Content-Based Communication: a Research Agenda*

Antonio Carzaniga[†,‡] Cyrus P. Hall[†]

[†]Faculty of Informatics
University of Lugano
6900 Lugano, Switzerland

[‡]Department of Computer Science
University of Colorado
Boulder, Colorado 80309-0430

ABSTRACT

A content-based publish/subscribe system is a message-oriented communication facility based on the idea of interest-driven routing. A message, published by the sender without a set destination, is delivered to any and all the receivers that expressed an interest in its content. We refer to this communication style and to the distributed infrastructure that realizes it as *content-based communication* and *content-based networking*, respectively. In this paper we review what we consider the foundations of content-based networking, including some of the major advances of the past years. We then present a vision for further research in this area as well as for the practical realization of a content-based network. In particular, we discuss the implications of content-based communication for the network, the middleware, and applications.

Keywords

Distributed publish/subscribe systems, content-based communication, content-based networking.

1. INTRODUCTION

The publish/subscribe communication style is based on the idea that receivers *subscribe* to information of interest, that is, they specify the information they intend to receive, while senders simply *publish* information without addressing it to any particular receiver. A third-party communication system, often called broker or dispatcher, is then responsible for transmitting published information to all interested receivers. This communication style is also often referred to as "event notification," whereby published information is assumed to describe "events." Consistently, applications that use this communication style are also said to conform to an event-based architecture, and are also referred to as event-based applications.

In this paper, as in the research we have conducted in the past years, we ignore to a large extent both the nature of the information being transmitted and the nature and architecture of the applications that produce and consume such information. Instead, we focus on the novel *mode of communication* embodied in publish/subscribe communication, and on the design and implementation of a communication system that realizes this mode of communication.

Our first fundamental observation regarding publish/subscribe communication is that it is very similar to the communication style of standard network services such as IP multicast. In fact, subscribing is equivalent to joining a multicast group, and publishing is equivalent to sending to a multicast group. The only difference is that the addressing scheme of IP multicast defines what amounts to a *partition* of the space of datagram, while modern publish/subscribe systems are generally assumed to be *content-based* systems that allow expressive conditions over the content of publications. This in turn allows subscriptions to define *arbitrary subsets* of the space of publications.

To exemplify this difference, consider a simplistic highway traffic control system consisting of traffic sensors and supporting private motorists as well as public-safety patrols. East-bound and West-bound motorists would be interested in receiving accident reports for their respective directions, while highway patrols would want to receive both types of alerts. With IP multicast, the designer of the system would have to decide between a large and a small granularity of multicast groups. With large groups, all alerts would be sent once to a single group, motorists and patrols would join this group, and motorists would have to filter out alerts regarding the opposite direction. With small groups, East-bound motorists would join a group for alerts in the East direction, etc., while patrols would join both groups, and alerts would have to be sent to both groups. The first solution favors the multicast infrastructure at the expense of applications. The second solution is better for applications, but puts a significant burden on the multicast infrastructure. In a content-based publish/subscribe system, this problem would have a much simpler solution: traffic sensors would publish once, and applications would also subscribe once with their specific and possibly overlapping interests.

In essence, we argue that publish/subscribe communication is not special because of its publish/subscribe nature, but rather because of its *content-based addressing scheme*,

*Research supported in part by the US National Science Foundation, US Army Research Office, European Commission, and Swiss National Fund under agreement numbers ANI-0240412, DAAD19-01-1-0484, IST-026955, and 200021-109562.

SEM'06, November 10, 2006, Portland, Oregon, USA.

which is more expressive than the "channel-based" scheme used by IP multicast.

This observation lead us to adopt and promote a new view and a new terminology. First of all, we view a publish/subscribe system as a communication system. Therefore we drop the terms "publishers," "subscribers," and "notifications" or "events." Instead we talk about *messages*, *senders*, and *receivers*. We then define the interface of this communication service by explicitly referring to its core functionality, which is to allow receivers to declare their interests in terms of arbitrary sets of messages. Thus, we define an interface consisting of two main functions: *send-message(msg)*, which allows a sender to send a message, and *set-predicate(p)*, which allows a receiver to declare the set of messages the receiver is interested in. A predicate p is a total Boolean function defined on the universe of messages, so the set of messages of interest is $\{m : p(m)\}$. We refer to the mode of communication defined by this interface as *content-based communication*.

