Highly Available Publish/Subscribe

Zbigniew Jerzak
Dresden University of Technology
D-01062 Dresden, Germany
Telephone: ++49 351 463–39708
Fax: ++49 351 463–39710

***Abstract*— We propose a novel approach for ensuring the availability of a publish/subscribe (P/S) service with limited resources. Our approach complies with the fully decoupled nature of P/S services.**

I. Introduction

The rapid growth of the Internet and local area networks has contributed to the increasing size and importance of the distributed systems. Distributed systems span thousands of loosely coupled entities whose behaviour and location varies during the system's lifetime. Moreover, due to the network component failures and the limited capabilities of different devices, the point-to-point synchronous communication model does not meet the high availability requirements. Therefore, we believe publish/subscribe (P/S) scheme to pose an alternative to the current approach to the construction of the highly available distributed systems.

A P/S scheme and the recently introduced content-based networking [1], [2] permit the construction of fully decoupled systems as far as *time*, *space* and *synchronisation* are concerned [3]. This makes the P/S scheme a perfect candidate for information dissemination in loosely coupled networks where no process has full knowledge about the network structure.

Our work focuses on the issues of high availability of distributed systems using publish/subscribe services. We especially address the issues of computational limitations of the network routers and potential link failures when low bandwidth links are used for backup. We propose a novel scheme for coping with the problem of load shedding and service availability in the P/S scheme, where the users compete with each other for the access to the system resources.

II. Related Work

Publish/subscribe schemes have recently drawn a considerable attention. They have evolved from relatively simple and static topic-based systems [4] to complex and powerful content-based solutions [5], [6]. Most applications and derived services do, however, not utilize the flexibility of the fully decoupled publish/subscribe design. In case of Astrolabe [7] this is due to the need for hierarchical system architecture and gossip based status updates. The GridStat approach, described in [8], uses mixed peer-to-peer and hierarchical structures in order to enforce the service level agreements on the parameters of the information delivery, which yields scalability problems. We believe that in order to take full advantage of the publish/subscribe model, one should not impose any requirements that are in contrast with its decoupled nature. We believe that content-based networking and approach proposed in [1], [2] is the closest one to the publish/subscribe nature. The aforementioned work does not, however, address the issue of the system availability in case of the resource shortage. Our high availability algorithm has been influenced by the *smart market* approach presented in [9].

III. High Availability in P/S

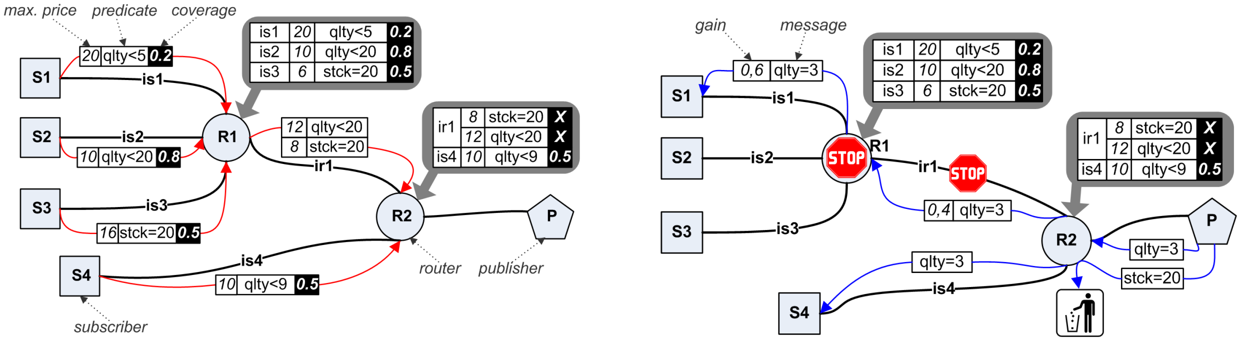
We believe that when facing link congestion and willing to provide high availability for the publish/subscribe scheme we must not violate any of the three fundamental P/S decoupling assumptions mentioned in the introduction. With this in mind we can clearly see that there are currently no approaches available which would allow us to cope with this problem. Moreover, we believe that it is not possible to provide guarantees on the system availability to all the users. It is obvious that attempts to deliver service to all the requesting parties in case of an overload lead to further congestion and degradation in the service's availability. On the other hand, it is not possible to perform a priori link reservation as proposed in [8] and eventual load shedding without violating the synchronisation decoupling property.

In our approach clients of a P/S service compete with each other by providing prices describing their willingness to pay for information. The price which is propagated along with the demand for the information indicates the maximum amount a client is willing to pay for that information. When there is no congestion and there are no computational limitations in the network, users are charged flat rate for their access. However, as soon as there is a need to perform load shedding, the routers which forward the information choose to skip the links or the messages which will be the least "profitable" – i.e. would result in smaller revenue/profit. In case of multiple routers between the publisher and the subscriber, the prices are aggregated on the links in order to allow for recognition of the most "profitable" ones.

We use two load shedding strategies. The first one is used in case of the overload of the router and involves removing entries from the router's routing table and hence, reduces the time needed to search it. The second one is used whenever there is a link overload on any of the outgoing links. It limits the forwarding of messages to the most "profitable" ones.

