Survey on Routing in Data Centers: Insights and Future Directions

Kai Chen, Northwestern University
Chengchen Hu, Xi'an Jiaotong University
Xin Zhang, Carnegie Mellon University
Kai Zheng, IBM Research
Yan Chen, Northwestern University
Athanasios V. Vasilakos, National Technical University of Athens

Abstract

Recently, a series of data center network architectures have been proposed. The goal of these works is to interconnect a large number of servers with significant bandwidth requirements. Coupled with these new DCN structures, routing protocols play an important role in exploring the network capacities that can be potentially delivered by the topologies. This article conducts a survey on the current state of the art of DCN routing techniques. The article focuses on the insights behind these routing schemes and also points out the open research issues hoping to spark new interests and developments in this field.

Data centers are driven by large-scale computing services such as web searching, online social networking, online office and IT infrastructure outsourcing, and scientific computations. It is expected that in the future a substantial portion of Internet communication will take place within data center networks (DCNs) [1]. To take the advantage of economies of scale, it is common for a DCN to contain tens or hundreds of thousands of servers. Many data center applications are data and communication intensive. For example, a simple web search request may touch 1000+ servers, and data storage and analysis applications may interactively process petabytes of data on thousands of machines. They define various traffic patterns such as one-to-one, one-to-all, and all-to-all communications.

A fundamental question for DCN is how to interconnect a large number of servers with significant aggregate bandwidth requirements. To this end, many research efforts have been spent on designing scalable and efficient DCN structures. Generally, the proposed structures include server-centric structures [2, 3], switch-centric structures [1, 4] and hybrid electrical/optical structures [5, 6]. In order to fully explore the network capacities within the physical structures, researchers have proposed a series of DCN routing protocols. Different from Internet routing such as OSPF/ISIS (link-state routing) or BGP (path-vector routing), most of the existing DCN routing schemes are specially customized to DCN topologies.

In this article, we survey the current state of the art of DCN routing algorithms. We review the new DCN structures and relevant basic routing schemes. We introduce opportunities with DCN traffic engineering. We discuss open questions on DCN multicasting. We explore potential security issues with DCN routing. We then conclude the article.

This work was supported in part by NSF CNS 0917233, and NSFC 60903182 and 60921003.

Basic Data Center Routing

Motivations and Challenges

Data centers are the foundations to support many Internet applications, enterprise operations, and scientific computations. They are large-scale and have data-intensive communications. The main challenge is how to build a scalable DCN that delivers significant aggregate bandwidth. On this question, research efforts such as BCube, DCell, PortLand, VL2, Helois, and c-Through [1–6] have been proposed in recent years. We first categorize and review the routing schemes within these structures. Then, we discuss some open questions.

Existing Solutions

Routing in Server-centric Structures — In server-centric DCNs, servers act not only as end hosts but also as relay nodes for multihop communications. Structures such as BCube [3] and DCell [2] fall into this category. To illustrate the routing, we use BCube as an example. The logic of routing in DCell and other server-centric structures is similar to that in BCube in that they all are performed by taking advantage of topological properties.

In BCube, servers are configured with multiple ports, and switches connect a constant number of servers. BCube is a recursively defined structure. A $BCube_0$ is simply n servers connecting to an n-port switch. A $BCube_1$ is constructed from n $BCube_0$s and n n-port switches. In BCube, two servers are neighbors if they connect to the same switch. BCube names a server in a $BCube_k$ using an address array $a_k a_{k-1} \ldots a_0$ ($a_i \in [0, n-1]$, $i \in [0, k]$). Two servers are neighbors if and only if their address arrays differ in one digit. More specifically, two neighboring servers that connect to the same level i switch are different at the ith digit. Based on this, BCube build its routing path by "correcting" one digit at one hop from the source to

0890-8044/11/$25.00 © 2011 IEEE

destination. Figure 1 is an example of a $BCube_1$ (n = 4) network. The routing paths from 00 to 23 can be 00-20-23 or 00-03-23.

