Channel Islands in a Reflective Ocean: Large Scale Event Distribution in Heterogeneous Networks

Jon Crowcroft

University of Cambridge
Computer Laboratory
William Gates Building
J J Thomson Avenue
Cambridge
CB3 0FD
Jon.Crowcroft@cl.cam.ac.uk

Abstract. This is a discussion paper about the possible future use of network and transport level multicast services to support extremely large scale event distribution.
To date, event notification services[40] have been limited in their scope due to limitations of the infrastructure At the same time, Internet network and transport layer multicast services have seen limited deployment due to lack of user demand (with the exception more recently of streaming services, e.g. on Sprint's US core network, and in the Internet II). Recent research in active and reflective middleware suggests a way to resolve these two problems at one go.
Event-driven and messaging infrastructures are emerging as the most flexible and feasible solution for enabling rapid and dynamic integration of legacy and monolithic software applications into distributed systems. Event infrastructures also support deployment and evolution of traditionally difficult-to-build active systems such as large-scale collaborative environments and mobility aware architectures.
Event notification is concerned with propagation of state changes in objects in the form of events. A crucial aspect of events is that they occur asynchronously. Event consumers have no control over when events are triggered. On the other hand, event suppliers do not generally know what entities might be interested in the events they provide. These two aspects clearly define event notification as a model of asynchronous and de-coupled communication, where entities communicate in order to exchange information, but do not directly control each other.
The IETF is just finishing specifying a family of reliable multicast transport protocols, for most of which there are pilot implementations. Key amongst these for the purposes of this research is the exposure to end systems of router filter functionality in a programmable way, known as *Generic Router Assist*. This is an inherent part of the Pragmatic General Multicast service, implemented by Reuters, Tibco and Cisco in their products, although it has not been widely known or used outside of the *TIBNET* products until very recently.

E. Gregori et al. (Eds.): NETWORKING 2002, LNCS 2345, pp. 1–9, 2002.

The goal of this paper is to describe a reflective middleware system that integrates the network, transport and distributed middleware services into a seamless whole.
The outcome of this research will be to integrate this 'low-level' technology into an event middleware system, as a toolkit as well as evaluation of this approach for massive scale event notification, suitable for telemetry, novel mobile network services, and other as yet unforeseen applications.

1 Background and Introduction

The last decade has seen the great leaps in the maturity of distributed systems middleware, and in one particular area in support of a wide variety of novel applications, event notification systems. Current work on event notification middleware[39][40][41], has concentrated on providing the infrastructure necessary to enable content-based addressing of event notifications. These solutions promote a publish-subscribe-match model by which event sources publish the metadata of the events they generate, event consumers register for their events of interest passing event filter specifications, and the underlying event notification middleware undertakes the event filtering and routing process. Solutions differ usually on whether they undertake the filtering process at the source or at an intermediary mediator or channel in which the event filtering takes place. The trade-off lies on whether to increase the computational load of sources and decrease the network bandwidth consumption, or minimise the extra computational load on the sources and outsource the event filtering and routing task to a mediator component (hopefully located close to the source). All of these solutions do not leverage on the potential benefits that event multicasting to consumers requiring the same type of events, and applying very similar filters could bring. They usually require an individual unicast communication per event transmitted.

At the same time, the underlying network has become very widespread. New services such as IP multicast are finally seeing widespread deployment, especially in core networks and in intranets.

The combination of these two technologies, event services and multicast, originates historically with Tibco[20], a subsidiary of Reuters. However, their approach is somewhat limited as it takes a strict layered approach.

At the highest level, there is a publish/subscribe system, which in *TIBNET* uses *Subject Based Addressing* and *Content Based Addressing*. Receivers subscribe to subjects. The Subject is used to hash to a multicast group. Receivers subscribe to a subject but can express interest by declaring filters on content. The *TIBNET* system is then hybrid. In the wide area, IP multicast is used to distribute all content on a given subject topic to a set of site proxy servers. The site proxy servers then act on behalf of subscribers at a site and filter appropriate content out of each subject stream and deliver the remains to each subscriber.

Between the notification layer and the IP layer there is a transport layer, called Pragmatic General Multicast. To provide semi-reliable, in-order delivery,

the subject messages are mapped onto PGM[10] messages, which are then multicast in IP packets. PGM provides a novel retransmission facility which takes advantage of router level "nack aggregation" (which itself prevents message implosion towards the event source), to provide filtering[15][16] of retransmissions so that only receivers missing a given message sequence number, receive it. The PGM protocol is essentially a light weight signaling protocol which allows receivers to install and remove filters on parts of the message stream. The mechanism is implemented in Cisco and other routers that run IP multicast. The end system part of the protocol is available in all common operating systems.

