Volume 6, Issue 1, January 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Reliable and Timely Content-Based Publish/Subscribe System

Rahul Shinde, Ashvini Jadhav
ME Computer Networks, Savitribai Phule Pune University, Nutan Maharashtra Institute of Engineering and Technology College Pune, Maharashtra, India

Abstract— Publish/Subscribe systems are now becoming popular. The peer to peer architecture is most powerful and can be combined with publish/subscribe systems to form a reliable, secure and faster system. There are two kinds of publish subscribe systems, subject or topic based and content based. In content based publish subscribe systems, the power of peer to peer systems can be utilized effectively. Peer to peer systems are highly scalable, fault tolerant, decoupled, flexible and expressive. Currently there are various content based peer to peer systems present. Each system has its own pros and cons. The efficiency of the peer to peer systems is mostly depend on the load balancing, searching algorithms, reliable data dissemination and better loss recovery. This work is going to cover detailed study of these systems and help to improve the throughput of the system by timely and reliable data dissemination using network coding.

Keywords— Peer to Peer, Publish/Subscribe, DHT- Distributed Hash Tables, Content Based, Network Coding.

I. INTRODUCTION

It's been always said that current world is Internet's world. Lots of people treat Internet as a medium of sending and receiving information. One of the way to receive such kind of information is event notification systems. Basically event notification systems follow the Event Driven Architecture. Event based system came from their beginnings as database triggers to current Internet - wide notification services. Event dissemination systems are also called publish subscribe systems where a subscriber subscribes to an event and publishers publishes to the interested subscribers. In software architecture publish subscribe is the messaging pattern where publisher (sender) don't have knowledge of all the subscribers. It characterize publish subscribe messages into classes and subscribers express their interest into that classes. There is a process to select the subscriber to send the message. This process also called as filtering. There are two common forms of filtering: topic-based filtering and content based filtering.

In topic-based systems, messages are published to particular "topics" or we can say logical channels. So the subscribers who are subscribed to the topic will receive all the messages regarding that topic. Scribe is one of the most popular systems which is based on this paradigm. In content-based systems the messages are delivered to subscribers if the contents or attributes of the message match to the constraints of the subscriber. In content-based systems, subscribers are able to define their criteria so that they will receive only information in which they are interested.

When we consider distributed content-based publish/subscribe data dissemination systems, peer to peer networks can offer benefits. A peer to peer networks total resources grow as the number of participants increases into network. There are various large scale publish subscribe systems increasingly common in the industry. Publish subscribe systems arise in many applications including online stock quotes, Internet games, and sensor networks. In the stock quote application, the events are generated by various stock exchanges where trading occurs. The events contain information about the open, close, low, and high values of companies stocks. The subscribers are clients interested in trading, and they are usually interested in the values of the stocks for which they trade. It is most popular way now a day to get a data based on certain event.

II. LITERATURE SURVEY

DMM [1] and DMM-AR [1] are algorithms that implements a pub/sub system over a Distributed Hash Table, without requiring any centralized schema knowledge. This is done by mapping publications and subscriptions in the publish/subscribe domain into regions in a multidimensional space. This multidimensional space is indexed with a distributed search tree, which allows matching multiple publish/subscribe attributes simultaneously. The multidimensional index gets advantage by storing subscriptions at multiple nodes so that it can achieve a bottom-up publication matching that avoids root hotspots in the index. Also, it extends traditional publish/subscribe semantics, the matching algorithm highly supports publications with different level of range values.

Here the search tree grows as the complexity of subscriptions increases. Two fault tolerance techniques, active failure detector and period state refreshing allow the algorithms to recover from any type of crash.

