Distributed Publish/Subscribe

SIENA

Content–Based Networking

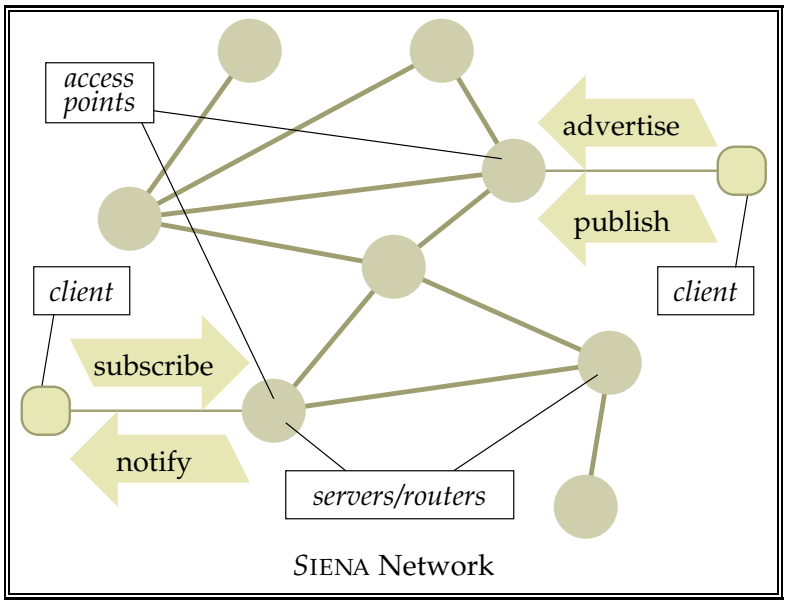
University of Colorado at Boulder — Software Engineering Research Laboratory

Publish/subscribe is a communication pattern in which information flow is directed by the interest of receivers rather than by explicit addresses determined by senders. This pattern is particularly suitable for a wide range of distributed applications.

The goal of SIENA is the realization of a ubiquitous publish/subscribe notification service accessible from every site on a wide-area network and suitable for supporting highly distributed applications requiring component interactions ranging in granularity from fine to coarse. The service is implemented as a network of servers that provide access points to clients. Clients use the access points to *advertise* information about events and subsequently to *publish* multiple notifications of the kind previously advertised, and to *subscribe* for notifications of interest. The service uses the access points to then *notify* clients by delivering any notifications of interest.

Applications

- ***system management:*** *network management, intrusion detection, software deployment*
- ***e-commerce:*** *information sharing, electronic auctions, news*
- ***middleware:*** *service discovery, component integration, message-oriented communication*



SIENA Network

The conceptual basis for the design of the SIENA network is our novel theory of *content-based networking*.

Content-Based Networking

Content-based networking is a generalization of a distributed publish/subscribe notification service [3]. At the physical-architecture level, a content-based network is identical to a traditional datagram network. We can think of it as a graph, where nodes are *hosts* or *routers*, and arcs are direct communication links.

Content-based networking differs significantly from traditional (unicast or multicast) networking in its service model. In a content-based network, nodes are not assigned a unique network address, nor are datagrams addressed to any specific node or node group. Instead, each node submits a *receiver predicate* (or *r-predicate*) that matches datagrams of interest for that node. Nodes can also send out datagrams, which the network will forward to all the nodes with matching r-predicates.

Similar to traditional networks, a router in a content-based network implements a *forwarding* and a *routing* function. *Content-based forwarding* means evaluating an incoming message against a table of predicates (the forwarding table) to determine to which neighbor nodes the message should be forwarded. *Content-based routing* amounts to exchanging predicates and other information with other routers in order to compile the forwarding table in such a way that messages will be forwarded correctly from senders to the designated receivers.

SIENA implements a routing optimization strategy that reduces the propagation of predicates exploiting the logical implications among r-predicates [2]. The effects are to reduce routing traffic, to reduce the size of the forwarding table at every router, and thereby allowing a faster forwarding function.

Prototype Implementation

The current implementation of SIENA offers two application programming interfaces, one for C++ and the other for Java. Both interfaces provide the complete data model and subscription language of SIENA. Server implementations are also available in C++ and Java. SIENA is integrated with XML event producers and consumers through SXML, a tool we have developed that offers an end-to-end interface based on XML transformation rules and XPath subscriptions. SIENA has a modular communication mechanism which currently supports links over UDP, TCP with either simple one-time connections or more efficient persistent connections, and SSL for complete communication security.

Notification Model

An *event notification* (or simply a *notification*) is a set of typed attributes. For example, the following represents a stock price change event.

string	*class*	= *finance/exchanges/stock*
time	*date*	= *Mar 4 11:43:37 MST 1998*
string	*exchange*	= *NYSE*
string	*symbol*	= *DIS*
float	*prior*	= *105.25*
float	*change*	= *-4*
float	*earn*	= *2.04*

Each individual attribute has a *type*, a *name*, and a *value*, but the notification as a whole is purely a structural value derived from its attributes. Attribute names are simply character strings. The attribute types belong to a predefined set of primitive types commonly found in programming languages and database query languages.

Subscription Model

An *event filter*, or simply a *filter*, selects event notifications by specifying a set of attributes and constraints on the values of those attributes. Each constraint is a tuple specifying a type, a name, a binary predicate operator, and a value for an attribute. The operators provided by SIENA include all the common equality and ordering relations ($=, \neq, <, >$, etc.) for all of its types; substring ($*$), prefix ($> *$), and suffix ($* <$) operators for strings; and an operator *any* that matches any value.

The following is an example of a filter that matches price increases for stock DIS on the NYSE.

string	*class*	$>*$	*finance/exchanges/*
string	*exchange*	=	*NYSE*
string	*symbol*	=	*DIS*
float	*change*	>	*0*

Expressiveness and Scalability

SIENA achieves a good balance of expressiveness and scalability [1]. Our chosen level of expressiveness represents a compromise, at which notification structure, attribute types, and attribute operators approximate those of the well-understood and widely-used database query language SQL. At the same time, by having a well-defined set of types and operators, SIENA can perform efficient routing and forwarding based on the content of notifications.

References

[1] 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.

[2] 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.

[3] A. Carzaniga and A. L. Wolf. Content-based networking: A new communication infrastructure. In *NSF Workshop on an Infrastructure for Mobile and Wireless Systems*, Scottsdale, Arizona, Oct. 2001.

Additional Information and Available Software

http://www.cs.colorado.edu/serl/siena/

Principal Investigators

Antonio Carzaniga
University of Colorado at Boulder
Dept. of Computer Science
430 UCB
Boulder CO 80309-0430 USA
Phone: +1-303-492-7852
Fax: +1-303-492-2844
E-mail: carzanig@cs.colorado.edu

Alexander L. Wolf
University of Colorado at Boulder
Dept. of Computer Science
430 UCB
Boulder CO 80309-0430 USA
Phone: +1-303-492-5263
Fax: +1-303-492-2844
E-mail: alw@cs.colorado.edu

Acknowledgments

Effort sponsored by the Defense Advanced Research Projects Agency, and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement numbers F30602-94-C-0253, F30602-97-2-0021, F30602-98-2-0163 and F30602-99-C-0174; by the Air Force Office of Scientific Research, Air Force Materiel Command, USAF, under grant number F49620-98-1-0061; and by the National Science Foundation under Grant Number CCR-9701973. 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 or the U.S. Government.