Almost all other event notification systems have taken the view that IP multicast was rarely deployed[1], and that the overheads in the group management protocols were too high for the rate of change of interest/subscription typical in many applications usage patterns.

Instead, they have typically taken an alternative approach of building a server level overlay for event message distribution. Recent years have seen many such overlay attempts[22] [23] [24] [25] [26] [27] [28] [29] [30]. These have met with varying degrees of success. One of the main problems of application layer service location and routing is that the placement of servers does not often ,match the underlying true topology of the physical network, and is therefore unable to gain accurate matching between a distribution tree and the actual link throughput or latencies. Nor is the system able to estimate accurately the actual available capacity or delay. Even massive scale deployments such as Akamai[31], for example, do not do very well.

Secondly, the delays through application level systems are massively higher than those through routers and switches (which are after all designed for packet forwarding, rather than server or client computation or storage resource sharing). The message is that overlays and measurement are both hard to optimise, and inefficient.

We see a number of advantages in continuing forward from where Tibco left off in integrating efficient network delivery through multicast, with an event notification service including:

Scale. We obviate the need to deploy special proxy servers to aid the distribution.

Throughput. We will be able therefore to distribute many more events per second.

Latency. Event distribution latency will approximate the packet level distribution delay , and will avoid the problems of high latency and jitter incurred when forwarding through application level processes on intermediaries.

There are two ideas we will draw from in moving forward. Firstly we will exploit advances in the network support for multicast, such as Generic Router Assist service in the PGM router element in IP multicast. Secondly, we will carry

[1] Ironically, this view was fuelled partly by a report by Sprint[21], when in fact the entire Sprint IP service supports multicast and they have at least 3500 commercial customers streaming content.

out research in ways to distribute an open interface to the multicast tree computation that IP routers implement. The way we propose doing this is through reflection.

Reflection is becoming commonplace in middleware[32] [33] [34], but has not been applied between application level systems and network level entities to our knowledge. The intent here is to offer a common API to both the multicast service, and the filtering service, so that the event notification module implementor need not be aware which layer is implementing a function.

We would envisage an extremely simple API, viz:

```
Create(Subject)
Subscribe/Join(Subject)
Publish/Send(Subject, Content)
Receive(Subject, Content Filter Expression)
```

The router level will create both a real distribution tree for subjects, and a sub-tree for each filter or merged filter set. This will be done with regard to the location (and density) of receivers. It is possible that we can use an multicast tunnel or multicast address translation service such as the one described in[11], to provide further levels of aggregation within the network. This will require the routers to perform approximate tree matching algorithms.

1.1 Solution, and Proposed Experiment

The approach we will take in the work is one of "build and learn". We will build a piece of reflective middleware that is a shim between an existing event notification service and the reflective routing and filter service.

This will involve extending the PGM *signaling* protocol that installs and activates (via IP router alerts) the filters.

We will also investigate efficient hashes for subject to group and content to sequence number mapping.

Subsequently, we aim to evaluate our approach by applying it to a large-scale event driven (sentient) application, such as novel context-aware applications for the emerging UMTS mobile telephony standard[37] or large-scale location tracking applications[38]. For example, there is the possibility of developing a location tracking (people, vehicles and baggage) for large new airport terminals.

2 Overlays and Reflection

As we can see, what we are designing is effectvely a two-tier system, which entails multicast trees, and within these, filters. To these, we believe we have to add a third layer, which is illustrated in figure 1.

The purpose of the overlay is to accomodate a varieity of qualitative heterogeneity, where the lower two layers of multicast and filtering target the area of quantitative performance differences.

Firstly, initial event systems are built without any notion of a multicast filter-capable transport. Thus we must haev an overlay of event distribute servers. These can, where the lower services are available, be programmed to take advantage of it, *amongt themselves*, thus providing a seemless mechanism to deploy the new service transparently to publisher and subscriber systems. However, we also believe that there are *inherent* structural reasons why such an applicaiton layer overly is needed. These include:

Policies. Different regions of the network will have different policies about which events may be published and which not.

Security. There may be firewall or other security mechanisms which impede the distribution via lower level protocols.