Routing in Switch-Centric Structures — In switch-centric DCNs, switches are the only relay nodes. PortLand [1] and VL2 [4] belong to this category. Generally, they use a special instance of a Clos topology called Fattree to interconnect commodity Ethernet switches. The routing is based on this particular topology. Next, we show how the base topology is leveraged for routing.

Figure 2 is an example for PortLand and its address form. PortLand includes core, aggregate, and edge switches. End hosts directly connect to edge switches. Each end host has its actual MAC address (AMAC) and pseudo MAC address (PMAC). PortLand encodes topology information of a host into its PMAC with the form of *pod:position:port:vmid*. *Pod* (16 bits) reflects the pod number of an edge switch, *position* (8 bits) is its position in the pod, *port* (8 bits) is the switch-local view of the port number the host is connected to, *vmid* (16 bits) is for virtual machines on the same physical machine. For example, 00:00:01:02:00:01 means the third host in the first pod.

In PortLand, each switch knows its own position in the physical topology. PortLand switches forward a packet based on its destination PMAC address. Upon receiving a packet, the core switch inspects *pod* bits in PMAC to decide an output port. When an aggregation switch receives a packet, it first checks PMAC to see whether it is destined to a host in the same pod or a different one. If the same pod, the packet is forwarded to an output port according to position bits. Otherwise, it is forwarded to a core switch. Similarly, when an edge switch receives a packet, it decides to forward the packet upward or downward based on the topology information in PMAC.

Hybrid Electrical/Optical Structures — By using high-capacity optical fibers and optical switches, hybrid structures such as c-Through [5] and Helios [6] have been proposed. In these structures, ToRs (or Pods) are connected via an electrical part and an optical part (Fig. 3). The electrical part can be a switch or some switches forming a tree topology. The optical part is the focus of such structures and can dynamically changed according to real traffic patterns. At any moment, it is a bipartite graph with direct links assigned to heavy communication pairs via circuit. The routing in hybrid structures is straightforward: the optical part is a one-hop communication, and the electrical part is just a routing in a tree.

Comparison — Most DCN routing schemes are customized to specific topologies. Each structure has its own pros and cons. For example, switch-centric structures cannot directly use existing Ethernet switches and do not support various traffic patterns well (e.g., one-to-all, all-to-all). While server-centric structures like BCube require commodity switches and well support all-to-all traffic patterns, it relies on servers for relaying. In terms of complexity and power consumption, hybrid structures are better than pure electrical ones. However, the optical devices are expensive as they have not been widely used in data communications yet.

Open Questions and Design Guidelines

Automatic Naming — For existing DCNs, their addresses are based on their topology properties. The naming is a human process. An interesting question is how to automate this process with a machine. Recently, DAC [18] proposed a solution to automatically assign addresses for machines assuming such addresses for the whole network are known. But, what if such addresses are not known for a new DCN? To summarize, the open question here is to query an automatic process such that, given a DCN, it can automatically learn the topology and specify a suitable address space. The objective is that such an address space can be leveraged for efficient routing.

Generic Naming — In automatic naming, the goal is to find an automatic process to identify an address space for a DCN or a set of DCNs. One possible solution can be: First, categorizing the well-known structures. Then, given a DCN, the algorithm tries to match it to one category and do the naming accordingly. However, such method has limitation since it assumes the DCN belongs to the well-known structures. In practice, it is hard to enumerate all possible structures for DCNs. What if the DCN is totally new? This requires a generic naming approach that works for arbitrary structures.