Having defined content-based communication in terms of its service interface, we focus on its realization as a distributed system in which the transmission of messages is controlled by a more or less decentralized algorithm. In this context, we adopt a view that once again is inspired by modern communication networks. Our second observation is to view a distributed implementation of a content-based communication service (and a distributed content-based publish/subscribe system) exactly as a packet-switched network. Thus, we refer to a distributed implementation of a content-based communication service as a *content-based network*.

In particular, we assume an interconnection of routers, where the connections between routers may be physical links or overlay links (this distinction is inessential at this point, although it is clearly important when designing an actual system). We then call *content-based routing* the distributed algorithm that collects, propagates, assembles, and transforms receiver predicates as well as topological information (e.g., link costs) to then compile router-local forwarding functions. We call *content-based forwarding* the router-local algorithm that, based on the information established by the routing algorithm, processes incoming messages to decide to which next-hop routers or to which local applications to pass the message.

Admittedly, the vision of a content-based network may not be so profound in and of itself. Nevertheless, we believe that it helps to structure our research efforts in this area, and to relate them to the numerous relevant results available from the networking literature.

2. RESEARCH QUESTIONS

The following sections describe an unsorted list of open problems and ideas, which we believe to be interesting and important in the area of content-based networking.

2.1 Basic Interface

The idea of content-based communication rests on the assumption that senders and receivers "connect" by somehow speaking the same language. More specifically, senders and receivers must agree on the syntax and semantics of the messages they send and of the predicates they declare. Otherwise, senders might not communicate to receivers when they should, or they might also communicate when they should not. For example, a traffic sensor application might send [alert=traffic, level=intense] while a control application might request [application=traffic, alert=congestion, intensity>2]. In this case, the two applications will not communicate. Notice that what is important in this case is not the low-level syntax of messages. Obviously, that must also be agreed upon, but we view that as a minor engineering issue that mounts to agreeing on a basic external representation format.

The essence of this problem goes well beyond content-based communication, and beyond information systems in general. Notice in fact that this problem is analogous to that of finding addresses in traditional address-based networks. Networks typically provide a look-up service (e.g., DNS in IP networks) whereby addresses can be searched for by name, which helps because names are more easily remembered and communicated by human beings. However, a look-up system does not solve the essential problem of *finding* addresses, which is simply shifted from the space of addresses to the space of names.

The most common approach to solve this problem is to make it possible and convenient for senders and receives to agree on terms (semantics) and structure (syntax), but then leave the burden of the actual agreement to higher-level systems or protocols. We believe that this remains the most sensible approach. However, we notice that current systems implement this approach in such a way that senders and receivers *share* this responsibility. In other words, both senders and receivers must make an effort to reach a midway point of contact.

More concretely, consider the interface of current content-based communication systems. All the systems we know of define a structure for messages and a predicate language that is based on that structure. For example, some systems structure messages as XML documents, and use selection predicates based on XPath expressions. Other systems structure messages as tuples of named attributes, and use SQL-like expressions for predicates. In the first case, senders must choose a specific XML schema (or simply a set of tag names) and receivers must write XPath expressions based on the same schema (or tags). Similarly in the second case, senders must choose attribute names, and receivers must refer to the same attribute names in their predicates.

Our view is that this "meet-in-the-middle" approach aggravates the problem, and that a better approach is to shift the responsibility for an agreement almost completely towards receivers. Specifically, we propose to remove constraining message structures, giving senders maximum autonomy as to how they express information, while at the same time giving receivers an expressive language to select messages over the potentially many ways that senders express themselves.

We have recently started working on this new approach, and we will present here some of our initial results. However, before doing that, we should clarify our position with respect to the risks of this approach. As we have argued in the past [4], there are good reasons to structure messages and to limit the expressiveness of predicates. One is that message structures and simple predicates allow for efficient implementations of the matching function, which is a crucial component in content-based forwarding. Another benefit of having a limited form of predicates is that routers can reason about predicates, and can combine, approximate, and index them to make the routing problem tractable on large-scale networks. So, one might wonder whether we are now proposing to ignore the complexities of forwarding and routing. This is not what we are proposing. Instead, our intend is to revisit the trade-offs between expressiveness and scalability in an attempt to facilitate content-based communication.