Evolution. We would like to accomodate evolution (in the same way that inter-domain routing protocols such as BGP allow intra-domain routing to eveolve).

Interworking. We would like to accomodate multiple event distribution middleware.

Others. There are other such "impedence mismatches" which we may encounter as the system scales up.

A novel aspect of our approach is that the overlay system does not, itself, construct a distribution tree. Isntead, a set of *virtual* members are addd to the lower level distribution system whcih then uses its normal multicast routign algorithms to cnstruct a distributio ntree amongstr a set of event notificaiton servers seperated in islands of multicast capable networks. These servers then use an open interface to quret the routers as to the computed tree, and then use this as their own distribtion - in this way the overlay can take advanatage of detaield metric information that the router layer has access to (such as delay, throughput and current load on links) instead of measuring a poor shadow of that data which would lead to, an inaccurate and out of date parameters with which to build the overlay. In some senses, what we are doign here is like multicast traffic engineering!

We believe that our system provides a number of engineering performance enhancements over previous event notificaiton architectures. Future work will evaluate these, which include:

1. System performance - improvement in scalability, including reduction in join/leave publish/subscribe latency, increase in event throughput, etc.
2. Network impact - impact on router load by filter cost group join, leave and multicast packet forwarding.
3. Expressiveness and seamlessness of API - try it with variety of event notification systems! export via public CVS and see what open source community do?

3 Discussion

For now, its an idea, but we can envisage a world in which perveasive computing devices generate 10,000,000,000 events per second. We can foresee a time when

there are thousands of millions of event subscribers all over the planet, with publishers having popularities as low as no or only a single subscriber, or as high as the entire world.

One of the goals of this system is to explore the way that the multicast treees evolve and the filtering system evolves. Another goal is to see how multicast routign can be "laid open" as a service to be used to build distribution trees for other layers. Fianlyl, we belieev that the three levels we have may not be enough, and that as the system grows larger still, other services may emerge.

It is frequently the case that in the long term, business migrates into the infrastructure. (c.f. voice, IP, etc). We expect many overlay services to do this. We believe that this process by will accelerate due to use of state of the art network, middleware and software engineering approaches. However, this process will not stop - there is an endless stream of new services being introduced "at the top", and makign their way down to the bottom, to emerge as part of the critical information infrastructure.

The architecture is illustrated in figure 1. In this we can see that a publisher creates a sequence of events, which carry attributes with given values. A consumer subscribes to a publisher, and may express content based filters to the publisher. In our system, these filter expressions can be distributed up-stream from the consumer towards the publisher. As they pass through Application-

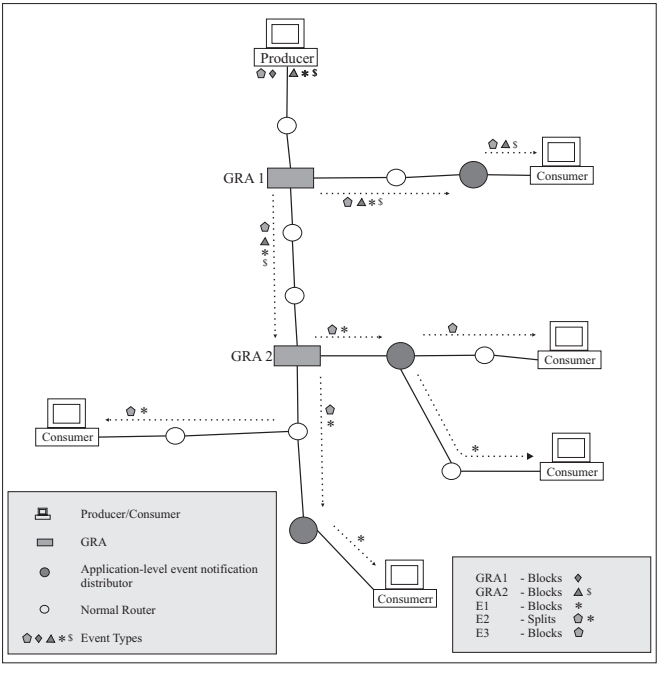

Fig. 1. Channel Islands System Architecture

level event notification distributers, they can be evaluated and compared, and possibly combined with other subscription filters. Notifications of interest are passed up stream all the way to the publisher, or to the application-level cevent notification distributer nearest the publiser, which can then compute a set of fixed tags for data; it can also, by consulting with the IP and GRA routers, through the reflective multicast routing service, compute a set of IP multicast groups over which to distribute the data, which will create the most efficient trade-off between source and network load, and receiver load, as well as tag and filter evaluation, as the events are carried downstream from the publiser, over the IP multicast, GRA, and application-level event notification nodes. Devising and evaluating the detailed performance of the algorithms to carry out these tasks out form the core of the requirements for future work.