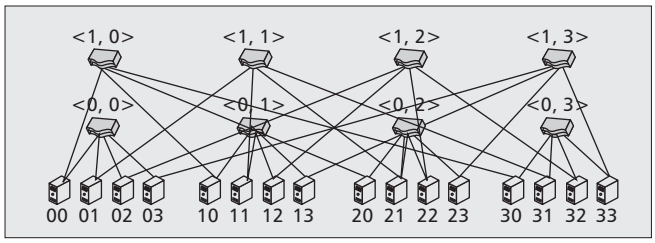

Figure 1. *An example of BCube and its address array.*

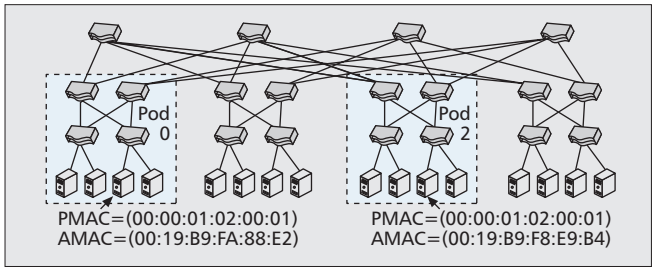

Figure 2. *An example of PortLand structure and PMAC address.*

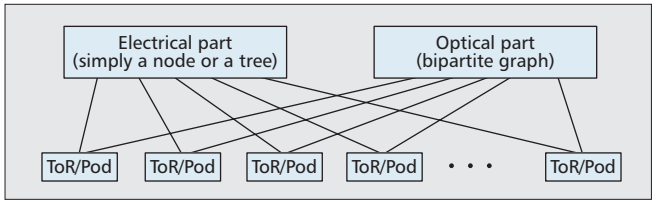

Figure 3. *An example of hybrid electrical/optical structure.*

Traffic Engineering

Motivations and Challenges

While most basic routing schemes seek routes between any two servers with short latency, a more sophisticated routing in DCNs requires further consideration and optimization in latency, reliability, throughput and energy, etc. Such kind of optimization is known as the traffic engineering (TE) problem. There are inter-DCN TE and intra-DCN TE problems. In this article, we focus on intra-DCN TE as most of the data center communication happens within the data center [6].

TE is well investigated for the IP/multiprotocol label switching (MPLS)-based Internet; however, the unique features of data centers have created several challenges. The traffic pattern is the most important input for TE problem, but

we still have very little knowledge about how to model and characterize the traffic matrix for TE in data centers. The traffic patterns can vary significantly for different applications, and the traces are usually confidential to data center owners. Meanwhile, the scale of data centers keeps evolving, and it is expected that hundreds of thousands of servers will be held in a single data center, so control of computation complexity and scalability is also quite challenging.

Existing Solutions

There are few mechanisms for DCN routing optimization nowadays. Equal-Cost Multi-Path (ECMP) routing is one way for optimization. However, ECMP does not check the load before path assignment, which may lead to temporary congestion on the paths. Valiant Load Balancing (VLB) is adopted in [6] to cope with the traffic load volatility, where each node selects path for each flow in a random manner. Both ECMP and VLB can lead to congestion when large flows are present. In [9], a central scheduler with global knowledge of active flows is employed to distribute the large flows on different paths. In [8], the authors pinpointed that the traffic pattern in data centers can be predicted only in a very short period so the TE should be performed in a fine-grained manner. In the context of green data centers, a recent work [10] proposed a dynamic adjustment on active links and switches to satisfy changing traffic loads. Their evaluation demonstrated a 50 percent power saving while still maintaining the network performance. With the motivation of per-service routing, people proposed symbolic routing and built the prototype named CamCube [7]. CamCube has a base multihop routing service using a link-state routing protocol exploiting the shortest paths. Meanwhile, it allows applications to adopt their own particular routing protocols to optimize the performance. The idea of symbolic routing is similar with overlay network and the base routing protocol and the customized routing protocols run as services in the servers.

Open Questions and Design Guidelines

A good data center TE solution should well utilize the unique features of data centers, instead of simply adopting the Internet TE techniques.

