Harmony: Holistic messaging middleware for event-driven systems

P. Dube
N. Halim
K. Karenos
M. Kim
Z. Liu
S. Parthasarathy
D. Pendarakis
H. Yang

In this paper, we present Harmony, a holistic messaging middleware for distributed, event-driven systems. Harmony supports various communication paradigms and heterogeneous networks. The key novelty of Harmony is the unified provision of end-to-end quality of service, security, and resiliency, which shields the applications from the underlying network dynamics, failures, and security configurations. We describe the Harmony architecture in the context of cyber-physical business applications and elaborate on the design of its critical system components, including routing, security, and mobility support.

INTRODUCTION

Distributed event-driven applications have become increasingly popular in recent years, in part as a result of the proliferation of embedded computation and communication capabilities. Examples of these applications include energy and utility grids, intelligent transportation, telemedicine, industrial control, and inventory management. Until recently, most of these systems were deployed on a relatively small scale, typically using proprietary hardware and software components, such as special-purpose processors and operating systems. However, as the scope of these applications increases to cover a much larger number of nodes and they are distributed across larger geographic areas, a migration from mostly proprietary solutions to commoditized hardware, software, and networking technologies is desirable. While this migration will be aided by the pervasive deployment of wired and wireless networks, the commoditization of secure hardware components, and the maturing of virtualization technologies, it will only become feasible if the new infrastructures succeed in providing real-time performance, reliability, security, and privacy features on par with those associated with customized and proprietary solutions.

This paper presents Harmony, a novel messaging middleware for distributed, event-driven applications with stringent requirements for security, resiliency, and quality-of-service (QoS) parameters such as delay and throughput. It can operate on heterogeneous infrastructures using both wired and wireless networking technologies, and leverage the emerging capabilities in the area of trusted computing hardware.

To handle the mobility of and dynamic associations between endpoints, Harmony provides capabilities for autonomous organization of endpoints into domains based on their credentials, administrative ownership, application attributes, network connectivity, location, etc. Connectivity between these domains is controlled by security mechanisms that ensure delivery of encrypted messages to authorized endpoints only. The overlay is QoS-aware—i.e., it routes messages so as to achieve target levels of delay and throughput, and employs resilient multipath routing and message storage techniques to ensure message delivery in the presence of a wide range of failure scenarios, including both independent and correlated failures. Harmony abstracts the capabilities and states of underlying physical devices, such as sensors, actuators, and processing nodes, to the application layer, and provides application programming interfaces (APIs) that facilitate the dynamic deployment of software components onto the most appropriate physical devices, as well as maintaining interconnections between these components.

The security, resiliency and QoS requirements of event-driven applications have to date been addressed only in a piecewise manner; this approach increases deployment and management complexity, reduces performance, and is error-prone. This is particularly the case when these piecewise solutions are individually developed with different models and assumptions concerning the underlying network and set of systems. As a result, it is nontrivial to introduce these solutions into an existing system and achieve the performance, resiliency, and security levels required by critical infrastructure applications, such as energy production and distribution, telemedicine, and intelligent transportation. In contrast, Harmony takes a holistic approach by providing the critical system capabilities through a unified architecture and an array of novel techniques that interact seamlessly with each other.

ARCHITECTURE

An event-driven system typically consists of a large set of endpoints that are distributed over a large geographic area and span many different physical communication networks. For example, in a "cyber-physical-business" (CPB) application,[1] a system collects information about the physical world through embedded sensor devices, using advanced sensor and wireless communication technology. This information is sent to the back-end processing center through wide-area networks, normally the Internet, and the processing results are further dispatched to actuators in the field for closed-loop feedback control or to the front ends for user awareness and decision support.

The messaging layers in such systems must be able to support various communication paradigms, such as the unicast, multicast, and publish/subscribe models. In the publish/subscribe model, the endpoints may not directly communicate with each other. Instead, the messages pass through mediation agents or brokers that compare the messages with the subscriptions and locate the intended receivers. Also, the period of the subscriptions may be specified as an event consisting of one or multiple messages. Nevertheless, Harmony focuses on the networking support of delivering individual messages among the communicating endpoints (sensors, brokers, processing elements, actuators, etc.), subject to various nonfunctional requirements of the applications. In particular, many event-driven applications rely heavily on the quality parameters of end-to-end communications, such as latency and throughput, so that the system can respond to the physical events in real time and in a deterministic manner. Moreover, the system must achieve a high degree of resiliency in the presence of various network and system failure scenarios, and provide security and privacy protection for the messages as they pass through the system.