Acknowledgements. The author gratefully acknowledges discussions with his colleagues, particularly Jean Bacon and George Coulouris.

References

[1] A. Mankin, A. Romanow, S. Bradner and V. Paxson, "IETF Criteria for Evaluating Reliable Multicast Transport and Application Protocols" RFC2357, June 1998.
[2] Reliable Multicast Research Group `http://www.east.isi.edu/RMRG/`
[3] S.Floyd, V.Jacobson, C.Liu, S.McCanne, L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing, Scalable Reliable Multicast (SRM)", ACM SIGCOMM'95.
[4] M.Handley and J.Crowcroft, "Network Text Editor (NTE): A scalable shared text editor for the Mbone", ACM SIGCOMM'97, Cannes, France, September 1997.
[5] "TCP-like Congestion Control for Layered Multicast Data Transfer", L.Vicisano, L.Rizzo, J.Crowcroft, INFOCOM'98.
[6] "IEEE Standard for Distributed Interactive Simulation - Application Protocols" IEEE std 1278.1-1995, IEEE Computer Society
[7] "IEEE Standard for Distributed Interactive Simulation - Communications Services and Profiles", IEEE std 1278.2-1995, IEEE Computer Society
[8] Mark Handley et al, Building Blocks for Reliable Multicast Transport Protocols, Work in progress, RMT Working Group, IETF.
[9] "Rate Adjustment Protocol" Handley, M. et al Proc Infocom 1999, NY
[10] Pragmatic Generalised Multicast Tony Speakman, et al, Work in Progress, `http://search.ietf.org/internet-drafts/draft-speakman-pgm-spec-07.txt`
[11] "Multicast Address Translation" Work in Progress, `http://www.ietf.org/internet-drafts/draft-crowcroft-mat-00.txt`
[12] "Self Organising Transcoders", Kouvelas, I. et al Proc NOSSDAV 1998, Cambridge England
[13] "Router Mechanisms to Support End-to-End Congestion Control", S.Floyd, K.Fall, Technical report, `ftp://ftp.ee.lbl.gov/papers/collapse.ps`.
[14] "RMTP: A Reliable Multicast Transport Protocol", J.C. Lin, S.Paul, IEEE INFOCOM '96, March 1996, pp.1414-1424.
Available as `ftp://gwen.cs.purdue.edu/pub/lin/rmtp.ps.Z`