IV. The Protocol

The flow of the messages from the publisher to the subscriber is driven by the message content which is structured as attribute/value pairs. Subscribers express their interest in certain messages by means of selection predicates which are logical disjunctions of conjunctions of elementary constraints over the values of individual attributes. For brevity of description we use only single attribute messages and predicates. Messages issued by the publishers are broadcasted to the network with broadcast distribution tree branches pruned by the selection predicates. Selection predicates are broadcasted by the subscribers and piggyback following information: the maximum price (per message) a client is willing to pay for its reception and the approximate coverage of the space state (expressed in percent) by the provided selection predicate – see Figure 1(a). Selection predicates and the maximum prices can be aggregated in the routers. The new price of the aggregate predicate is calculated as $\sum_{i:i\in\, A} c(i)p(i)$



(a) Routing of predicates. Router R1 aggregates the $qlty < 5$ and $qlty < 20$ predicates. Subscription messages consist of three parts: maximum price, predicate and approximate coverage.

(b) Forwarding of events. Router R2 knowing that link ir1 is overloaded drops the $stck = 20$ message in favour of the $qlty = 3$ one and calculates the gain value. The overloaded router R1 eliminates the link is2 from its routing table and does not forward the $qlty = 3$ message to the subscriber s2.

Fig. 1. Information dissemination in the highly available P/S scheme.

where A is a set of all aggregated predicates, $c(i)$ is the coverage declared in the ith predicate and $p(i)$ is the declared maximum price.

In order for the actual “charging” to take place the messages issued by the publishers piggyback the so-called gain value – G. The gain value describes how much of the original advertised maximum price should the client be charged. The computation of the total charge is distributed among all the routers on the way from the publisher to the subscriber of the message. Iff there is a need to drop messages or perform load shedding in the router it computes the local gain value (G_{l1} or G_{l1}) with the equation given below:

$$G_{l1} = \frac{\sum_{i:i\in\, E} p(i)}{\sum_{j:j\in\, N} p(j)} \qquad G_{l2} = 1 - \frac{\sum_{i:i\in\, F} p(i)}{\sum_{j:j\in\, N} p(j)} \qquad (1)$$

In case of overload, router computes G_{l1}, where E represents all excluded from consideration (shedded) links, whilst N represents the set of all outgoing links. G_{l2} is computed in case of link overload. F represents the predicates from the given link in the routing table which are covered by messages that are forwarded, whilst N represents the set of all predicates on the given link. The two different strategies are represented by routers R2[1] and R1 on Figure 1(b).

The second step in calculation of the final gain value (G – Equation 2) for a given message is the inclusion of the previous gain value (G_{old}) calculated by the router which had to perform load shedding and was upstream from the current one.

$$G = G_{old} + (1 - G_{old}) \cdot G_l \qquad (2)$$

The gain values calculated this way increase with the flow of the messages through the routers which had to perform load shedding. The final price a subscriber will be charged for receiving a message is calculated by the last router delivering the message directly to the subscriber. It is a multiplication of the gain value piggybacked on the message and the maximum price declared by the client stored in the routing table of the last router next to the client's predicate. Charged amount is saved on the router and associated with the client.

The coverage value in the message helps the system to cope with the problem of determining how many users will actually receive a message if it is propagated on the given link. Please note that aggregated predicates do not allow for such estimation. The coverage value describes the probability that the message arriving at the router will actually be forwarded to the given subscriber. If we consider scenario where the publisher P publishes a message $qlty = 8$ we can clearly see that the subscriber $S1$ will not receive it although its maximum price will be taken under consideration when deciding about the link “profitability”. In order to prevent subscribers from specifying the high coverage values, they will be charged for every message arriving at the router and not forwarded to them. The charged amount a is equal to: $a = gain \cdot coverage \cdot maxprice$. For direct delivery $a = gain \cdot maxprice$.

V. Conclusions and Future Work

We believe the proposed scheme offers a promising alternative regarding the current approaches to the high availability in the P/S schemes. We are currently working on evaluating our design.

References

[1] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, “A routing scheme for content-based networking,” in *Proceedings of IEEE INFOCOM 2004*, Hong Kong, China, March 2004.

[2] A. Carzaniga and A. L. Wolf, “Forwarding in a content-based network,” in *Proceedings of ACM SIGCOMM 2003*, Karlsruhe, Germany, Aug. 2003, pp. 163–174.

[3] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, “The many faces of publish/subscribe,” *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003.

[4] “TIB/Rendezvous,” White Paper, TIBCO, 1999.

[5] A. Carzaniga, D. S. Rosenblum, and A. L. Wolf, “Design and evaluation of a wide-area event notification service,” *ACM Trans. Comput. Syst.*, vol. 19, no. 3, pp. 332–383, 2001.

[6] Y. Zhao, D. Sturman, and S. Bhola, “Subscription propagation in highly-available publish/subscribe middleware,” in *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*. New York, NY, USA: Springer-Verlag New York, Inc., October 2004, pp. 274–293.

[7] R. V. Renesse, K. P. Birman, and W. Vogels, “Astrolabe: A robust and scalable technology for distributed system monitoring, management, and data mining,” *ACM Trans. Comput. Syst.*, vol. 21, no. 2, pp. 164–206, 2003.

[8] K. H. Gjermundrød, I. Dionysiou, D. Bakken, C. Hauser, and A. Bose, “Flexible and robust status dissemination middleware for the electric power grid,” School of Electrical Engineering and Computer Science Washington State University, Pullman, Washington 99164-2752 USA, Technical Report EECS-GS-003, September 2003.

[1] Router R2 has to be able to see both messages $quality = 3$ and $stock = 20$ (Figure 1(b)) prior to deciding about forwarding one of them in favour of the other. This can be achieved by parsing of the incoming message queue.

[9] J. K. MacKie-Mason and H. R. Varian, “Pricing the internet,” prepared for the conference Public Access to the Internet, JFK School of Government, May 1993.