subscribers is 100 with a link error of 0.1. We can see that Model 1 and Model 2 are better than Legacy, but the configuration of the virtual subscribers does not affect the results too much (less than 2ms). Recall that latency is mostly affected by the overlay multicast. On the other hand, regarding RTT and the number of messages, related performance is improved as the number of subscribers of a virtual subscriber increases. Model 2 outperforms Model 1 and Legacy in terms of RTT and the number of messages. From the results, we demonstrate that the virtual subscriber scheme can increase system performance through efficient retransmission in the case of link errors.

6 Related Work

While the network industry is continuously pushing toward developing DDS [15,16,17,18,19,20], there is only a small amount of work on DDS routers and routing services. RTI [15] provides routing services as an option, but their work is far from solving network issues. Instead, they try to bridge legacy applications and other technologies.

Multicast is a reliable, well-investigated research area. One major issue is the scalability problem caused by message implosion. Tree-based protocols are known for their high efficiency [7,8], but the cost of managing a multicast group is too high. SRM [21] is robust at dealing with link failure and provides more scalable multicast than other protocols, but it may generate too many unnecessary messages. Since all aforementioned protocols need support from IP-multicast enabled routers, we cannot adopt them into our network environment.

Recently, overlay multicast protocols have received much interest, because they can overcome the limitation of topology-specific constraints (e.g., multicast-enabled routers) [9,10]. However, in order to employ the overlay multicast in current DDS, all DDS participants require many modifications they need to understand the overlay. In contrast, we simply apply overlay multicast to DDS routers, and thus the other DDS entities need not undergo any changes. DDS routers actually work as gateways, so they can understand the proposed overlay protocol.

7 Conclusion

In this paper, we propose a router design for DDS which provides efficient data distribution over WAN. The proposed DDS router boosts system performance by using overlay multicast and local recovery, while keeping the DDS semantics. Through NS-3 simulations, we show that the system with a DDS router obtains significant performance gain in terms of latency and RTT.

Acknowledgement

This work was supported by the IT R&D Program of MSIP/KEIT [10035708, “The Development of CPS (Cyber-Physical Systems) Core Technologies for High Confidential Autonomic Control Software”] and by the IT R&D program of MSIP/KEIT. [10041861, Development of Wired/Wireless i-AVB System Technology for Concurrent Transmission of HD Media and Control Data].

References

[1] 3GPP TR V11.0.0, “the Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol specification (DDSI), v2.1”, http://www.omg.org/spec/DDSI/2.1/PDF/formal-09-01-05.pdf, (2009).

[2] Speakman, T. and Crowcroft, J. and Gemmell, J. and Farinacci, D. and Lin, S. and Leshchiner, D. and Luby, M. and Montgomery, T. and Rizzo, L. and Tweedly, A. and Bhaskar, N. and Edmonstone, R. and Sumanasekera, R. and Vicisano, L., “PGM Reliable Transport Protocol Specification”, RFC 3208 (Experimental), Internet Engineering Task Force (IETF), (2001).

[3] Tony Ballardie, “Core Based Tree (CBT) Multicast - Architectural Overview and Specification”, (1995).

[4] Deering, Stephen E. and Cheriton, David R., “Multicast routing in datagram internetworks and extended LANs”, ACM Trans. Comput. Syst., ACM, (1990)

[5] Deering, S. and Estrin, D.L. and Farinacci, D. and Jacobson, V. and Ching-Gung Liu and Liming Wei, “The PIM architecture for wide-area multicast routing”, IEEE/ACM Transactions on Networking, (1996).

[6] Larry L. Peterson and Peter S. Davie, “Computer Networks”, Morgan Kaufmann, (2011).

[7] Paul, S. and Sabnani, K.K. and Lin, J.C.-H. and Bhattacharyya, Supratik, “Reliable multicast transport protocol (RMTP)”, Selected Areas in Communications, IEEE Journal on, (1997).

[8] Rajagopalan, Bala, “Reliability and scaling issues in multicast communication”, SIGCOMM Comput. Commun. Rev., ACM, (1992).

[9] Fahmy, Sonia and Minseok Kwon, “Characterizing Overlay Multicast Networks and Their Costs”, Networking, IEEE/ACM Transactions on, (2007).