Our first step in this direction was to experiment with predicate languages that include regular expressions, and in particular we chose and refined a regular expression language that is amenable to matching algorithms that are fast in the most common cases. Specifically in terms of predicate language, we started from the concrete case of the Siena Fast Forwarding algorithm (SFF), which implements a typical predicate language based on disjunctions of conjunctions of attribute-constraints, and we explored three ways to use regular expressions.[1]

- *Regular expressions over attribute values:* regular expressions are simply added as a new operator for constraints over strings. For example, in a web caching application, a receiver could request updates to all the URLs matching a given expression. For example, [url=$^{\text{regex}}$`/php(-[0-9.]*)?\.ini$/`].
- *Regular expressions to match attribute names:* a constraint could refer to the variety of attributes matching a given regular expression. This case is specifically intended to improve over the rigid structure defined by attribute names in predicates.
- *Regular expressions over flat messages:* taking the idea of regular expressions in content-based communication to the extreme, we could have free-text and use regular expressions over that whole, flat message content.

These variations over the predicate language would require more or less radical changes to the core infrastructure. The first extension requires the implementation of a regular expression matcher, but does not require any changes in the routing or forwarding algorithms. In fact, we have already implemented the matcher in a recent version of our Siena Fast Forwarding module. Our implementation uses a modified non-deterministic finite-state automaton (NFA). All regular expressions in the system are composed into a single NFA using an initial ε transition. This data structure has proved to be quite fast, forwarding upwards of 15 MB/s on many real-world workloads (system monitoring and spam filtering). We have not considered the implications to the routing layer at this time.

In the other two cases, both routing and forwarding algorithms would have to undergo significant changes, and possibly be redesigned completely.

Going even further with the idea of less constraining messages and more expressive predicates, we also envision a content-based communication model where predicates are *probabilistic* in nature. Apart from the potential increase in complexity for the forwarding function, this idea seems to be in contrast with much of the current work on content-based forwarding, where false-positives are generally assumed to be undesirable. While we do not directly disagree with this view, we believe that content-based networking is also applicable in areas where the concept of matching a predicate is fuzzy at best. One clear motivating example of such a space is the number booming social networking sites. Imagine a user attempting to find a date via an on-line dating service. They enter information detailing their ideal partner (a predicate if you will), but yet they would never expect an exact "match," least they never go on date. Instead they are looking for "fuzzy" matches, including those users who *generally fit* their profile.

Many algorithms have been developed to implement various form of probabilistic pattern-matching functions. For example, many common information retrieval techniques are based on term vectors combined with natural-language processing. However, most of these techniques focus on the training and creation of models, and are not amenable to indexing within a forwarding engine. It might be interesting to study specific algorithms or classes of algorithms that implement probabilistic matchers and that can be executed on numerous patterns with sublinear complexity in the number of patterns.

Note that probabilistic matching by itself does not eliminate the problem of senders and receivers agreeing on message structure and terminology. Nevertheless, several probabilistic matching algorithms are themselves tolerant to data sets in which attribute names and/or positions change (e.g., connectionist approaches to recognition). This separation of concerns, where the probabilistic matching algorithm deals with small semantics inconsistencies and the content-based matcher deals with the general and wider search space, may in fact lead to better and more flexible content-based systems.

We envision routing in content-based networks with probabilistic matching to look radically different than existing networks. Whereas current content-based networks are either hierarchical or based on coverage properties, routing in the presence of probabilistic matching could and probably should be based on different properties. One such property we are exploring is a measure of *proximity* between interests. Specifically, almost all classifiers can output a form of distance between two inputs, in this case the predicate and a message, or two predicates. By comparing predicates to each other we can structure networks based on links of varying distance. We are currently developing a system that implements such a routing layer using a distributed clustering algorithm specifically targeted at small-world networks.

2.2 Routing: Internetworking and Policies

The idea of a global, planet-wide content-based network is very appealing. Like other researchers, we have envisioned such a large-scale system as an open network, characterized by standard protocols and also by uniform and generally permissive access and transit policies. This idea has some merits, if nothing else for the views that fueled the development of other large scale networks such as the Internet, or even the telephone network. However, if we take a look at the evolution of those systems, we can not fail to notice that large networks tend to evolve naturally into hierarchical systems made up of autonomous systems, which in turn tend to create barriers for communication.