Meghdoot [2] implements a content-based publish/subscribe system over a peer to peer based Distributed Hash Table to improve scalability, based on CAN. P2P architecture offers the flexibility of incorporating additional resources at any time, thus providing performance scalability. Meghdoot don't impose any restrictions over type of subscriptions and allows them to be specified in terms of range predicates over all attributes in a schema. Meghdoot uses the semantics of

© 2016, IJARCSSE All Rights Reserved

the subscriptions and the events to store subscriptions and deliver matching events to them. Since real world datasets are not strait forward or we can say skewed, most of the existing systems fails to distribute load among peers. Meghdoot uses the characteristics of the load and uses it efficiently to distribute the load among various nodes. Subscription load leads to zone splitting, while event propagation load leads to zone replication. In Meghdoot subscriptions are replicated in an innovative and systematic way to handle fault tolerance.

As there are always more publications than subscriptions it is not helpful to optimize design of subscription state, which is done by Meghdoot.

Scribe [3] is channel / topic based publish / subscribe system. Scribe uses the different properties of Pastry like scalability, locality, fault-resilience and self-organization properties. The Pastry works on a mechanism which builds an efficient multicast tree. Scribe creates groups of nodes and multicast them which balances the load among the participating nodes. Pastry's properties enable Scribe to expressively use locality to build an efficient multicast tree and to handle group join operations in a decentralized manner. As Scribe is channel based it has limited filtering capabilities.

Hermes [4] is a pub/sub system that is built over the Pastry DHT. Hermes uses special nodes which can be called as rendezvous nodes, which are set up through event type messages submitted prior to publishing. Herms uses two types of components event brokers and event clients. Event brokers implement all the functionality of the Herms and event clients can be either publishers or subscribers. Event clients are light weight and connect to event brokers to use any type of service. Herms supports two types of content-based routing: In *type-based* routing subscribers receives all the events. In *type and attribute based* routing allows subscribers to further filter to the event type's attributes.

Terpstra [5] is built over the Chord DHT, is a content-based pub/sub system. Instead of creating a single multicast tree from root, it creates a separate multicast trees rooted at each broker and because of this it is able distribute node equally. This uses flooding of subscriptions to create multicast tree. Terpstra is built on top of a dynamic peer-to-peer overlay network. Separate components in Terpstra ensure that the network self-organises itself to maintain the topology and can survive simultaneous failure of up to half of its nodes. The main disadvantage of Terpstra is its flooding mechanism which may be drastic.

New communication paradigm - network coding [6], in which the intermediary relay stations are allowed to perform encoding/decoding operations on the information they receive. Network coding is most effective for multicast communication. If the intermediary nodes are allowed to perform coding on the information they receive such as taking the xor of two packets or other operations within the appropriate finite field, then we have a coding strategy. Network coding may have impact on the design of new networking and information dissemination protocols [6].

Recovery Strategies:

Automatic Repeat Request (ARQ) [7] is the scheme that consists of detection of possible omission of messages due to some kind of faults and asks for retransmissions. ARQ gets divided into 3 classifications depending on contacted nodes for retransmissions. *Sender based:* all nodes always contact source of multicaster for retransmission. *Parent based:* a node always contact its parent node in the multicast tree for retransmission. *Neighbour based:* retransmissions are done by neighbouring nodes.

Neighbour based gets divided into three categories:

- Lateral Error Recovery [8]: Random nodes are grouped together and node selects its neighbours for recovery.
- Cooperative Error Recovery [9]: Nodes are clustered together in different groups whose members are characterized by negligible loss correlation and node selects its neighbours between members of its group.
- Gossiping [13]: Node stores received message in in a buffer and sends it to a randomly selected nodes for limited number of times. There are various gossiping algorithms exists like *Push Approaches, Pull Approaches and Push/Pull Approaches*.

Forward Error Correction (FEC) schemes contains techniques which are based on generation of additional information based of message content, so that lost data can be recovered by decoding all the information received from packet. FEC is divided into three different approaches [14]:

- End-to-End FEC: Encoding is done by sender / multicaster.
- Link-by-Link FEC: Every node in the network performs encoding and decoding.
- Selective Network-Embedded FEC: Only a subset of selected nodes can perform encoding.

Last group of strategies includes approaches based on path redundancy in which nodes are interconnected by more than one link. There are two different approaches depending on organization of nodes, Mesh and Tree.