[10] Sherlia Y. Shi and Jonathan S. Turner, “Routing in Overlay Multicast Networks”, IN PROCEEDINGS OF IEEE INFOCOM, (2002).

[11] Thulasiraman, Krishnaiyan and Swamy, Madisetti NS, “Graphs: theory and algorithms”, John Wiley & Sons., (2011).

[12] Xiao, Ying and Thulasiraman, Krishnaiyan and Xue, Guoliang and Jüttner, Alpár, “The constrained shortest path problem: algorithmic approaches and an algebraic study with generalization”, AKCE International Journal of Graphs and Combinatorics, 2, 2, 63-86, (2005).

[13] Wang, Z. and Crowcroft, J., “Quality-of-service routing for supporting multimedia applications”, Selected Areas in Communications, IEEE Journal on, 14, 7, 1228-1234, (1996).

[14] Boyd, Stephen P and Vandenberghe, Lieven, “Convex optimization”, Cambridge university press, (2004).

[15] Pardo-Castellote, G., “OMG Data-Distribution Service: architectural overview”, Distributed Computing Systems Workshops, 2003. Proceedings. 23rd International Conference on, 200-206, (2003).

[16] PrismTech, “OpenSplice DDS Development Guide, v 4.1”, (2009).

[17] Busch, D. “Introduction to OpenDDS”, (2012).

[18] Computing, T. O. “Learn About How it Works: Take the CoreDX DDS Tour Twin Oaks Computing, Inc.”, (2012).

[19] Lee, SooHyung and Kim, JaeHyuk and Kim, WonTae and Ryou, JaeCheol, “Communication Entities Discovery in Complex CPS System”, Control and Automation, and Energy System Engineering, Springer, 213–219, (2011).

[20] IBM Corp. Web. Rational Rhapsody Developer, “IBM Rational Rhapsody. version 7.5. 1.” (2010).

[21] Floyd, Sally and Jacobson, Van and Liu, Ching-Gung and McCanne, Steven and Zhang, Lixia, “A reliable multicast framework for light-weight sessions and application level framing”, IEEE/ACM Trans. Netw., (1997).

Kyu-haeng Lee received the B.S. degree in computer science and technology from Tsinghua University, Beijing, China, in 2009. He is currently working towards the Ph.D. degree at Seoul National University, Seoul, Korea. His research interests include MIMO/OFDM systems, IEEE 802.11, system optimization, and CPS.

Taewan Noh is a master student in the Department of Computer Science at Seoul National University, Seoul, Korea. He received his B.S. in Computer Science from Kwang-Woon University, Seoul, South Korea in 2013. His research interests include wireless LANs and Social Network.

Kyeong Tae Kim received the B.S. degree in Computer Engineering from Kangwon National University, Korea in 2004 and the M.S. degrees in Information and Communications from Gwangju Institute of Science and Technology, Korea in 2006, respectively. Since 2006, he has been a senior research member of Electronics and Telecommunications Research Institute (ETRI). His research interests include Network Security and Wireless Communication.

Won-Tae Kim received BS, MS and Ph.D from the department of Electronic Engineering, Hanyang University, 1994, 1996 and 2000 respectively. He established a venture company named Rostic Technologies, inc. at Jan. 2001 and worked as CTO from Jan. 2001 to Feb. 2005. He joined ETRI at March 2005 and Now he is the manager of CPS (Cyber-Physical Systems) research section in SW Contents Technology Research Lab. He is a president of CPS project group of TTA (Telecommunication Technology Association) from 2011. And he is a vice-president of Military SW Research Association in Korea Institute of Information scientists and engineers from 2011.

Chong-kwon Kim received the B.S. degree in industrial engineering from Seoul National University, the M.S. degree in operations research from Georgia Institute of Technology, and the Ph.D. degree in Computer Science from University of Illinois at Urbana-Champaign in 1981, 1982, and 1987, respectively. In 1987, he joined Bellcore as a Member of Technical Staff and worked on Broadband ISDN and ATM. Since 1991, he has been with Seoul National University as a Professor in the School of Computer Science and Engineering. His research interests include wireless and mobile networking, high speed network control, distributed processing, and performance evaluation.