To appear in Proceedings of the NSF Workshop on an Infrastructure for Mobile and Wireless Systems.
Lecture Notes in Computer Science, Springer-Verlag.

Content-Based Networking: A New Communication Infrastructure

Antonio Carzaniga and Alexander L. Wolf

Department of Computer Science
University of Colorado
Boulder, CO 80309-0430 USA
{carzanig,alw}@cs.colorado.edu

Abstract. We argue that the needs of many classes of modern applications, especially those targeted at mobile or wireless computing, demand the services of content-based publish/subscribe middleware, and that this middleware in turn demands a new kind of communication infrastructure for its proper implementation. We refer to this new communication infrastructure as *content-based networking*. The service model of this network must directly support the interface of an advanced content-based publish/subscribe middleware service. At the same time, the implementation must be architected as a true distributed network, providing appropriate guarantees of reliability, security, and performance. We do not propose content-based networking as a replacement for IP, nor do we advocate an implementation of a publish/subscribe middleware at the network level (i.e., within routers). Instead, we argue that content-based networking must be designed according to established networking principles and techniques. To this end, in this paper, we formulate the foundational concepts of content-based networking, and relate them to the corresponding concepts in traditional networking. We also briefly review our experience with content-based publish/subscribe middleware and suggest some open research problems in the area of content-based networking.

1 Introduction

Distributed auction systems, such as e-Bay$^{\mathrm{TM}}$, and information sharing systems, such as Napster$^{\mathrm{TM}}$ and Gnutella, are classes of network applications that have had an impressive growth in popularity over the past few years. Other examples are applications especially suited to mobile and wireless computing platforms, such as instant messaging, personalized news distribution, and service discovery. E-bay implements a traditional auction system, where sellers advertise items for sale and where buyers can bid on selected items. Napster and Gnutella allow users to share files (usually multimedia content) by providing a search capability based on file-name match and possibly content type. Instant messaging allows users to participate in real-time, one-to-one or many-to-many discussions. Personalized news distribution allows users to receive announcements and updates regarding news of interest (e.g., stock quotes for a specific portfolio, weather reports for a specific geographic area, or scores for a favorite sports team). Service discovery is the process of gaining access to a specific computing service or device (e.g., a printer or a fax machine) based on properties, such as the type of service, its physical location, its capabilities, or its cost.

What all these applications have in common is a new style of communication, whereby the flow of information—from senders to receivers—is determined by the specific interests of the receiver rather than by an explicit destination address assigned by the sender. With this communication pattern, receivers subscribe to information that is of interest to them without regard to any specific source (unless that is one of the selection criteria), while senders simply publish information without addressing it to any specific destination.

One approach to supporting this new communication style is to use a multicast network service, which provides some level of mediation between senders and receivers, allowing them to communicate through *virtual* (multicast) addresses. For example, a personalized sportscast application could be implemented on top of a multicast network by associating with each team a separate multicast address. Receivers interested in knowing about a team would join the multicast group associated with that team, while information sources would send to that group the datagrams containing information relevant to the team.

Unfortunately, the selection capability offered by multicast addresses is limited and in certain situations is simply not adequate for the kinds of applications mentioned above. In fact, the sportscast example would work as long as a receiver is interested in *everything* regarding only *one* team. Obviously, for every individual receiver's interests, there exists a mapping of information to multicast addresses that satisfies that receiver. However, there is no general mapping that satisfies all receivers. As it turns out, there are two opposing strategies for associating multicast addresses to receivers' interests: one could either define many specific multicast groups or define a few generic groups. Both solutions have significant limitations. With specific groups, receivers would be able to select information of interest with high accuracy, but at the same time senders would be forced to send to multiple groups whenever they produce information that spans multiple specific selections, thus defeating the main purpose of multicast. Moreover, the multicast routing infrastructure would have a hard time efficiently serving a large set of very sparse groups, and would have a hard time dealing with highly dynamic changes in interest that would lead to highly dynamic restructurings of the groups. The case of a few generic groups has the opposite advantages and disadvantages: senders could send to a few groups and the multicast routing infrastructure would benefit from a lower number of dense groups, but receivers would receive, and therefore would have to process (i.e., filter out), a large volume of uninteresting information.

