Review for Event Delivering Techniques in Publish/Subscribe Scheme

Lajitha H.
PG Scholar
Department of Computer Science
College of Engineering, Perumon

Ratheesh S.
Assistant Professor
Department of Computer Science
College of Engineering, Perumon

Neevan R.
Assistant Professor
Department of Computer Science
College of Engineering, Attingal

ABSTRACT

Most of the challenges in the creation of a publish/subscribe scheme is an effective delivery of message in bounded time and reliability of transmission. Because of the communication in WAN may be affected by the uncertain behavior of the network, in which messages can be lost or delayed. To enforces both reliability and timeliness in a publish/subscribe scheme requires two different approaches: Network Coding and Gossiping. This paper presents the concepts associates with these approaches in the publish/subscribe environment. The objective of this paper is to analyze how these approaches works and what are the impacts after applying it.

General Terms

Publish/Subscribe scheme,Events

Keywords

Network Coding,Gossipping

1. INTRODUCTION

The publish/subscribe interaction scheme is messaging middleware which is claimed to provide the loosely coupled form of interaction required in such large scale settings like financial services, business intelligence and critical infrastructure [1].Publishers publish events, and subscribers subscribe to and receive the events they are interested in. A distributed publish/subscribe system for scalable information dissemination can be decomposed in three functional layers: namely the overlay infrastructure, the event routing and the algorithm for matching events against subscriptions. The attractive characterization of publish/subscribe is in the way notifications flow from senders to receivers. Receivers are not directly targeted from publisher but indirectly addressed according to the content of notifications. Subscriber expresses its interest by issuing subscriptions for specific notifications, independently from the publishers that produces them, and then it is asynchronously notified for all notifications, submitted by any publisher, that match their subscription. The strength of this event-based interaction style lies in the full decoupling in time, space, and synchronization between publishers and subscribers.

There are mainly three publish/subscribe scheme [2] Topic-based, Content-based and Type-based. Topic-based publish/subscribe systems introduce a programming abstraction which maps individual topics to distinct communication channels. Topics are strongly similar to the notion of groups and it enforces platform interoperability by relying only on strings as keys to divide the event space. The content-based publish/subscribe is a variant of topics scheme in which events are not classified according to some predefined external criterion, but according to the properties of the events themselves. Type-based publish/subscribe provide a natural description of content-based scheme through public members of the considered event type, while ensuring the encapsulation of these events.

Network coding is a paradigm for sending information over networks. Instead of simply relaying packets, network nodes can also combine incoming packets and send the resulting coded packets to outgoing edges. But these operations do not generate new information in the network. There are two main benefits of this approach: potential throughput improvements and a high degree of robustness. Benefits of network coding over mere routing in improving the throughput of a single-source multicast transmission, i.e., when the same data at a source is to be transmitted to multiple destinations in the network.Network coding lays the foundations for an efficient and fast recovery where redundancy is judiciously added to the dissemination or recovery operations.

Gossiping is a distributed retransmission protocol and is used to retrieve missing packets in case of incomplete information. In the gossip paradigm, an event is disseminated like the spread of a contagious disease or the diffusion of a rumor over unstructured overlay networks. The protocol exploits local knowledge of nodes about subscriptions made by their neighbor nodes, coupled with a gossip strategy. It can be employed quite effectively to build publish/subscribe systems on top of these nodes to easily manage the networks. The dissemination is based on pure local decisions; nodes employ a mixed strategy that combines gossip together with a local knowledge of subscriptions made their neighbors.The gossip approach has been applied to a variety of application domains, such as database replication, cooperative attack detection, and publish/subscribe-based data dissemination.Gossip algorithms achieve reliability in a probabilistic sense, by guaranteeing that all participants in the system receive any message only with a certain, quantifiable probability. Gossip algorithms reach high level of scalability at the price of a small loss in reliability.