We believe that these subdivisions, grouped and interconnected more or less according to natural geographic and social boundaries, are unavoidable and to some extent are even desirable. Our conclusion is therefore that a scalable design for a content-based network must include effective methods to establish and to enforce *policies* for the trans-

[1] *http://www.cs.colorado.edu/serl/cbn/forwarding/*

mission and handling of messages across administrative domains. Following once again our analogy with IP networks, we must be able to implement the equivalent of policies of the Border Gateway Protocol (BGP). In particular, an autonomous system in a content-based network must be able to determine which messages are transmitted to other systems, and which messages are accepted from other systems for local delivery and/or for transit. Similarly, administrative policies should be able to determine which predicates are propagated outside an autonomous system, and in what form.

It is not completely clear to us which of these problems, if any, would pose fundamentally new research questions. Clearly, some mechanisms could be realized by engineering existing architectures and algorithms. For example, establishing filters, whether for outgoing, incoming, or transit messages, seems straightforward. In fact, the predicate language implemented by the network itself would probably suffice for this task. On the other hand, designing a good content-based routing protocol that interacts smoothly with the various filtering policies does not seem trivial. Therefore, we propose to study the design of a functional and efficient policy-controlled, content-based routing protocol.

2.3 Theoretical Foundations of Content-Based Routing

The problem of content-based routing, albeit not explicitly called that way, has been studied quite extensively [1, 2, 5, 6, 8, 11]. Most if not all these studies, including our own, focus on concrete routing protocols (or combined routing-forwarding protocols) and use simulation to analyze, compare, and validate solutions. This approach is standard practice in the design of network protocols, and, to be sure, we consider it very much valid in the context of content-based networking. However, we also see the need to frame the problem of content-based routing within a more general and solid theory. Such a theory would help us—indeed, it would be necessary—to understand more fundamental properties of routing protocols.

- *Protocol correctness.* A simulation for a network protocol is equivalent to testing for software: it may uncover incorrect behaviors, but it can not prove a protocol correct. In order to do that, we must first *define* correctness. Then we must deduce correctness from a suitable representation of the protocol and the network environment. In short, in order to analyze protocol correctness, we must formalize content-based routing.

- *Protocol complexity.* Simulation analyses are difficult to read because of the many configuration parameters that define each experiment. Even if one understands the performance trend observed through simulation in a section of the parameter space, it is by no means clear whether and how such trends can tell anything about other parts of the parameter space. Also, simulations may not capture essential, implementation-independent properties of a protocol. Having a formal model of networks and protocols would allow us to formulate more general statements about the complexity of protocols. In particular, instead of showing the cost (e.g., in terms of control messages) of a protocol for a few network sizes, we would like to characterize the *asymptotic complexity* of that protocol.

- *Inherent properties of content-based routing.* Ideally, we would also like to understand universal properties of content-based routing. For example, we would like to establish some lower-bounds for the complexity of content-based routing. Better yet, we would like to characterize the trade-offs between forwarding costs and memory or control-traffic costs. Non-trivial lower bounds are notoriously hard to find, so this is a somewhat ambitious goal. Nonetheless, we consider it essential to gain an in-depth understanding of content-based routing.

With these goals in mind, we have begun to formulate a theory of content-based routing [3]. Once again, we got inspirations and ideas from the quite extensive literature on theoretical approaches to traditional (unicast) routing. Our initial approach was to extend the standard network and routing model of Peleg and Upfal [9], and based on this model, study the memory requirements of various routing schemes at steady-state. This means the memory necessary to implement the forwarding functions realized by routers (total for the entire network, or maximum for an individual router). Intuitively, this is the amount of forwarding state produced and maintained by a routing protocol.

The rationale for this choice of focus is twofold. First, the amount of forwarding state gives a good indication of the complexity of forwarding. Second, the memory requirement at steady state is independent of the dynamics of both the network and its applications, and therefore characterizes an essential property of the chosen routing scheme. In order to analyze this complexity, we also introduce the concept of *disjunction advantage*, which expresses the memory savings obtained by combining partially-overlapping predicates.

2.4 Routing: Engineering Plus New Ideas

A fair number of ideas have been explored in content-based routing. Nonetheless, we believe that a lot remains to be done in this fundamental area. Our own plan is to follow two opposite research directions. First, we believe that existing protocols and ideas should be studied in depth, going beyond simulation, exploring deployments on distributed testbeds, and focusing on protocol engineering.