III. PROBLEM STATEMENT

Publish subscriber systems should be scalable and it must take care of scalability in terms of *Subscription Management, Efficient Event Matching and Efficient Events Distribution.* DMM [1] and DMM-AR [2] algorithms are good at subscription management and efficient event matching but when it comes to recovery of missing information at destination nodes these algorithms have scope of improvement. It takes too much time to recover lost information / packets at node level because there is no mechanism in place which helps to ensure more reliable and timely dissemination of the data.

The DMM-AR [1] algorithm uses TreeCache [1] optimization to improve the performance by reducing number of hopes a publication or subscription travels and thus reduces the chances of message being lost. This mechanism helps to

© 2016, IJARCSSE All Rights Reserved

reduce retransmission but not the extent which nowadays publish subscribe systems needed. Hence there is a need of an algorithm which helps in reducing retransmission rate and helps destination nodes to recover information with negligible retransmissions.

IV. PROPOSED SOLUTION

The system design consists of algorithms for encoding and decoding. In encoding network coding will be used to create encoded packets which can be sent with original packets. Also at destination node decoding will be performed for original packets and if there is any loss then redundant packets will be used to find lost packets.

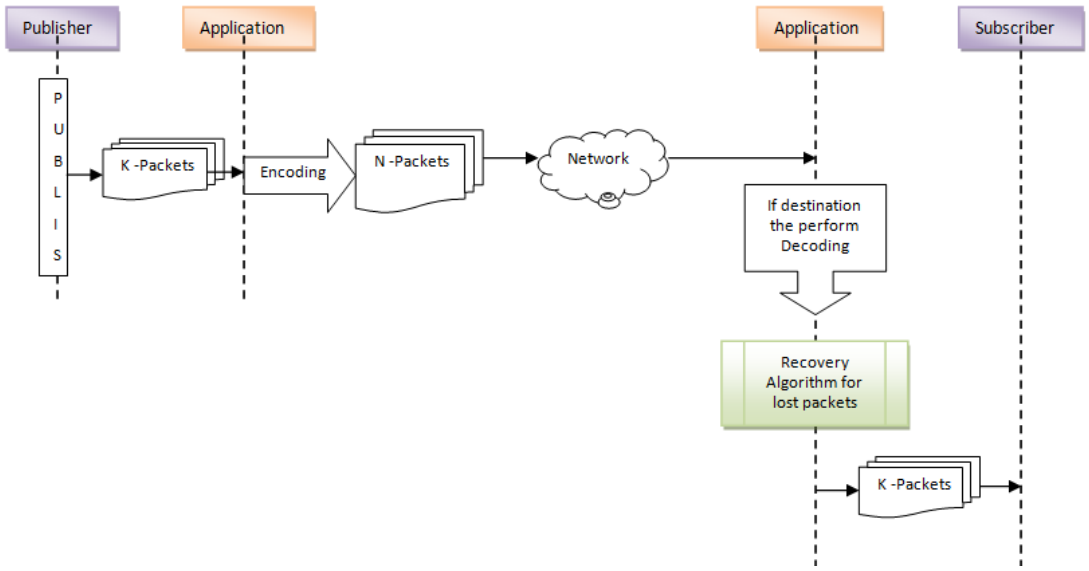

Fig. 1 System design with use of network coding for encoding and decoding.

Figure 2, 3 and 4 gives an idea of traditional approach and network coding approach. In the traditional approach if the packet is lost then intermediate node sends request for retransmission and waits to receive the lost packet. But in case of network coding approach the intermediate node generate the lost packet by linearly encoded packets and again sends linearly encoded packets.