- First, the well-structured topology in data centers is an opportunity. Compared with the Internet, data center topologies are usually more symmetric and have more redundant paths between any two servers. Utilizing these topology properties to optimize the performance requires more efforts.
- Second, in a traditional TE problem, the locations of sources and destinations are fixed and the traffic is distributed among links. In data centers, it could have a new scenario that each physical server runs services using virtual machines (VMs). People can change the locations of services (or VMs) for a better performance.
- Third, data centers are appropriate places to exploit centralized TE. Usually, a single owner operates the data center, it is possible to collect all the link status and control the settings of all network components. This makes the central TE solution possible and also simplifies the implementation. In the meanwhile, the scalability on the central controller should be carefully studied.

In the design of data center TE, the following design principles should be considered.

- **Reliability.** Optimizing the routing so as to provide high reliability is the first motivation of data center TE [3]. Enterprises, service providers, and content providers rely on data and resources in their data centers to run business operations, deliver services and distribute revenue-producing content. Reliability is thus a concern for both providers and subscribers. Exploring the routing redundancy through multi-pathing, TE provides fault-tolerance and improves the robustness of data centers.
- **Load-balancing.** The second motivation is to better utilize the link capacity in order to tackle the trade-off between latency and throughput. A data center can run a large variety of applications and services, and a smart routing protocol should guarantee the performance of each application by efficiently utilizing the link capacity, e.g., distributing the traffic among the links inside the data center as evenly as possible.
- **Energy-efficiency.** Data center TE is motivated by the concern that data centers consume a huge amount of energy [10]. Besides the efforts on the energy-efficient hardware and software of servers, there is also space for energy saving at the routing layer. For example, by intelligently routing traffic through only a few active links and switches, a number of other links and switches can be switched off to save energy.

Multicast Routing

Motivations

Network layer multicasting benefits group communications in saving network traffic and releasing the sender from repeated transmission tasks. For modern product DCNs, especially the ones serving public services multicasting becomes useful and important, since it increases the capacity of DCNs so as to lower down operation cost and increase the competiveness in terms of providing lower $/VM.hour. More particularly, the traffic pattern of group communication is popular in both online applications and back-end infrastructural computation [11] hosted by data centers. Examples include redirecting search queries to multiple indexing servers, distributing executable binaries to a group of servers participating Map-Reduce alike cooperative computations, replicating file chunks in distributed file systems, and so forth.

Existing Solutions

Traditionally, multicast routing is difficult due to the need to resolve the optimum delivery tree a.k.a., the Steiner-tree building problem, which is NP-Hard. The challenges lie in the fact that the result should be obtained very fast. Many approximate solutions and standards have been proposed for delivery tree building in Internet multicasting, among which PIM is the most widely used one. In PIM-SM or PIM-SSM, receivers independently send group join/leave requests to a rendezvous point or the source node, and the multicasting tree is therefore formed by the reverse unicast routing in the intermediate routers.

Some other efforts have been paid to handle the scalability issue of IP multicasting. As we know, the large state requirements in routers make applications using a large number of trees impractical. No mechanism proposed to date would allow the IP multicasting model to scale to millions of senders and/or multicasting groups. For these reasons, IP multicasting is not, in general, used in the commercial Internet backbone [14]. Explicit Multi-Unicast (XCAST) is an alternate multicasting scheme to IP multicasting that provides reception addresses of all destinations with each packet. The XCAST model assumes that the stations participating in the communication are known ahead of time, so that distribution trees can be generated and resources allocated by network elements in advance of actual data traffic. In this sense, as for the case of modern DCNs, such challenge may look relatively easier to solve. Compared with the Internet, routing within DCNs is

usually with fewer hops and the topologies are usually well-structured. There should be ways to reduce the complexity of calculating the delivery tree.

On the other hand, new challenges may also appear. Different from the case of Internet routing, DCN usually comprises ordinary layer 2/3 switches and has much higher link density. Hence, one can hardly assume that the nodes are as strong or reliable as Internet routers as cost efficiency is the key concern of DCN operators. Meanwhile, due to the requirement of sustaining public cloud applications, usually a much larger set of multicasting groups should be supported simultaneously in DCNs.