We have developed a messaging middleware, Harmony, which addresses the above challenges through a holistic architecture. As shown in ***Figure 1***, the Harmony architecture consists of a data plane, a routing plane, and a management plane.

The *data plane* supports various communication paradigms and patterns through a unified messaging substrate. Depending on the requirements of the application, this substrate may deliver the messages through multiple parallel paths for increased resiliency and throughput, with various source and network coding schemes being employed along these paths. For security and privacy protection, the messages are encrypted and authenticated at the

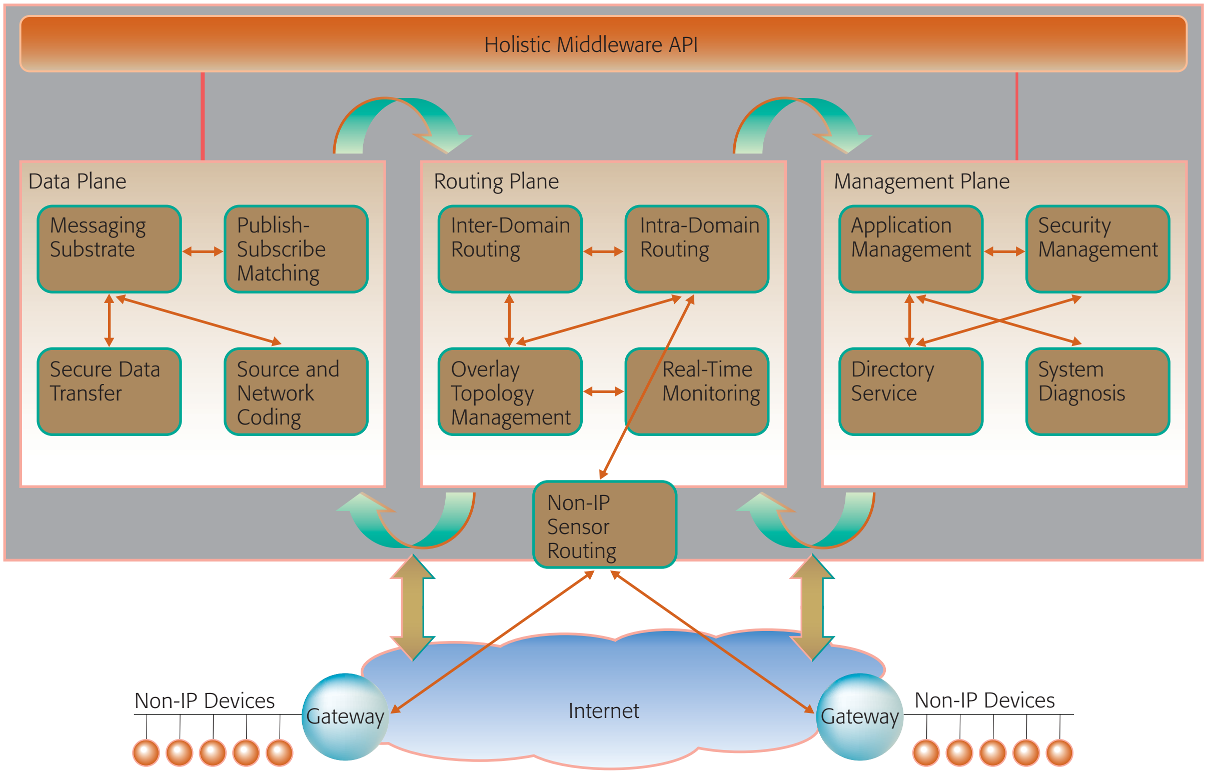

Figure 1
Harmony architecture

source, so that no adversary can eavesdrop or tamper with the traffic in transit.