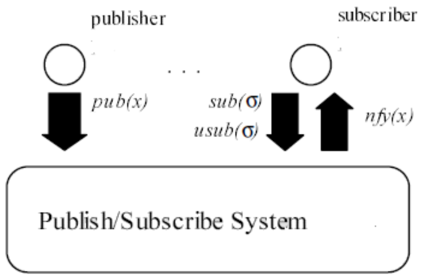

Fig. 1. General view of Publish/Subscribe System

2. LITERATURE REVIEW

2.1 The Publish/Subscribe Paradigm

A publish/subscribe communication system (PSS) can be represented by a triple $< \pi, B, \sum >$ where,

—π=$p_1, \ldots, p_n$ is a set of n processes, called the publishers, which are producers of information.
—$\sum$=$s_1, \ldots, s_m$ is a set of m processes called the subscribers, which are consumers of information.
—Δ= $B_1, \ldots, B_o$ is a set of o processes, called brokers.

$\sum$ and π may have a non-zero intersection, that is a same process may act both as a publisher and as a subscriber.The general view of publish/subscribe paradigm as shown in the figure 1.

The publishers and subscribers can exclusively communicate with any other process in Delta. Then, the set of brokers Δ represents a logically centralized component of the architecture, which is responsible for collecting subscriptions and forwarding events to subscribers. Delta considered as Notification Service (or Event Service). Publishers and subscribers act as clients for the Notification Service.

The operations of a publish/subscribe system are registration of a subscription σ, cancellation of a subscription , publication of a notification x and issue of the notification of x. A publisher submits a piece of information x to other processes by executing the pub(x) operation on the Notification Service. The Notification Service dispatches a piece of information x submitted by other processes to a subscriber by executing the nfy(x) on it. A subscription is respectively installed and removed on the Notification Service by subscriber processes by executing the sub(σ) and usub(σ) operations. The operation nfy(x) is issued by a process p_i and executed by the p_i, while the other three operations are issued by a process and executed by the PSS. The operation nfy(x) occurs either after the execution of pub(x) or a matching operation executed within the PSS[3].

Notifications provide nature of the information exchanged between publishers and subscribers. A publisher produces an event, while the notification Service issues the corresponding notification on subscribers. Both publishers and subscribers communicate only with a single entity, the Notification Service which,

(1) Stores all the subscriptions associated with the respective subscribers.
(2) Receives all the notifications from publishers.
(3) Dispatches all the published notification to the correct subscribe.

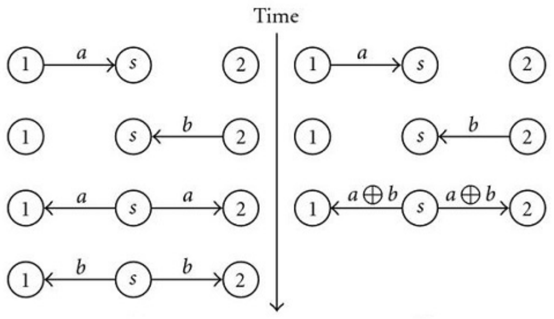

Fig. 2. A network coding example. Nodes 1 and 2 want to exchange packets via an intermediate node S . 1[resp. 2] sends a packet a [resp. b] to 2, which then broadcasts a xor b instead of a and b in sequence. Both 1 and 2 can recover the packet of interest.Both the number of transmissions and time are reduced.

The result is that publishers and subscribers exchange information without directly knowing each other. This anonymity is one of the main features of the publish/subscribe paradigm and simply stems from the level of indirection provided by the Notification Service. Publish/subscribe is an anonymous, many-to-many, asynchronous communication paradigm, where multiple producers may propagate information to multiple consumers. Anonymity is an effective solution to easily get scalability at abstraction level. Participants do not have to know each other and when the size of the system grows, they still have to contact only the Notification Service.

2.1.1 Challenges. The Notification Service should be able to face a large amount of users and to span large-scale communication networks, always maintaining an acceptable level of performance. The first challenge in publish/subscribe is building proposing general models and frameworks that precisely capture and describe the peculiarities of the paradigm. On the algorithmic side, the main trigger of publish/subscribe scheme has been the attempt to build systems that offer a high flexibility to their users, for example allowing them to precisely characterize their interest with powerful. There are two requirements that have to be satisfied: First, pushing the scalability limits of publish/subscribe one step forward, by devising new solutions; second, care about those aspects related to the ease of deployment providing system with dynamic self organization capabilities.