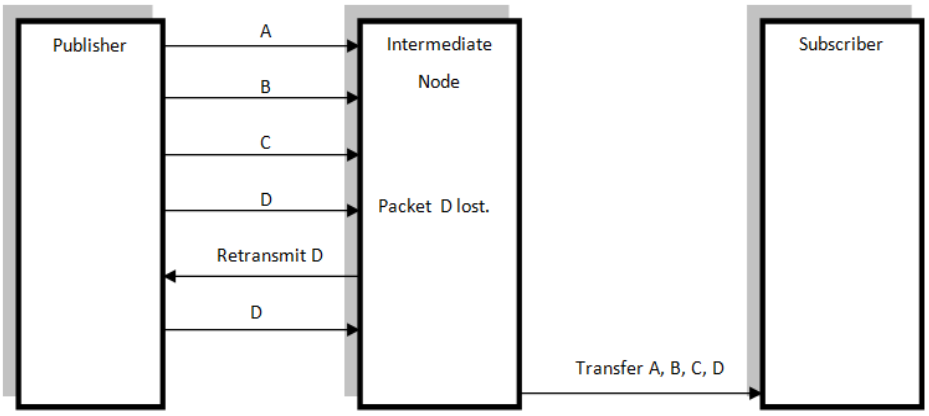

Fig. 2 Traditional Approach.

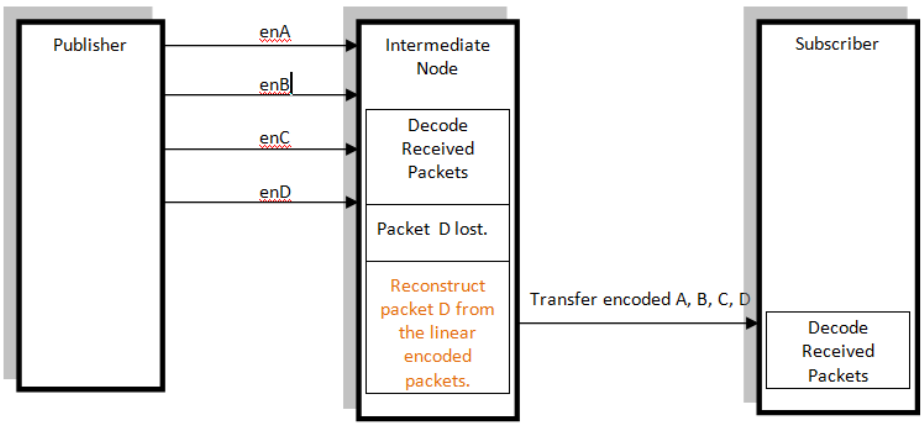

Fig. 3 Network Coding Approach.

© 2016, IJARCSSE All Rights Reserved

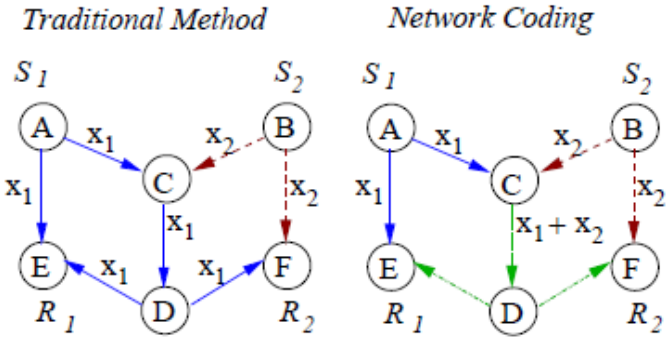

Fig. 4 [6] (Butterfly Network) S1 and S2 are senders and multicast to R1 and R2. All links are with capacity 1. With network coding (by xoring the data on link CD), the achievable rates are 2 for each source, the same as if every estimation were using the network for its sole use. Without network coding, the achievable rates are less (for example if both rates are equal, the maximum rate is 1.5).

V. CONCLUSIONS

As discussed in the report there are 3 main scalability options present in publish subscribe systems. Subscription Management, Efficient Event Matching and Efficient Event Distribution. The algorithms discussed in literature survey are very good at first two options but there is still scope for improvement in third option. Traditional approaches of loss recovery generally follows techniques which involves retransmissions. In this report a new approach is discussed and can be beneficial to improve performance and timeliness of publish subscribe system. The concept uses network coding paradigm to encode packets at sender and at intermediate nodes and the encoding can be implemented in such a way that if some information is missed at receiver or at intermediate node then it can be recovered using total received information. This technique will not only save retransmission time but also utilize computing power of nodes of peer to peer systems.