The *routing plane* involves a set of overlay routing mechanisms for establishing and maintaining the end-to-end paths within the infrastructure nodes, subject to the QoS, security, and resiliency requirements of the application. It also provides sensor routing mechanisms for the embedded sensors to self-organize into a communication network, possibly using non–IP (non–Internet Protocol) technology. The topology of these networks is optimized based on a distributed monitoring mechanism that periodically collects essential network characteristics, such as link quality and available resources at different nodes.

The *management plane* provides the essential management functionalities, such as registering the application endpoints, configuring the security policies, and detecting the current system and network nodes status. Using a directory service, such information is shared and updated across the entire system in an efficient and consistent manner. The real-time system states can further be visualized in an interactive control console for the purpose of system diagnosis.

DESIGN

In the following, we briefly describe the design of the Harmony system. Due to space limitations, we focus on a few critical system components here, namely QoS-aware and resilient routing, security provisioning, and mobility support.

QoS-aware and resilient routing

The end-to-end paths are established in Harmony through two tiers of routing mechanisms: intra-domain routing and inter-domain routing. Each routing domain consists of a set of nodes that are relatively close to each other and able to communicate through a given medium (e.g., wireless channel or Ethernet) and protocol stack (e.g., TCP/IP [Transmission-Control Protocol/Internet Protocol], sensor).

The function of intra-domain routing can be stated as follows: Given the node locations, interconnection topology, failure statistics, and event stream requirements, an end-to-end path must be computed and constructed (consisting of one or more sub-paths) between the source and the destination end points, that jointly: (1) satisfies QoS objectives (i.e., meets the end-to-end delay requirements); (2) satisfies resiliency constraints (i.e., maintains the required level of message delivery despite both independent and correlated failures); and (3) optimizes resource utilization (cost). Once multiple sub-paths are constructed, the messages are duplicated along each sub-path. In other words, a message is lost if and only if all sub-paths fail simultaneously.

We consider two types of node failures: independent and geographically correlated failures. For the independent failure model, we associate each node with an independent failure probability. With respect to geographically correlated failures, we consider that faults affect an area which is represented as a disk with a given center and a radius of exponentially distributed length. We then use the euclidean distance to compute the failure correlations among the nodes. The key advantage of the correlated disk-failure model is that although simplistic, it allows for effective, tractable, and practical definition of the geographical failure correlations among various nodes.

The main challenge in path computation is that the combination of correlated as well as independent failures results in a reliability function that is non-convex. As such, traditional flow optimization and linear programming techniques cannot be directly applied. Therefore, in order to solve the optimization problem, we decompose it into two basic steps. In the first step, we compute *k* non-disjoint shortest paths ordered by their respective delays. This is done by using an efficient variant of the well-known *k*-shortest path algorithm. In the second step, we use a search algorithm to calculate the most cost-efficient combination of these paths that satisfies the resiliency requirement. As the resiliency computation is a complicated non-polynomial-time process, we use an efficient heuristic, based on the branch-and-cut algorithm, to reduce the number of computations. Specifically, we eliminate a number of candidate routes (and thus the computation overhead) by computing the resiliency of those routes that have multiple paths and ignoring all routes that are subsets of multipath routes which have a resiliency below the requirement.

In contrast, inter-domain routing is jointly performed by gateway nodes of multiple domains. These nodes may be spread over a large geographic area, and their total may grow to a large number. They may also be mobile, as domains can move and be created or removed dynamically. As a result, we explore distributed routing schemes for inter-domain routing that would be scalable and efficient in terms of mobility support. Proactive routing schemes, such as the standard link state (SLS) and standard distance vector (SDV) schemes,[2] continuously maintain routing information. These may be efficient for high-traffic workloads, but can be too costly for high-mobility nodes. Reactive schemes, such as dynamic source routing (DSR),[3] compute paths when requests arrive. These schemes may be effective for high-mobility nodes but may not be efficient for high-traffic workloads. As we expect the gateway nodes in Harmony to be semi-mobile and the traffic among them to be high, we plan to explore hybrid routing schemes that will be suitable for our particular environment.