Another approach to supporting this new communication style is to implement what amounts to an application-level information broker with a rich information selection capability. Such a system is referred to as a *content-based publish/subscribe* middleware service [8, 12]. The term "content-based" characterizes those systems whose subscriptions can express predicates over the whole content of a publication. This is in contrast with *channel-based* and *subject-based* systems, in which only a few well-known attributes of a publication are available for selection to subscriptions. The strength of content-based publish/subscribe middleware over a multicast network service is the greater expressive power of its data model and of its subscription language. Its weakness is scalability. In fact, only a few content-based publish/subscribe middleware services are implemented as true distributed systems, and none of the existing ones is designed to achieve levels of scalability comparable to those of existing network communication infrastructures, such as IP.

We argue that the needs of many classes of modern applications, especially those targeted at mobile or wireless computing, demand the services of content-based publish/subscribe middleware, and that this middleware in turn demands a new kind of communication infrastructure for its proper implementation. We refer to this new communication infrastructure as *content-based networking* [7]. The service model of this network must directly support the interface of an advanced content-based publish/subscribe middleware service. At the same time, the implementation must be architected as a true distributed network, providing appropriate guarantees of reliability, security, and performance.

Note that content-based networking is not intended as a replacement for IP or other traditional unicast or multicast services. Rather it is intended to implement the specific communication style typified by publish/subscribe middleware services in a way that is superior to current approaches. Note also that we are not advocating the implementation of content-based networking at the network level itself (i.e., at level "3"). While there is no conceptual obstacle to doing so, it is not yet clear whether it would be better (or even feasible) from an engineering standpoint. For practical reasons, initial prototypes should be implemented as application-level networks, most likely on top of existing Internet protocols, much the way that implementations of multicast were first developed (and where final implementations may end up [9]). Nevertheless, we argue that many established networking techniques can and should be adapted more or less directly to content-based networking where appropriate.

In this paper we formulate the foundational concepts of content-based networking, and relate them to the corresponding concepts in traditional networking. We also briefly review our experience with content-based publish/subscribe middleware and suggest some open research problems in the area of content-based networking.

2 Background

Content-based networking is an evolution of our work on distributed publish/subscribe event notification systems, particularly of our project Siena [4, 6, 8]. With Siena, we formulated a service interface and semantics that achieves a good balance of expressiveness (in the data model and subscription language) and scalability. To our knowledge, Siena is the first work that applies a form of subnet and supernet address relations (see Section 3.2), which we call *covering relations*, to optimize the routing function in a distributed content-based publish/subscribe system. In our studies, we have also introduced an event-matching algorithm integrated with the same data structures used by the routing algorithms.

For the implementation architecture of Siena, we studied both a hierarchical and a peer-to-peer arrangement of servers. We then combined these topologies with two classes of routing algorithms. We evaluated the combinations of topologies and algorithms by means of simulated network scenarios, populated by synthetic parameterized applications. With these simulations, we were able to characterize which topologies and algorithms are more sensitive to which behavioral factors in applications.

The idea of content-based networking is related to both distributed content-based publish/subscribe systems and network technology. We present an analysis of several of those technologies and classify them according to their architectures and to their service models elsewhere [4, 8]. Our analysis shows that Siena is currently the only system that combines a rich content-based subscription language with a truly distributed implementation architecture.

3 Model of Content-Based Networking

At the physical-architecture level, a content-based network is identical to a traditional network. We can think of it as a simple graph. Nodes of the graph are *hosts* or *routers*, arcs are direct communication links. For simplicity, we assume that all connections are bi-directional, and therefore our architectural model is a non-directed graph. We also assume that the graph is connected. Hosts are nodes that have exactly one link, while nodes with more than one link act as routers. In reality, subsets of nodes may be directly connected to each other in *subnetworks*, thereby forming complete subgraphs, that in turn are connected to each other via routers. For simplicity, we ignore the internals of subnetworks and we model them as single router nodes in our graph.

It is the mode of communication, or *service model*, in content-based networking that differs significantly from traditional (unicast or multicast) networking. In a content-based network, nodes are not assigned any unique network address, nor are datagrams addressed to any specific node or node group. Instead, each node advertises a *receiver predicate* (or *r-predicate*) that defines datagrams of interest for that node and, thus, the datagrams that the node intends to receive. Nodes can also send out datagrams, which the network will forward to all the nodes with matching r-predicates. A node may also advertise a *sender predicate* (or *s-predicate*). An s-predicate defines the datagrams that a node intends to send.