The second direction leads amounts to what is sometimes referred to as "radical design." In particular, we would like to question some basic assumptions and we intend to explore completely new ideas or old ideas that are generally considered inapplicable or impractical. Rather than exploring any specific idea in detail here, we will list some in breadth-first order.

2.4.1 Peer-to-Peer Network Models

Many have studied ways to implement content-based networking on top of simple and well-understood peer-to-peer networks such as distributed hash tables (DHTs) [12, 13]. We like these ideas, and believe that this area of research is likely to evolve, more or less following the evolution of peer-to-peer systems themselves. In particular, research in content-based routing on peer-to-peer overlays should look at the following questions.

- How does the combination of content-based routing and the peer-to-peer network behave under significant churn?

- Is it possible to use DHTs and still maintain some kind of locality principle, where short distances in the content-based network correspond to short distances in the P2P overlay?
- How does the combination of content-based routing and the peer-to-peer network behave in the presence of *non-complete* overlays? Many P2P systems are designed under the implicit assumption that every peer can connect directly to any other peer. With IP overlays, this is possible only in theory. In practice, links are very much asymmetric because of the prevalence of firewalls and network-address translation.
- Can DHTs work on statically-configured, non-full network topologies? And what are the implications on content-based routing?
- Intuitively, it would be easier to implement content-based routing on top of a structured peer-to-peer system where the addressing scheme (e.g., the hash values in a DHT) maintain some of the properties of predicates in content-based communication. Are there any good "structures" for peer-to-peer systems that preserve the "content-based relations" between addresses?

We are exploring some combinations of these ideas. Specifically, we are working on novel peer-to-peer "structures" constructed on the basis of content-based addresses as defined by some probabilistic matching function as discussed in Section 2.1. In particular, the idea here is to build a typical peer-to-peer index where identifiers amount to term vectors, and distances are defined by a measure of similarity between term vectors.

2.4.2 New, Old Ideas for Routing

We see an opportunity to apply well-known protocols and other simple, perhaps naïve protocols to content-based routing. The essence of this idea is that these approaches have not even been considered because their seemingly high cost (or low effectiveness). However, we believe that they merit to be reconsidered more openly.

The first idea here is to revive simple *broadcast* protocols to transmit content-based routing information. For obvious reasons of cost, this is generally considered a bad idea in a content-based network. However, we are not convinced that the costs are really that high, especially for relatively stable networks. In fact, following once again the inspiration of Internet routing, we believe that a *content-based link-state* routing protocol might be the best option for small and medium-size networks. The obvious advantage of this approach is that it would give each router a complete and updated view of the entire network. This would then allow routers to implement various types of routing schemes, including schemes that achieve optimal (e.g., functionally correct and latency-minimal) delivery. In our recent study of memory requirements of content-based routing, we define two new schemes that do just that [3].

This first idea is best applied to networks where the topology is very stable and where content-based information (i.e., predicates) are also relatively stable. Another idea is to focus on an opposite scenario, where both topology and predicates are very fluid. This case corresponds, for example, to an ad-hoc network of mobile devices, but also it may apply to high-churn peer-to-peer networks. Effective routing is very difficult in these cases because routing information is very quickly invalidated by fast-paced changes in the network.

Extremely fluid networks lead us to consider what we call *almost-zero-information routing*. The idea is to rediscover routing protocols that are based on local decisions that are almost completely oblivious of topology and non-local predicates. An approach that takes this idea to an extreme is to route messages on a random walk. Starting from this end of the spectrum, our intuition is that even little routing information could strike a good balance between the well-known bounds of random walks and the more expensive routing protocols based on shortest-path routes. At this point, we do not have any concrete protocol under development, but we are planning to explore this idea.

2.5 Privacy for Interests

One of the fundamental problems in content-based networking, as in any communication service, is *privacy*. However, the privacy goal in this context differs significantly from that of a traditional communication service. In the traditional model, Alice sends a message to Bob and does not want anyone else to read it. The network must be told the address of Bob in order to deliver the message, but the content of the message can remain hidden from the network. In content-based networking, Alice does not specify a destination, but instead simply sends a message to the network. Alice's intent is to have the content of the message examined in order to have it delivered to interested parties.

On the receiver side, Bob does not passively wait for messages addressed to him, but instead actively requests that the network select messages of interest on his behalf. The problem is that Bob must tell the network his interests, and he might consider this disclosure of personal preferences an unacceptable violation of his privacy.