Opportunities and Challenges

The characteristics of DCN multicasting actually lead to the following new challenges and open problems. Some survey shows that the ordinary commercially available access switch holds no more than 1500 multicast group state [12] and logically it is difficult to aggregate multicasting route entries since typically they are nothing to so with the topological information.

One cannot count on commodity switches for high availability and reliability as in the case of using high-end routers. Packet repair schemes should be developed against packet loss due to packets drops at switch buffer overload or temporal routing loop. Unlike the case of retransmission in unicast, packet repair in multicasting usually incur much higher overhead [13], thus efficiency and effectiveness are the keys behind.

Design Guidelines

As for improving the scalability of the routing system to support more multicasting groups, a possible direction is to leverage "in-switch bloom filters" or "in-packet bloom filters" or the combination of them. The idea of in-switch bloom filter is simply to use bloom filter as a multicasting group membership query mechanism on the switches/routers. The issue is that we need to give an entry to each and every switch/router interfaces/ports, which turns out to still occupy too much memory considering the scale of the mega datacenters.

The idea of in-packet bloom filter is to encode the multicasting deliver tree information into a bloom filter signature which will be placed within the packet headers. The switches can therefore determine how to forward/replicate the packet simply based on the encoded signature, so that there will be no need to maintain a huge multicast routing table within the switches/routers. The problem is that it incurs communication overhead and waste bandwidth.

As for efficiently improving the reliability of the data center multicasting when using commodity switches, the p2p packet fix/retransmission mechanism can be introduced. This releases the source node from handling numerous retransmission (removing the potential bottleneck) and improve the reliability. However, dedicated protocols and logical topology might be designed to make it real and practical. Load balancing among the peer receivers should be considered as well.

Routing Security

Motivations

Data center designs should envision providing superior resilience and security properties as an intrinsic consequence of good design principles, without needing additional add-on protocols or external checks to provide resilience to malicious attacks or misconfiguration. From a high level, securing the routing infrastructure necessitates securing both the control plane and data plane:

- Securing the control plane involves protecting topology discovery, by which nodes in the network can learn the genuine topology. Based on the learned topology, each node should be able to further perform path selection to select a correct path to send traffic.
- Given the correct communication path derived from the control plane, data plane security is mandatory to ensure data packets can be correctly delivered at each hop during forwarding.

Routing in data centers presents inherent the following uniquenesses compared to routing in the Internet, which lend both opportunities and challenges for securing the data center routing infrastructure in the presence of the above attacker model.

- Well-defined topology. Recent data center designs have unveiled the tendency of using particular physical interconnects and topologies to increase network throughput and scalability. Consequently, routing protocols are closely coupled with and even intrinsically derived from the topologies. For example, BCube [3] utilizes a hybercube-like topology and assigns node IDs in a specific way such that path discovery is automatically achieved by correcting one digit of the node IDs at each hop.
- Short flow life and high routing agility. A recent measurement study [4] indicates that most flows in a DCN are small (short-lived). In addition, the routing paths tend to be highly agile and volatile (commonly through the use of multipath load balancing) to provide desired load-balancing. For example, researchers in VL2 [4] propose to randomize forwarding paths across flows for load balancing.

Existing Solutions

Many secure routing protocols have been proposed in the context of both the Internet and enterprise networks. However, due to different network settings and characteristics, these protocols are either inapplicable to DCNs or result in suboptimal performance. For example, among the control-plane secure routing protocols, S-BGP employs heavy-weight asymmetric signatures to prevent prefix hijacking and path falsification attacks, while both attacks are substantially alleviated in DCNs with control-plane topology discovery tightly coupled with well-defined physical topologies. For data-plane security, most existing mechanisms require a long flow duration over the same path, which contradicts with the agility requirement of data center routing. For example, recent secure and relatively light-weight protocols [16, 17] leverage *low-rate packet sampling* or *approximate flow fingerprinting* to prevent packet modification attacks while reducing protocol overhead. However, these techniques result in long detection delays and thus require paths being monitored to be long-lived (e.g., after monitoring 108 packets over the same path in statistical fault localization in [16]).