This service model is generic with respect to the form of datagrams and predicates. We denote $\mathcal{D}$ the universe of datagrams, and $\mathcal{P} : \mathcal{D} \rightarrow \{true, false\}$ the universe of predicates over D. We say that $\mathcal{P}$ and $\mathcal{D}$ define a content-based addressing scheme, which in turn defines the content-based network. Consistently we say that the r-predicate p_n advertised by n is the content-based address of the node n. (We elaborate on this definition in Section 3.2.) We also say that a datagram d is implicitly addressed by its content (or *cb-addressed*) to a node n with content-based address p_n if $p_n(d) = true$.

A specific content-based network must instantiate its own content-based addressing scheme ($\mathcal{D}$ and $\mathcal{P}$). In practice, the designer of a content-based network will define the following two models.

- *Datagram model:* a schema for datagrams that defines the kind of information that may be expressed within datagrams, and the encoding of that information.
- *Predicate model:* the syntax and semantics for predicates, where r-predicates and s-predicates may conform to two different models.

Plausible examples of datagram models are the format specification for Internet e-mail messages (RFC 822) or the specification of IP datagrams (RFC 791). A datagram model may prescribe a rigid structure, such as that of the header section of an IP datagram, or a flexible structure, such as that of the header section of an Internet e-mail message, or even free-form content. At an extreme, a datagram model can be as generic as a sequence of octets (or even just bits). Closely related to the datagram model is the predicate model, which defines a class of boolean functions operating over the datagram model. Assuming, for example, datagrams in the form of an IP packet, a valid predicate model might define the class of functions

$$p_k(d) = \begin{cases} true & \text{if } k = destination\_address(d) \\ false & \text{if } k \neq destination\_address(d) \end{cases}$$

At an extreme, the predicate model $\mathcal{P}$ can be as generic as the set of computable boolean functions over $\mathcal{D}$. Again, assuming datagrams in the form of IP packets, it would be possible, although probably unreasonable, to allow predicates such as

$$prime(d) = \begin{cases} true & \text{if } payload(d) \text{ is a prime number} \\ false & \text{if } payload(d) \text{ is a composite number} \end{cases}$$

or

$$cc(d) = \begin{cases} true & \text{if } payload(d) \text{ is a valid C program} \\ false & \text{otherwise} \end{cases}$$

Clearly, the choice of data and predicate models is driven in opposite directions by the demands of expressiveness and scalability [6].

Below we focus on the generic content-based network model, within which we define fundamental concepts such as content-based forwarding and routing, and content-based subnetting and supernetting. We present specific data and predicate models in other documents [6, 8].

3.1 Content-Based Forwarding and Routing

Similar to a traditional network, the semantics of a content-based addressing scheme is realized by the *forwarding* and *routing* functions performed by routers:

forwarding: the router decides where to forward incoming datagrams. A datagram is output to a subset of the router's adjacent nodes. The router computes the output set based on the datagram content and on its internal *forwarding table*;

routing: the router compiles and maintains its forwarding table. In order to do that, the router gathers, combines, and exchanges predicates and possibly other routing information with adjacent nodes.

The forwarding table is the materialization of a map between interfaces to adjacent nodes and r-predicates:

$$FwdTable : I \to P$$

Performing the forwarding function for an incoming datagram d amounts to computing the set of interfaces:

$$forward(d) = \{i \in I : d \quad matches \quad FwdTable(i)\}$$

As in traditional networks, the scope of the forwarding function is localized to each router. Also, the throughput of the forwarding function determines the throughput of a router. It is therefore crucial that the forwarding function be computed very efficiently. The applicable optimization techniques can be classified in the following general groups:

- fast matching algorithms;
- compact representations of *FwdTable*, amenable to manipulation by fast matching algorithms;
- reduction of the size of *FwdTable*; and
- reduction of the unnecessary traffic of datagrams flowing through the router.

Notice that the first two techniques implement pure forwarding optimizations, while the last two are essentially routing strategies that yield efficient forwarding.

Again similar to a traditional network, the scope of the routing function is the whole content-based network. In fact, each router must compile its forwarding table in such a way that the combined effect of the forwarding function on all routers is consistent with the semantics of the content-based network—that is, to deliver every datagram to all its implicitly addressed nodes, possibly using optimal paths. Perhaps the most important difference between content-based routing and traditional routing is in the assumption about the volatility of addresses. In traditional networks, addresses and topology are supposed to be fairly stable, meaning that the assignment of addresses to interfaces (or subnetwork addresses to groups of interfaces) and the distances between them changes at a rate that is several orders of magnitude lower than the rate at which datagrams flow through the network. This assumption might not hold in content-based networking, since content-based addresses are determined directly by user-defined predicates. Notice that this issue in content-based networks is complementary to mobility in mobile and wireless computing, in which addresses remain stable, but instead the topology changes in direct response to application behavior (e.g., a device moving from one area to another).