Security provisioning

Harmony was designed, from its inception, with built-in security mechanisms. In particular, we leverage the advances in secure hardware such as the trusted platform module (TPM) microcontroller that is available in most commodity PCs today. The TPM provides capabilities for securely generating and storing the cryptographic keys, as well as remote attestation (i.e., the reporting, by a trusted-platform device, of its integrity state) and verification. These capabilities can be used to secure both the management plane and the data plane in the Harmony system.

In order to prevent an adversary from hijacking the system, we verify the identity and security status of each Harmony node when it joins the system. This is done through a direct anonymous attestation (DAA) process as described in Reference 4. The basic idea is to establish the trust chain from the endorsement key (EK) that is embedded in the chip during its manufacture. The TPM also reports its platform configuration register (PCR) values, which store the signatures of the critical system states (e.g., boot sector and binary files), so that the system can ensure that it has not been compromised or tampered with in the past. Based on these security bindings, the source and the destination nodes can further negotiate shared keys which are used to encrypt and authenticate the data traffic.

We also considered scenarios where the Harmony system consists of nodes with heterogeneous security capabilities for the secure hardware or software available on the devices. In these cases, each node is associated with a security label, and multiple nodes with the same label further form a security domain. Depending on the application requirements, the traffic can be confined within one or multiple security domains that provide the desired level of protection for the in-transit traffic. In this case, the overlay routing algorithm discussed earlier is applied separately in each security domain, so that the resulting security-enforced paths also satisfy the QoS and resiliency requirements.

Mobility support

The endpoints in an event-driven system, especially the sensor and actuator nodes, are often mobile. Thus, it is critical for Harmony to provide seamless mobility support. From the end-to-end communication perspective, this is achieved through late binding between the application identifiers and the Harmony addresses for these mobile endpoints. Specifically, each endpoint has a unique application identifier, which does not change as it moves. In contrast, when a mobile endpoint is attached to Harmony from a different location, it is reassigned a new Harmony address. The mapping between application identifiers and Harmony addresses is maintained internally by Harmony. In this way, the endpoint mobility is transparent to the application, which deals only with the stable application identifiers, while Harmony can always divert the traffic to the current location of an endpoint based on its latest Harmony address.

Despite these functions, transient message losses may occur during the period when a mobile endpoint changes its Harmony address, because it takes some time (up to a few seconds) for the endpoint to acquire a new address and update the directory. To minimize such service interruptions, we developed several intelligent association schemes that can reduce the expected frequency of handoffs for the mobile endpoints. We use the association that has the longest lifetime, which is the optimal scheme when the node has a predictable mobility pattern. In practice, however, an endpoint may have a complicated mobility pattern that is unknown in advance.

To address this issue, we proposed two online association algorithms that seek to estimate the lifetime of each association and make the association decisions accordingly. The first algorithm simply selects an attaching point uniformly at random from the available ones. Despite its simplicity, this algorithm provides a worst-case competitive ratio of log K, where K is the maximum number of attaching points available at any time.

■ Harmony is designed with built-in security mechanisms. ■

Our second online algorithm uses the past movement trajectories of an endpoint to predict its future movements. Given the estimated future trajectory, we can predict how long each available association can persist, and then choose the one with the longest predicted lifetime. Details of these two schemes are described in Reference 5. Our extensive trace-driven simulations show that they can reduce the handoff frequency for typical mobile endpoints by 40 percent as compared to the existing signal-strength-based association schemes.[5]

RELATED WORK

In this section we compare Harmony to other state-of-the-art messaging middleware and overlay networking technologies.

Much of the existing messaging middleware, such as Gryphon[6] or Siena,[7] focuses on providing scalable and efficient communication using the publish/subscribe model. This is achieved by a number of techniques that optimize broker performance, such as matching latency, system throughput, and availability. However, communication in such systems is typically based on TCP/TP over the Internet. Little consideration has been given to the end-to-end communication performance parameters, such as latency and resiliency, in the context of event-driven applications that may run over mobile, low-capacity, and failure-prone networks. As such, Harmony and these existing systems can complement each other and work together as a responsive and reliable messaging substrate for event-driven applications.