Opportunities and Challenges

We observe that the uniquenesses of routing in DCNs open both opportunities and challenges for a secure routing protocol design, as detailed below.

Opportunities — Due to the underlying coupling between routing path discovery and the physical topology in current DCN designs, the control plane security in a DCN can be easily bootstrapped from the correctness of the physical topology formation. *As long as the physical topology of a DCN can be correctly established*, topology discovery can also be easily achieved because the physical topology is well-defined.

Challenges — Although securing routing control plane in DCNs can be simplified due to the well-defined physical topology, the large network scale, high traffic volume, and

high routing agility pose unique challenges in data plane security. More specifically, high routing agility and short flow life render great flexibility in data center routing. However, as a common tenet, there exists a fundamental trade-off between flexibility and ss ecurity. As mentioned earlier, existing secure data-plane protocols require a source node to know the exact path being used in order to perform monitoring and detection. In addition, these protocols require paths to be stable over a long period, which is an unrealistic assumption for DCNs.

Design Guidelines

In light of the above opportunities and challenges, we summarize a list of motivating (but by no means exhaustive) design guidelines of secure routing protocols in DCNs as follows.

- A good design should be tailored for the new (and somewhat simplified) attacker model, especially for the control plane. For example, due to the coupling of control-plane topology discovery and physical topology, certain long-standing routing attacks such as *wormhole* attacks which rely on falsifying neighboring relationship can be automatically eliminated, because the well-defined physical topology can automatically prune out false neighboring relationship. Not having to deal with these sophisticated attacks any longer, can the security protocol over-head be reduced.
- A good design should be *stateless* at the core switches to reduce the protocol overhead irrespective of the number of active flows, while existing data-plane secure routing protocols usually incur per-flow state or per-path which is impractical for data center networks with a large number of active flows [15].
- Finally, a good design should be *path-oblivious* in the sense that it should not require the source node to know exactly how its packets are being routed, due to the existence of dynamic multipath load balancing. Failing to meet this criterion renders most existing protocols inapplicable to DCNs.

Summary

In this article, we made a survey on the current state of the art of data center routing techniques. We discussed the opportunities and challenges of data center routing from different aspects, including basic routing schemes, traffic engineering, multicasting and security issues. We reviewed the existing schemes and also pointed out the open research issues with the hope to spark new interests and developments in this area.