2.2 Network Coding

Network Coding simply ”refer to coding at a node in a network as network coding”, where coding means that arbitrary causal mapping from inputs to outputs.To improve a network’s throughput, efficiency and scalability, as well as resilience to attacks,instead of simply relaying the packets of information they receive, the nodes of a network take several packets and combine them together for transmission.Network coding may have impact on the design of new networking and information dissemination protocols.Traditionally, when forwarding an information packet destined to some other node, it simply repeats it.A simple example in a wireless context is a three node topology, as shown in Figure 2.Network coding allow the node to combine a number of packets it has received or created into one or several outgoing packets.

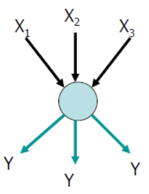

Fig. 3. A network coding working example.Here,h_1,h_2,h_3 are the linear coefficient, which produce the combination of the form,Y=$h_1X_1+h_2X_2+h_3X_3$.Encoding vector is attached with the packet.

With linear network coding, outgoing packets are linear combinations of the original packets, where addition and multiplication are performed over the field [4],where s is a consecutive bits of a packet as a symbol over the field F_{2^s}.Each packet consists of L bits.The packets which do not have the same size, the shorter ones are padded with trailing 0s.s consecutive bits of a packet can be considered as a symbol over the field F_{2^s}.

The original packets produced by several publishers are M^1, M^2 ...,M^n.The coefficients which is used to produce the linear combination are obtained by any of the following way:

—Each node in the network select uniformly at random the coefficients over the field F_{2^s} [9].
—Certain probability of selecting linearly dependent combinations-probability is related to the field size 2^s.[10]
—Polynomial-time algorithm[11] for multicasting which examines each node of the network, and decides what linear combinations each node performs.So that each node uses fixed linear coefficients and the packets only need to carry the information vector.

Each linear combination is given by:

$$X = \sum_{i=1}^{n} g_i M^i$$

The summation have been carried out at each symbol position k,

$$X_k = \sum_{i=1}^{n} g_i M_k^i$$

The coefficient (g_1, ..., g_n) are called encoding vector and the encoded data X is called information vector.A simple example for network coding, as shown in Figure 3.

Encoding can be performed for already encoded packets. Consider a node that has received and stored a set which contain encoding vector and information vector of encoded packets. This node produce the newly encoded packet as the form (g’,X’)by picking a set of coefficients h_1, ..., h_m and computing the linear combination is,

$$X' = \sum_{j=1}^{m} h_j X^j$$

Decoding requires solving a set of linear equations which can be obtained from the system,

$$X^j = \sum_{i=1}^{n} g_i^j M^i$$

This is a linear system with m equations and n unknowns.To solve the linear system ,the coefficients must be independent.To retrieve the all encoded packets,[7] the number of received packets needs to be at least as large as the number of original packets.

For solving a linear system each node stores the encoded vectors with its own original packets, row by row, in a *decoding matrix*. Initially,the matrix only contains the non-encoded packets issued by this node with the corresponding encoding vectors. When an encoded packet is received, it is inserted as last row into the decoding matrix. The matrix is transformed to triangular matrix using Gaussian elimination. A received packet is added to the matrix if it increases the rank of the matrix else discarded. If the matrix contains a row of the form e_i, this node knows that x is equal to the original packet M_i.

The network coding requires finite field operations on F_{2^s}, it means that operations on strings of s bits.Addition is the standard bitwise xor.In multiplication,the sequence is the form of b_0,......,b_{s-1} and this sequence can be interprets as a polynomial $b_0+b_1X+...+b_{s-1}X_{s-1}$.Multiplication is obtained by computing the usual product of two polynomials and then computing the remainder modulo for obtaining an irreducible polynomial. Division is computed by the Euclidian algorithm.