[15] "Generic Router Assist Building Block", B. Cain, T. Speakman, D. Towsley, Internet Drafts, Work in progress.
http://search.ietf.org/internet-drafts/draft-ietf-rmt-gra-fspec-00.txt and
http://search.ietf.org/internet-drafts/draft-ietf-rmt-gra-arch-02.txt
[16] GMTS "Generic Multicast Transport Services" B. Cain, D. Towsley, in Proc. Networking 2000, Paris, France May 2000.
http://www.east.isi.edu/RMRG/cain-towsley3/
[17] "Incremental Depoyment of a Router-assisted Relaible Multicast Scheme" C. Papadopoulos, E. Laliotis Proc of NGC 2000 WOrkshop.
[18] "COBEA: A CORBA-Based Event Architecture" C. Ma and J. Bacon Proc of 4th Usenix Conference on Object Oriented Technologies and Systems, 1998
[19] "Building Event Services on Standard Middleware" Jean Bacon, Alexis Hombrecher, Chaoying Ma, Ken Moody, Peter Pietzuch Work in Progress.
[20] TIBCO http://www.tibco.com
[21] "Deployment Issues for the IP Multicast Service and Architecture", C. Diot, B. N. Levine, B. Lyles, H. Kassem, D. Balensiefen. IEEE Network magazine special issue on Multicasting. January/February 2000.
[22] "A Case For End System Multicast", Y. Chu, S. Rao, H. Zhang, Proceedings of ACM SIGMETRICS , Santa Clara,CA, June 2000, pp 1-12.
[23] "Enabling Conferencing Applications on the Internet Using an Overlay Multicast Architecture" Y. Chu, S. Rao, S. Seshan, H. Zhang, Proc. ACM Sigcomm 2001,
http://www.acm.org/sigs/sigcomm/sigcomm2001/p5-chu.pdf
[24] "Overcast: Reliable Multicasting with an Overlay Network", J. Jannotti, D. K. Gifford, K. L. Johnson, M. F. Kaashoek, and J. W. O'Toole, Jr., Proceedings of OSDI'00. http://gaia.cs.umass.edu/cs791n/Jannotti00.pdf
[25] "Tapestry: a fault tolerant wide area network infrastructure", B. Zhou, D. A. Joseph, J. Kubiatowicz, Sigcomm 2001 poster and UC Berkeley Tech. Report UCB/CSD-01-1141.
http://www.cs.berkeley.edu/ ravenben/publications/CSD-01-1141.pdf
[26] "Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications" I. Stoica, R. Morris, D. Karger, F. Kaashoek, H. Balakrishnan, ACM Sigcomm2001,
http://www.acm.org/sigcomm/sigcomm2001/p12.html
[27] S. Ratnasamy, P. Francis, M. Handley, R. Karp, S. Shenker, "A Scalable Content-Addressable Network" ACM Sigcomm 2001,
http://www.acm.org/sigcomm/sigcomm2001/p13.html
[28] "Application-Level Anycasting: a Server Selection Architecture and Use in a Replicated Web Service" E. Zegura, M. Ammar, Z. Fei, and S. Bhattacharjee. IEEE/ACM Transactions on Networking, Aug. 2000.
ftp://ftp.cs.umd.edu/pub/bobby/publications/anycast-ToN-2000.ps.gz
[29] "Evaluation of a Novel Two-Step Server Selection", K. M. Hanna, N. Natarajan, and B.N. Levine, Metric To Appear in IEEE ICNP 2001. November 2001.
http://www.cs.umass.edu/ hanna/papers/icnp01.ps
[30] "Finding Close Friends on the Internet" Christopher Kommareddy, Narendar Shankar, Bobby Bhattacharjee, To appear in ICNP 2001.
[31] "An Investigation of Geographic Mapping Techniques for Internet Hosts" Venkata N. Padmanabhan, Lakshminarayanan Subramanian, Proc of ACM SIGCOMM 2001, San Dieogo, 2001. http://www.acm.org/sigcomm/sigcomm2001/p14.html

[32] "Integrating Meta-Information Management and Reflection in Middleware", Fabio Costa and Gordon Blair 2nd International Symposium on Distributed Objects & Applications pp. 133-143, Antwerp, Belgium, Sept. 21-23, 2000. Internal report number MPG-00-20

[33] "The Role of Open Implementation and Reflection in Supporting Mobile Applications " Gordon Blair Proceedings of the IEEE Workshop on Mobility in Databases and Distributed Systems (MDDS'98), Vienna, August 1998. Internal report number MPG-98-35.

[34] "Open Implementation and Flexibility in CSCW Toolkits", Paul Dourish, PhD Thesis, 1996, Supervisor, Jon Crowcroft Available from `ftp://cs.ucl.ac.uk/darpa/dourish-thesis.ps.gz`

[35] "A Language-Based Approach to Programmable Networks", Ian Wakeman, Alan Jeffrey and Tim Owen, IEEE Conference on Open Architectures and network Programming, March 2000, Tel-Aviv, Israel.

[36] What is Reflective Middleware? Geoff Coulson http://computer.org/dsonline/middleware/RMarticle1.htm

[37] "UMTS Networks: Architecture, Mobility and Services", Wiley & Sons. 2001; ISBN: 047148654X, Heikki Kaaranen (Editor), Siamäk Naghian, Lauri Laitinen, Ari Ahtiainen, Valtteri Niemi

[38] The Graticule System `http://www.graticule.com/products/MapGPS.html`

[39] "A Survey of Event System", A. Rifkin and R. Khare. `http://www.cs.caltech.edu/ adam/isen/event-systems.html`

[40] "Notification Service Specification", Object Management Group, June 2000, `ftp://ftp.omg.org/pub/docs/formal/00-06-20.pdf`

[41] "Design and evaluation of a wide-area event notification service", Carzaniga A., Rosenblum D. S. and Wolf A. L. ACM Transactions on Computer Systems, Volume 19, no. 3, pp. 332-383, 2001