3.2 Subnetting, Supernetting, and Routing

It is fair to say that traditional networking technology scales well due in no small measure to hierarchical routing, which means in essence that a router can treat a subnet as a single entity. A subnet is a cluster of nodes with similar addresses, or in other words, a set of nodes topologically close to each other, with addresses close to each other in their address space. When propagating routing information, routers try to combine subnets into bigger subnets (supernetting). Obviously, this process makes sense as long as the representation of the supernet addresses is more efficient to store and to process than the list of addresses of its components. We believe that the same general principles of subnetting and supernetting must be applied to content-based routing.

Before we proceed with the definition of content-based subnetting and supernetting, we must review and extend our initial definition of content-based address. According to that definition, the content-based address of a node n is its r-predicate p_n. This might be ambiguous, however, since p_n could be confused with a *representation* of p_n within an actual addressing scheme. A representation of a predicate is its operational or declarative specification in a predicate language within a specific addressing schema. Notice that a predicate language might allow several different representations for the same logical predicate p. Therefore, in order to disambiguate the above definition in the discussions to follow, we distinguish predicates, representations of a predicate within some datagram and predicate model, and the set of datagrams defined by a predicate. The notation is as follows:

- p is an abstract predicate,
- p_n is the r-predicate advertised by node n,
- $p'_n, p''_n, \ldots$ are specific representations of p_n, and
- (p) is the set of datagrams matched by a predicate p, which means that $(p) = \{d | p(d) = \mathit{true}\}$, and that $(p) = (p') = (p'') = \ldots$

Without loss of precision we can use the set of datagrams (p_n) or alternatively the predicate p_n to denote the content-based address of a node n that advertises an r-predicate p_n.

The concepts of subnet address and supernet address follow immediately from the definition of content-based address. In fact, the subnet (or supernet) relation corresponds to the subset (or superset) relation between content-based addresses. Specifically, p is a *content-based subnet* of q, and q is a *content-based supernet* of p if $(p) \subseteq (q)$. In practice, content-based routers should attempt to combine content-based addresses of topologically-related nodes into *content-based supernet addresses*, with the obvious requirement that content-based supernet addresses be more efficiently stored and processed than the list of their sub-addresses.

4 Open Issues

Fully developing the concept of content-based networking in terms of its relationship to traditional networking, as we have begun to do in the previous section, is a fundamental focus of the research into this new communication infrastructure. But there are other, equally important issues to explore. We conclude this paper with a brief review of some of those issues.

Forwarding and Routing. The core research in content-based networking should focus on content-based forwarding and routing algorithms. Several algorithms have been proposed for the forwarding function in the context of publish/subscribe middleware systems [1, 3, 8, 10, 11]. We have recently developed an algorithm for content-based forwarding designed specifically for networked content-based routers [5]. Few research groups have considered specific optimizations of content-based routing [2, 8]. We believe that more work needs to be done, especially on the routing front, and also on the combination of the two functions. We also believe that all these efforts need to be integrated with solutions to common networking problems. One example is timing in the routing protocols. The proposed protocols do not detail this aspect, and therefore one should assume that routing information is propagated as it becomes available. Unfortunately, this strategy is known to cause a snowball effect on networks, and to incur dangerous congestions. Common practice in traditional routing protocols suggests that content-based routing information should be propagated using a heartbeat-type protocol. Other related issues are the stability and reliability of the content-based routing protocol. While traditional approaches can serve as valuable guidelines, it is unlikely that they can be directly applicable to content-based routing. In fact, the high volatility of address bindings (i.e., subscriptions) would probably require specific protocol features.

Reliable Transport Layer. So far we have discussed the content-based networking service model under the implicit assumption that it is a best-effort service. While we believe this to be the right assumption, it is not clear how a reliable "transport" layer can be implemented on top of the content-based network layer. Some ideas may be adapted from reliable multicast protocols. However, because the content-based service model lacks the concept of (unicast or multicast) channel, it is not clear what a content-based transport service should provide.