2.2.1 Challenges. For all practical purposes, the size of the matrices with which network coding operates has to be limited. This is straightforward to achieve for deterministic network codes but more difficult with random network coding.The fact that packets need to be decoded has a minor impact on delay. It is usually not necessary to receive all encoded packets before some of the packets can be decoded.

2.2.2 Advantages. Network coding mainly concern performance improvements in static settings.A primary result that sparked the interest in network coding is that it can increase the capacity of a network for multicast flows.The most compelling benefits of network coding might be in terms of robustness and adaptability. Network coding brings, is that the linear combining is performed opportunistically over the network, not only at the source node, and it is well suited for the (typical) cases where nodes only have incomplete information about the global network state.Network coding protocol suffers only a small performance penalty when incentive mechanisms to cooperate are implemented.

2.3 Gossiping

Gossiping is a probabilistic and fully distributed approach to event routing and achieves high stability under high network dynamics, and scales gracefully to a huge number of nodes.It is performed at the time of recovering any lost packets is found at receiver/subscriber side.In gossiping, each node contacts one or a few nodes in each round(usually chosen at random), and exchanges information with these nodes [6]. The dynamics of information spread resembles the spread of an epidemic[12] and lead to high robustness, reliability and self-stabilization[8]. Being randomized, rather than deterministic, this technique is simple and do not require to maintain any event routing data structure at each node.The existing systems implements random uniform selection mechanism for gossip partners, those who are participating in gossip rounds.

Missing messages are recovered through one or more gossip rounds, during which other processes potentially holding a copy of the data are contacted.A gossip round consists of the following steps:

(1) Process A chooses randomly another process to communicate with, say B.
(2) A sends to B information that allows to determine the presence of inconsistencies in their view of the system’s state.
(3) A and B reconcile their state by exchanging the actual messages that are not part of the history of both.

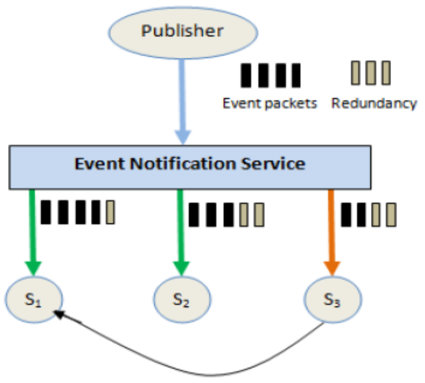

Fig. 4. Dissemination and recovery of an event notification

In gossiping, the random choice of the nodes to contact can be sometimes driven by local information, acquired by a node during its execution, describing the state either of the network or of the subscription distribution or both.If any subscriber detects that the message was incomplete then a gossip round is performed to recover complete message. As shown in the figure 4,the encrypted packets are disseminated from a publisher to all the subscribers of that corresponding event published. The nodes $s_1$and s_2 has enough packets to reconstruct the whole message but s_3 has to go for a gossip round.

Event dissemination protocols use gossip to carry out multicasts [5]. They report events, but the gossip occurs periodically and events dont actually trigger the gossip. One concern here is the potentially high latency from when the event occurs until it is delivered.For each gossip round, nodes receives a message and it stores that in a buffer of size ”b”, called as fan-in, if the node forwards a message limited number of times ”t” to other nodes in the overlay network is called as fan-out.There are several variants of gossiping algorithms exist:

—Push Approaches: Messages are forwarded from one to other nodes as soon as they are received.

—Pull Approaches: Nodes are periodically send it’s own list of recently received messages to another node.If a lost message is detected by comparing the received list with the history of messages received by the givennode, then a transmission is requested.

—Push/Pull Approach: Node forwards to the other nodes only the identifier of the last received message;if one of the receivers does not have such message, then it makes an explicit pull request.

Publish/subscribe based event dissemination services are applied with gossiping efficiently and these algorithms provided the timeliness constraint to publish/subscribe services.The number of gossip rounds to be performed to recover ”m” lost packets had drastically fallen from O(mlog(m)) to O(m).