REFERENCES

[1] Vinod Muthusamy and Hans-Arno Jacobsen, "Infrastructure-Free Content-Based Publish / Subscribe",VOL. 22, NO. 5, OCTOBER 2014.

[2] A. Gupta, O. D. Sahin, D. Agrawal, and A. El Abbadi, "Meghdoot: Content-based publish/subscribe over P2P networks," in Proc. Middleware, 2004, pp. 254-273.

[3] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: A large-scale and decentralized application-level multicast infrastructure," IEEE J. Sel. Areas Commun., vol. 20, no. 8, pp. 1489-1499, Oct. 2002.

[4] P. R. Pietzuch and J. Bacon, "Peer-to-peer overlay broker networks in an event-based middleware," in Proc. DEBS, 2003, pp. 1-8.

[5] W. W. Terpstra, S. Behnel, L. Fiege, A. Zeidler, and A. P. Buchmann, "A peer-to-peer approach to content-based publish/subscribe," in Proc. DEBS, 2003, pp. 1-8.

[6] C. Fragouli, J. Le Boudec, and J. Widmer, "Network coding, an instant primer," Computer Communication Review, vol. 36, no. 1, p. 63, 2006.

[7] X. Jin, W. P. K. Yiu, and S. H. G. Chan, "Loss Recovery in Application-Layer Multicast," IEEE MultiMedia, vol. 15, no. 1, pp. 18–27, January 2008.

[8] W.-P. K. Yiu, K.-F. S. Wong, S.-H. G. Chan, W.-C. Wong, Q. Zhang, W.-W. Zhu, and Y.-Q. Zhang, "Lateral Error Correction for Media Streaming in Application-Level Multicast," IEEE/ACM Transactions on Multimedia (T-MM), vol. 8, no. 2, pp. 219–232, April 2006.

[9] G. Tan, S. A. Jarvis, and D. P. Spooner, "Improving the Fault Resilience of Overlay Multicast for Media Streaming," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 18, no. 6, pp. 721–734, June 2007.

[10] A.-M. Kermarrec, L-Massouli´e, and A. J. Ganesh, "Probabilistic Reliable Dissemination in Large-Scale Systems," IEEE Transactions on Parallel and Distributed Systems (TPDS), vol. 14, no. 2, pp. 1–11, February 2003.

[11] S. Birrer and F. Bustamante, "A Comparison of Resilient Overlay Multicast Approaches," IEEE Journal on Selected Areas in Communications (JSAC), vol. 25, no. 9, pp. 1695–1705,December 2007.

[12] N. Magharei and R. Rejaie, "PRIME: Peer-to-Peer Receiverdriven Mesh-based Streaming," Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM 07), pp. 1415–1423, May 2007.

[13] A.-M. Kermarrec, L-Massoulié, and A. J. Ganesh, "Probabilistic Reliable Dissemination in Large-Scale Systems," IEEE Trans-actions on Parallel and Distributed Systems (TPDS), vol. 14, no. 2, pp. 1–11, February 2003.

[14] M. Wu, S. S. Karande, and H. Radha, "Network-embedded FEC for Optimum Throughput of Multicast Packet Video," Journal on Signal Processing: Image Communication, vol. 20, no. 8, pp.728–742, September 2005.

© 2016, IJARCSSE All Rights Reserved

ABOUT AUTHOR

Mr. Rahul Ramchandra Shinde, completed B.E. Computer Engineering from Pune University, Maharashtra, India in 2008 and currently pursuing M.E. Computer Networks from Savitribai Phule Pune University. I am conducting a research on publish subscribe systems over peer to peer networks and network coding to improve the performance of publish subscribe system.

Prof. Ashvini Jadhav, completed M.E. Computer Networks from Pune University, Maharashtra, India in 2013. I am currently working in Nutan Maharashtra Institute of Engineering and Technology College as Assistant Professor.

© 2016, IJARCSSE All Rights Reserved