Privacy in content-based networking is therefore primarily a problem of the trust relationship between receivers and the network, which leads to a contradictory and challenging goal: *How do we design an effective and efficient content-based networking service in which receivers can hide their preferences from the network and yet the network can deliver all and only the messages in which the receiver is interested?*

We have been studying this problem for quite a while, and in fact we already designed and implemented an encoding scheme that attempts to hide messages and predicates while at the same time allowing routers to efficiently match messages and predicates. This encoding also allows routers to verify the inclusion relation between predicates with the same efficient algorithms. This scheme is based on a categorization of filters and messages, and on Bloom filters built with cryptographic hash functions. In addition to the encoding, we developed several matching algorithms based on the predicate and message encoding.

Our approach seems promising. However, it was developed from a purely practical and informal perspective, without the necessary basis of a provable security model. Inspired by the work of Raiciu and Rosenblum [10], we are planning to study our scheme under the model proposed by Chang and Mitzenmacher [7].

2.6 Workloads and Network Models

Methodical evaluation of ideas in content-based networking is crucial. This is a truism that is still worth mentioning.

The evaluation of content-based networking protocols and systems is often difficult due to the lack of direct experience with real deployments, and therefore for the lack of models of applications and networks. Obviously, researchers do their best individually to evaluate their ideas. However, these evaluations should be understood and generally accepted by other researchers. As a first requirement, regardless of their validity, experiments must be repeatable by others. This first issue does not raise research questions, but is rather a matter of method and openness. One thing that could help is to fully publish all the materials related to experiments. We have tried to do just that consistently in recent years, by publishing the code and the workloads that defines the experimental results we publish, and we encourage others to do the same.

The second issue here is that the validity of an experiment depends on the validity of the models that define the experimental setup. Generally speaking, the problem is not the lack of models, but rather the lack of *widely accepted* models. This suggests two types of efforts. One is a typical community effort, where models gain acceptance because they are formulated and discussed by committees of researchers rather than by individuals. The second effort, which is what we propose here as a line of research, is based on the idea that good models *can* be constructed based on both real and synthetic applications and scenarios. This research would amount to studying and characterizing applications and networks that, although they do not currently use content-based communication, may be conveniently redesigned to do that.

2.7 A Middleware for Content-Based Communication

The content-based communication interface we defined in Section 1 is a *host* interface. What this means in practice is that the network allows a host to declare a predicate, but does not distinguish multiple applications running on a host. Also, this host interface is *stateless*, meaning that every predicate declaration overwrites all previous declarations. Notice that this choice of interface differs significantly from all existing publish/subscribe systems, where applications can publish and subscribe through one or more independent interfaces, and where subscriptions are accumulated by the system and must be explicitly canceled by applications (with "unsubscriptions").

The basic content-based networking interface is more similar in spirit to the interface of an IP host, where the network layer addresses hosts (or, more precisely, network interfaces) and where individual applications can be addressed only through higher-level protocols such as UDP and TCP.

While we do not see anything particularly wrong in having an application access the content-based *host* interface directly, we also believe that most applications would require higher-level services. These higher-level services may be implemented by the host operating system—in the same way most modern operating systems implement the IP protocol stack—or by a middleware service, or by library functions. Regardless of the type of implementation, which will depend upon engineering considerations, we see the opportunity and the need for designing high-level services on top of the basic content-based communication interface. The following is an initial list of such basic services.

- *Subscription service:* this is the stateful and multiplexed interface described above. An application creates one or more *content-based sockets*, and dynamically binds them to a set of subscriptions. Of course, just like normal sockets, content-based sockets would have their own message queues, which the application can access with the typical synchronous or asynchronous, blocking or non-blocking read operations.

- *Content directory:* this service should aid applications in finding other applications or higher-level services. At a minimum, this service should provide a directory of terms in use by senders. This service could be seen as the equivalent of the Domain Name Service (DNS). It is also easy to imagine more sophisticated services based on the semantic relations between terms.

In addition to these basic services, we envision a large set of higher-level services. The following are just a few ideas in this direction.