Java** Message Service (JMS)[8] is a set of APIs and programming models for point-to-point and publish/subscribe messaging. A JMS provider implements its own messaging infrastructure for routing and transportation of messages between JMS clients, resulting in non-interoperability due to the use of different transport protocols and wire formats. On the other hand, the Advanced Message Queuing Protocol** (AMQP)[9] enables complete interoperability by explicitly defining the semantics of each AMQP node through its service model, and by specifying common wire formats for message transfer. Both JMS and AMQP provide only interface and message format specifications, while assuming the existence of a provider or a messaging layer. In contrast, Harmony not only exposes a set of APIs that are compatible with JMS, but also provides a system implementation that accomplishes reliable, secure, and QoS-aware message delivery.

Recently there has been a large body of work on the use of overlay networks to improve Internet communication in terms of QoS and resiliency (see References 10 and 11 and the references therein). Harmony has similarities with these solutions, such as the use of multipath communication for improved resiliency. However, Harmony differs from them fundamentally in that it addresses the unique characteristics of distributed, event-driven systems, such as correlated failures and mobile endpoints, in a holistic manner.

CONCLUSION

With the increasing popularity of distributed event-driven applications, there is a strong need for a unified messaging solution to address the complicated performance, resiliency, and security requirements essential to these applications. To this end, we have created the architecture and design of the Harmony messaging middleware, which can provide an abstract layer of secure, QoS-aware, and resilient messaging on top of commoditized and heterogeneous hardware, software, and network infrastructure.

We have implemented a prototype of the Harmony system and integrated it with the Internet-Scale Control System (iCS), an intelligent energy and utility distribution system developed by the IBM Research Division. Through this integration, iCS not only benefits from the greatly reduced complexity in maintaining the communication infrastructure, but also extends its functionalities and end-to-end performance assurances in terms of timeliness, robustness, and security. Currently, we are investigating various optimization techniques that can further improve the scalability and efficiency of Harmony in a large enterprise application setting.

ACKNOWLEDGMENT

The authors would like to thank Francis Parr for his insightful input which helped improve the work described in this paper.

**Trademark, service mark, or registered trademark of Sun Microsystems, Inc., or the AMQP Working Group in the United States, other countries, or both.

CITED REFERENCES

1. E. A. Lee, "Cyber-Physical System—Are Computing Foundations Adequate?" Cyber-Physical Systems Workshop, National Science Foundation (2006), http://ptolemy.eecs.berkeley.edu/publications/papers/06/CPSPositionPaper/Lee_CPS_PositionPaper.pdf.
2. A. S. Tanenbaum, *Computer Networks*, Third Edition, Prentice Hall, Upper Saddle River, NJ (1996).
3. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks," in *Mobile Computing*, T. Imielinski and H. F. Korth, Editors, Kluwer Academic Publishers (1996), pp. 153–181.
4. E. Brickell, J. Camenisch, and L. Chen, "Direct Anonymous Attestation," *Proceedings of 11th Conference on Computer and Communications Security (CCS)*, ACM Press, New York (2004), pp. 132–145.
5. M. Kim, Z. Liu, S. Parthasarathy, D. Pendarakis, and H. Yang, "Association Control in Mobile Wireless Networks," *Proceedings of the 27th Conference on Computer Communications (INFOCOM)*, IEEE (2008, to appear).
6. M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, and T. D. Chandra, "Matching Events in a Content-based Subscription System," *Proceedings of the 18th Annual Symposium on Principles of Distributed Computing (PODC)*, ACM Press, New York (1998), pp. 53–61.
7. 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**, No. 3, 332–383 (2001).
8. M. Hapner, R. Burridge, R. Sharma, J. Fialli, and K. Stout, *Java Message Service Specification, Version 1.1*, Sun Microsystems, Inc. (2002).
9. *AMQP: A General-Purpose Middleware Standard*, AMQP Working Group protocol specification (2006), http://www.redhat.com/f/pdf/amqp/amqp0-10.pdf.
10. D. G. Anderson, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient Overlay Networks," *Proceedings of the 18th Symposium on Operating Systems Principles (SOSP)*, ACM Press, New York (2001), pp. 131–145.
11. L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Hatz, "OverQoS: An Overlay based Architecture for Enhancing Internet QoS," *Proceedings of the 1st Sympo-*

sium on Networked System Design and Implementation (NSDI), ACM Press, New York, pp. 71–84.