Security. Another essential point in the research agenda for content-based networking is security. We believe that traditional methods and techniques can be applied to content-based networking, but the nature of the content-based network service would highlight specific security goals [13]. In particular, because there is no concept of intended receiver, privacy of datagrams would be somehow less important than in traditional networks. On the other hand, it may be more important to protect the privacy of receivers by guaranteeing some form of confidentiality or anonymity of predicates.

Benchmarks. Orthogonal to all of the issues discussed so far is the issue of evaluating protocols and algorithms against their requirements of scalability, robustness, congestion control, and the like. We believe that simulation will be the primary evaluation and validation tool, and have used this technique in our own work [4, 8]. The difficulty is finding representative and credible workloads to drive the simulations. These workloads will likely derive from both existing applications, such as Napster or Gnutella, and from synthetic applications that can be argued as embodying the new demands on the communication infrastructure imposed by future applications. Agreement on a suite of such workloads will lead to benchmarks for evaluating solutions in this area.

Acknowledgements

The authors would like to thank David Rosenblum for the numerous discussions that helped shape and refine the ideas presented in this paper. The work of A. Carzaniga and A.L. Wolf was supported in part by the Defense Advanced Research Projects Agency, Air Force Research Laboratory, Space and Naval Warfare System Center, and Army Research Office under agreement numbers F30602-01-1-0503, F30602-00-2-0608, N66001-00-1-8945, and DAAD19-01-1-0484. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency, Air Force Research Laboratory, Space and Naval Warfare System Center, Army Research Office, or the U.S. Government.

References

1. M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra. Matching events in a content-based subscription system. In *Eighteenth ACM Symposium on Principles of Distributed Computing (PODC '99)*, pages 53–61, Atlanta, Georgia, May 4–6 1999.
2. G. Banavar, T. D. Chandra, B. Mukherjee, J. Nagarajarao, R. E. Strom, and D. C. Sturman. An efficient multicast protocol for content-based publish-subscribe systems. In *The 19^{th} IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, pages 262–272, Austin, Texas, May 1999.

3. A. Campailla, S. Chaki, E. Clarke, S. Jha, and H. Veith. Efficient filtering in publish-subscribe systems using binary decision diagrams. In *Proceedings of the 23th International Conference on Software Engineering*, pages 443–452, Toronto, Canada, May 2001.
4. A. Carzaniga. *Architectures for an Event Notification Service Scalable to Wide-area Networks*. PhD thesis, Politecnico di Milano, Milano, Italy, Dec. 1998.
5. A. Carzaniga, J. Deng, and A. L. Wolf. Fast forwarding for content-based networking. Technical Report CU-CS-922-01, Department of Computer Science, University of Colorado, Nov. 2001.
6. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Achieving scalability and expressiveness in an internet-scale event notification service. In *Proceedings of the Nineteenth ACM Symposium on Principles of Distributed Computing (PODC 2000)*, pages 219–227, Portland, Oregon, July 2000.
7. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Content-based addressing and routing: A general model and its application. Technical Report CU-CS-902-00, Department of Computer Science, University of Colorado, Jan. 2000.
8. A. Carzaniga, D. S. Rosenblum, and A. L. Wolf. Design and evaluation of a wide-area event notification service. *ACM Transactions on Computer Systems*, 19(3):332–383, Aug. 2001.
9. Y. Chu, S. G. Rao, and H. Zhang. A case for end system multicast. In *Proceedings of ACM Sigmetrics*, pages 1–12, Santa Clara, California, June 2000.
10. F. Fabret, H. A. Jacobsen, F. Llirbat, J. Pereira, K. A. Ross, and D. Shasha. Filtering algorithms and implementation for very fast publish/subscribe systems. In *ACM SIGMOD 2001*, pages 115–126, Santa Barbara, California, May 2001.
11. J. Gough and G. Smith. Efficient recognition of events in a distributed system. In *Proceedings of the 18th Australasian Computer Science Conference*, Adelaide, Australia, Feb. 1995.
12. D. S. Rosenblum and A. L. Wolf. A design framework for Internet-scale event observation and notification. In *Proceedings of the Sixth European Software Engineering Conference*, number 1301 in Lecture Notes in Computer Science, pages 344–360. Springer–Verlag, 1997.
13. C. Wang, A. Carzaniga, D. Evans, and A. L. Wolf. Security issues and requirements for Internet-scale publish-subscribe systems. In *Proceedings of the Thirty-fifth Hawaii International Conference on System Sciences (HICSS-35)*, Big Island, Hawaii, Jan. 2002.