References

[1] R. N. Mysore *et al.*, "PortLand: A Scalable Fault-Tolerant Layer2 Data Center Network Fabric," *SIGCOMM*, 2009.
[2] C. Guo *et al.*, "DCell: A Scalable and Fault Tolerant Network Structure for Data Centers," *SIGCOMM*, 2008.
[3] C. Guo *et al.*, "BCube: A High Performance, Server-centric Network Architecture for Modular Data Centers," *SIGCOMM*, 2009.
[4] A. Greenberg *et al.*, "VL2: A Scalable and Flexible Data Center Network," SIGCOMM, Aug 2009.
[5] G. Wang *et al.*, "c-Through: Part-time Optics in Data Centers," *SIGCOMM*, 2010.
[6] N. Farrington *et al.*, "Helios: A Hybrid Electrical/Optical Switch Architecture for Modular Data Centers," *SIGCOMM*, 2010.
[7] H. Abu-Libdeh *et al.*, "Symbiotic Routing in Future Data Centers," *SIGCOMM*, 2010.
[8] B. Theophilus *et al.*, "The Case for Fine-Grained Traffic Engineering in Data Centers," *INM/WREN*, 2010.
[9] A.-F. Mohammad *et al.*, "Hedera: Dynamic Flow Scheduling for Data Center Networks," NSDI, 2010.
[10] H. Brandon *et al.*, "Saving Energy in Data Center Networks," *NSDI*, 2010.
[11] Y. Vigfusson *et al.*, "Dr. Multicast: Rx for Data Center Communication Scalability," *Proc. ACM Eurosys'10*, Apr 2010
[12] D. Newman, "10 Gig Access Switches: Not Just Packet Pushers Anymore," *Network World*, vol. 25, no. 12, Mar 2008
[13] D. Li *et al.*, "RDCM: Reliable Data Center Multicast," *IEEE INFOCOM 2011*, Mini Conference.
[14] Wikipidea, http://en.wikipedia.org/wiki/Multicast.
[15] B. Theophilus, A. Aditya, and M. D. A., "Network Traffic Characteristics of Data Centers in the Wild," *IMC*, 2010.
[16] B. Barak, S. Goldberg, and D. Xiao, "Protocols and Lower Bounds for Failure Localization in the Internet," *Eurocrypt*, 2008.
[17] X. Zhang, A. Jain, and A. Perrig, "Packet-Dropping Adversary Identification for Data Plane Security," *ACM CoNext*, 2008.
[18] K. Chen *et al.*, "Generic and Automatic Address Configuration for Data Center Networks," *SIGCOMM*, 2010.

Biographies

KAI CHEN (kch670@eecs.northwestern.edu) is currently a Ph.D. student in the EECS Department at Northwestern University, Evanston, Illinois. Prior to this, he received his B.S. and M.S. degrees in computer science in 2004 and 2007, respectively, both from the University of Science and Technology of China, Hefei. He is interested in finding simple yet elegant solutions to real networking and system problems.

CHENGCHEN HU (huc@ieee.org) received his Ph.D. degree from the Department of Computer Science and Technology of Tsinghua University in 2008. He worked as an assistant research professor in Tsinghua University from June 2008 to December 2010 and is currently an associate professor in the Department of Computer Science and Technology of Xi'an Jiaotong University. His main research interests include computer networking systems, and network measurement and monitoring.

XIN ZHANG (xzhang1@cs.cmu.edu) is currently a Ph.D. student in the Computer Science Department at Carnegie Mellon University, working with Profs. Adrian Perrig and Hui Zhang. His research interests revolve around security and networking. Prior to joining CMU, he obtained his B.S from the Automation Department of Tsinghua University in 2006.

KAI ZHENG (zhengkai@cn.ibm.com) received his M.S. and Ph.D. degrees, both in computer science, from Tsinghua University, Beijing, China, in 2003 and 2006, respectively. He is currently working with IBM Research China. His research interests include high-speed packet forwarding, pattern matching associated network security issues, and new data center network architecture design.

YAN CHEN (ychen@northwestern.edu) is an associate professor in the Department of Electrical Engineering and Computer Science at Northwestern University. He got his Ph.D. in computer science from the University of California at Berkeley in 2003. His research interests include network security, and measurement and diagnosis for large-scale networks and distributed systems. He won the Department of Energy (DoE) Early CAREER award in 2005, the Department of Defense (DoD) Young Investigator Award in 2007, and the Microsoft Trustworthy Computing Awards in 2004 and 2005 with his colleagues. Based on Google Scholar, his papers have been cited over 2600 times.

ATHANASIOS V.VASILAKOS (vasilako@ath.forthnet.gr) is currently a visiting professor at the National Technical University of Athens (NTUA), Greece. He served or is serving as an Editor for many technical journals, such as *IEEE TNSM*, *IEEE TSMC—PART B*, *IEEE TITB*, *ACM TAAS*, and *IEEE JSAC* Special Issues in May 2009, and January and March 2011. He is Chairman of the Council of Computing of the European Alliances for Innovation.