- *Synthesis of predicates and data adaptation:* complex applications are likely to want to take advantage of an automatic derivation or synthesis of predicates as well as an automatic adaptation of data extracted from messages. The synthesis of predicates could be driven by application-specific languages or by the applications' data model. As an example, consider a family of management applications implemented within a spreadsheet. Specific examples include a portfolio management system and a logistics support in a computer-aided manufacturing system. In these cases, the applications need to have access to several data sources, and it would make sense for the spreadsheet application to mediate access to the content-based network through a uniform interface. In particular, it would make sense to (1) derive content-based communication predicates from formulas in the spreadsheet, and (2) to feed data extracted from messages directly into the appropriate positions in the spreadsheet. A similar content-based middleware could be designed also for database applications.

- *Guided composition of predicates:* interactive applications might give users more or less direct access to the content-based network. In these cases, the users would benefit from a composition and validation service for both messages and predicates. Such a service would guide users in the composition of messages and predicates so as to avoid errors and inconsistencies. Many similar services are implemented in other classes of applications. For example, database applications have query editors and query-by-example interfaces.

- *Integration with address-based services:* applications may require multi-modal communication. For example, a client might first discover a service through a service advertisement transmitted through the content-based network, but then would establish a session with the service provider through a normal unicast connection using a remote procedure call protocol. In this case, it might be beneficial or necessary to integrate these different communication services—content-based and address-based unicast—for example, because some properties of the unicast connection might be specified in the service advertisement. It is easy to imagine other such types of multi-modal sessions. Therefore, it is natural to assume that a middleware for content-based communication would also integrate traditional middleware services.

3. CONCLUSIONS

Content-based communication and networking in an exciting and promising research area that combines algorithmic problems, classical networking problems, and data management problems, including security. We have been working in this area for a number of years. Our approach is based on a systems-oriented view, but we also attempt to cast and analyze our ideas within solid theoretical models. Our intent is to continue in this general direction, exploring many of the research questions that remain open. In this paper we listed some of these questions, along with a few concrete ideas and plans.

4. REFERENCES

[1] 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 19th IEEE International Conference on Distributed Computing Systems (ICDCS '99)*, pages 262–272, Austin, Texas, May 1999.

[2] F. Cao and J. P. Singh. Efficient event routing in content-based publish-subscribe service networks. In *IEEE Conference on Computer Communications (INFOCOM '04)*, pages 929–940, Hong Kong, China, Mar. 2004.

[3] A. Carzaniga, A. J. Rembert, and A. L. Wolf. Understanding content-based routing schemes. Technical Report 2006/05, Faculty of Informatics, University of Lugano, Sept. 2006.

[4] 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 Annual ACM Symposium on Principles of Distributed Computing*, pages 219–227, Portland, Oregon, USA, July 2000.

[5] 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.

[6] A. Carzaniga, M. J. Rutherford, and A. L. Wolf. A routing scheme for content-based networking. In *IEEE Conference on Computer Communications (INFOCOM '04)*, pages 918–928, Hong Kong, China, Mar. 2004.

[7] Y.-C. Chang and M. Mitzenmacher. Privacy preserving keyword searches on remote encrypted data. Cryptology ePrint Archive, Report 2004/051, 2004.

[8] P. Ji, Z. Ge, J. Kurose, and D. Towsley. Matchmaker: Signaling for dynamic publish/subscribe applications. In *11th IEEE International Conference on Network Protocols (ICNP '03)*, pages 222–233, Nov. 2003.

[9] D. Peleg and E. Upfal. A trade-off between space and efficiency for routing tables. *Journal of the ACM*, 36(3):510–530, July 1989.

[10] C. Raiciu and D. S. Rosenblum. Enabling confidentiality in content-based publish/subscribe infrastructures. In *Securecomm '06: Proceedings of the Second IEEE/CreatNet International Conference on Security and Privacy in Communication Networks*, 2006.

[11] A. C. Snoeren, K. Conley, and D. K. Gifford. Mesh-based content routing using XML. In *Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles (SOSP '01)*, pages 160–173. ACM Press, Oct. 2001.

[12] P. Triantafillou and I. Aekaterinidis. Content-based publish-subscribe over structured P2P networks. In *Third International Workshop on Distributed Event-Based Systems (DEBS 2004)*, pages 104–109, Edinburgh, Scotland, UK, May 2004.

[13] R. van Renesse and A. Bozdog. Willow: Dht, aggregation, and publish/subscribe in one protocol. In *Proceedings of the Third International Workshop on Peer-to-Peer Systems*, pages 173–183, 2004.