2.3.1 Challenges. An event match many patterns instead of a single subject,and the source is not required to tag the published event in any way, since routing is entirely determined by the event content.In gossip interactions,to simply route gossip messages as normal events is difficult.In a topic-based system, the topic defines the set of nodes it is useful to gossip with, since it contains the set of receivers for a given event associated to that subject. In content-based systems,this set of nodes cannot be determined as easily since the subject notion is missing.

2.3.2 Advantages. Gossip algorithms impose a constant, equally distributed load on the processes in the system, and are very resilient to changes in the system configuration since they do not rely on the existence of one or more processes. Moreover, these properties are preserved as the size of the system increases, thus leading to good scalability. Finally, these algorithms are very simple to implement and rather inexpensive to run. Gossip algorithms are then a good match for highly distributed and dynamic scenarios.

3. CONCLUSION

A publish/subscribe scheme encloses a communication model providing the necessary component decoupling,reliablity,expressiveness, and scalability to deal with these characteristics at the application level.The problem related is that it either provide only a best-effort service or address just a single requirement at a time such as message ordering, bounded delivery time, or reliable delivery.While some applications can rely on a best-effort service (messaging, social networking), other applications may require several nonfunctional requirements to be met.This paper present two solutions that introduce reliability and timely event notification for publish-subscribe systems.Coding has a positive impact even on the mean notification latency and its standard deviation due to a decrease in the number of retransmissions required.And gossiping,which increases the retrieval of missing packets in case of incomplete information.There is a large scheme to develop more efficient techniques there by achieving speedy processing for publish/subscribe communication.

4. REFERENCES

[1] Christian Esposito, Marco Platania, and Roberto Beraldi ”Reliable and Timely Event Notification for Publish/Subscribe Services Over the Internet” *IEEE/ACM Transactions on Networking*, Vol 22,No1,February 2014.

[2] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec ”The many faces of publish/subscribe,” *ACM Computing Surveys* (CSUR), vol. 35, no. 2, pp. 114131, June 2003.

[3] R. Baldoni, M. Contenti, S. Piergiovanni, and A. Virgillito ”Modeling publish/subscribe communication systems: Towards a formal approach,” *in Proc. 8th IEEE Int.* Workshop Object-Oriented Real-Time Depend. Syst., 2003, pp. 304311.

[4] C. Fragouli, J. L. Boudec, and J. Widmer ”Network coding: an instant primer,” *Computer Communication Review*, vol. 36, no. 1, p. 63, 2006.

[5] C. Esposito, S. Russo, R. Beraldi, and M. Platania ”On the benefit of network coding for timely and reliable event dissemination in WAN,” *in Proc. 1st Int. Workshop Netw. Resilience*, Oct. 2011, pp. 8489.

[6] Paolo Costa, Matteo Migliavacca, Gian Pietro Picco, and Gianpaolo Cugola ”Introducing Reliability in ContentBased Publish/Subscribe through Epidemic Algorithms,” *20133 Milano*, Italy.

[7] S. Li, R. Yeung, and N. Cai ”Linear network coding,” *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371381, Feb. 2003.

[8] C. Esposito, S. Russo, R. Beraldi, M. Platania, and R. Baldoni. ”Achieving reliable and timely event dissemination over wan” *in Proc. 13th ICDCN*, 2012, pp. 265280.

[9] C. Fragouli, J. Widmer, and J.-Y. L. Boudec. ”The benefits of coding over routing in a randomized setting”, *International Symposium on Information Theory (ISIT)*, page 442, July 2003.

[10] Y. Wu, P. A. Chou, and K. Jain.” A comparison of network coding and tree packing”. *ISIT 2004*, 2004.

[11] P. Sanders, S. Egner, and L. Tolhuizen.” Polynomial time algorithms for network information flow,” *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures*, 2003.

[12] A.-M. Kermarrec, L-Massoulie, and A. J. Ganesh, ”Probabilistic Reliable Dissemination in Large-Scale Systems,” *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 14, no. 2, pp. 111,February 2003