Accepted for publication December 13, 2007.
Published online April 18, 2008.

Parijat Dube
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (pdube@us.ibm.com). Dr. Dube received his M.S. degree in electrical communications engineering from the Indian Institute of Science in Bangalore in 2001 and his Ph.D. degree in computer science in 2002 from the University of Nice-Sophia Antipolis, where he was affiliated with INRIA (Institut National de Recherche en Informatique et Automatique). He joined the IBM Thomas J. Watson Research Center in 2002. His research interests include distributed systems, performance engineering of IT systems, revenue management, and pricing.

Nagui Halim
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (halims@us.ibm.com). Mr. Halim is the director of Event and Streaming Systems at the IBM Thomas J. Watson Research Center in Hawthorne, NY. He completed his undergraduate studies in physics at Yale University in New Haven, CT, in 1978. His main research interests are transaction processing, memory, and software systems.

Kyriakos Karenos
Department of Computer Science and Engineering, University of California, Riverside, 900 University Ave., Riverside, CA 92521 (kkarenos@cs.ucr.edu). Mr. Karenos is currently a Ph.D. degree candidate at the University of California, Riverside. He received a B.Sc. degree from the University of Cyprus in Nicosia, Cyprus, and an M.Sc. degree from the University of California, Riverside, both in computer science. His area of research focuses on providing quality of service in event-driven information delivery systems including sensor networks and Internet-scale pervasive computing.

Minkyong Kim
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (minkyong@us.ibm.com). At the IBM Thomas J. Watson Research Center, Dr. Kim has been involved in the Harmony and Distillery projects. Her research interests include distributed systems, networks, and mobile computing. Prior to joining IBM, she worked as a postdoctoral research fellow in the department of computer science at Dartmouth College. She received her Ph.D. degree in computer science and engineering from the University of Michigan and both M.S. and B.S. degrees in computer engineering from Seoul National University, South Korea.

Zhen Liu
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (zhenl@us.ibm.com). Dr. Liu received a Ph.D. degree in computer science from the University of Orsay in Paris, France. He was with France Telecom Research and Development as a research associate from 1986 to 1988. He joined INRIA in 1988, first as a researcher, then as a research director. He joined the IBM Thomas J. Watson Research Center in 2000 and is currently the senior manager of the Next Generation Distributed Systems department. He is a fellow of the IEEE and has served on National Science Foundation panels and a number of conference program committees. He was the program co-chair of the Joint Conference of ACM Sigmetrics and IFIP (International Federation for Information Processing) Performance 2004, and the general chair of the ACM Sigmetrics conference in 2008. His current research interests are in distributed and networked systems, stream processing systems, sensor networks, performance modeling, distributed optimization, and control.

Srinivasan Parthasarathy
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (spartha@us.ibm.com). Dr. Parthasarathy is a research staff member at the IBM Thomas J. Watson Research Center. Prior to joining IBM, he received his M.S. and Ph.D. degrees in computer science from the University of Maryland at College Park (in 2003 and 2006, respectively) and his B.Tech degree from the Indian Institute of Technology in Madras, India, in 2000. His research interests lie at the junction of algorithm design and optimization on the one hand, and systems and networking on the other.

Dimitrios Pendarakis
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (dimitris@us.ibm.com). Dr. Pendarakis is the manager of the Secure Embedded Systems department at the IBM Thomas J. Watson Research Center. He received a diploma in electrical engineering from the National Technical University of Athens in 1990 and M.S. and Ph.D. degrees from Columbia University in 1992 and 1996, respectively.

Hao Yang
IBM Research Division, Thomas J. Watson Research Center, 19 Skyline Drive, Hawthorne, NY 10532 (haoyang@us.ibm.com). Dr. Yang received his Ph.D. degree in computer science from University of California at Los Angeles in 2006, and his M.S. degree in computer engineering from the National Lab of Pattern Recognition in China in 2001. He is currently a research staff member at the IBM Thomas J. Watson Research Center. His research interests include overlay networking, sensor networking, mobile computing, and network security. ■