Research on Building Large-scale Sensor Overlay Networks with Range Queriable P2P

Xun Shao, Masahiro JIBIKI, Yuuichi TERANISHI, and Nozomu NISHINAGA

In this research, we propose to realize sensor overlay networks by connecting their gateways with Skip Graph (SG), which supports retrieving sensing resources using their properties. To integrate proximity awareness into SG, we present the hierarchy-aware extension of SG (HSG) by integrating the hierarchy of the physical Internet into overlay construction and routing to improve latency and traffic locality. To alleviate the effect of flash crowds, we present a virtual-node based method to make nodes help each other automatically. With extensive simulations, we show that our proposals are efficient with little overhead.

1 Introduction

Nowadays, we can hardly do anything without the Internet. The Internet keeps growing, and its traffic volume is predicted to go beyond 10,000 times more than today's network by 2025. However, the Internet has several structural problems, such as redundant functions and compatibility issues due to the accommodation of additional features introduced to the network. If those problems remain unsolved, the Internet eventually will lose its function as a social infrastructure. For these reasons, we have been researching and developing the New-Generation Network (NWGN)[1], aiming to create a network that serves as a new social infrastructure with a life span of 50 to 100 years, free from the problems that today Internet has. In NWGN, the very large-scale information sharing network project is a very import component, which aims to realize a network treating over a trillion objects.

Owing to the recent technological advances on small, high performance, low power consumption CPU chips, memories with large storage and low cost, and high-speed mobile networks, we can soon expect the deployment of large numbers of autonomous sensor networks. Such networks can improve the overall utilization and convenience of social infrastructure. However, current Internet has no capacity to treat a huge number of objects, over one trillion by some estimations, distributed in the real world. Aiming at developing basic technologies for large-scale information sharing network platform that can treat a huge number of devices, we proposed a sensor overlay network structure[2]-[5] based on SG[6] which is a self-organizing, range queriable P2P technology. Figure 1 shows the architecture. The right part of the figure shows the architecture of one overlay node. An overlay node should contain at least three components: sensing resource, sensor data storage and computing resource. Users can be machines, end users, IoT (Internet of Things) service operators and even another sensor overlay node. In practice, owners of sensing resources can implement this architecture easily with smart gateways, Cloud computing and Fog computing technologies[7][8]. The left part of the figure shows the sensor overlay network developed by integrating separated sensor overlay nodes. The numbers near the gateways are the property of the sensing resources. The left lower part of the figure shows the overlay substrate based on SG, which is a structured range queriable P2P. Range query is a very important functionality as it allows users to find sensing resources without knowing their exact unique ID or IP address. For

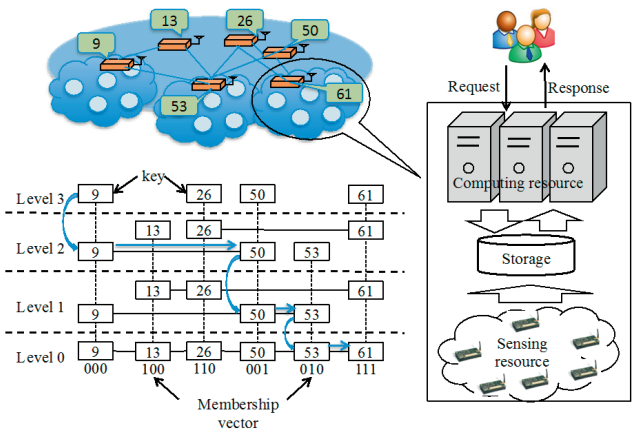

Fig. 1 Sensor overlay architecture

example, if an earthquake happens in a certain area, users can find the sensing resources they need with a query condition such as "sensors in the area within 100 km from the epicenter." In Section **2**, we introduced the functionality of SG in more detail.

Although SG is efficient in sensing resource retrieval, it suffers from a mismatch between the logical and physical topologies. In large-scale P2P systems, one hop between peers may cross multiple ASes, countries, or even continents, which would dramatically decrease latency and locality, thus affecting system performance. In fact, nodes in the Internet have natural hierarchical properties such as the ISP networks that they access and the countries in which they are located. The hierarchical properties can be used to estimate latency among nodes with coarse granularity. For example, the latency between nodes accessing the same ISP network is considered to be smaller, while the latency between nodes accessing different ISP networks is larger. Therefore, forwarding messages to nodes in the same ISP or AS can decrease the latency significantly. Furthermore, previous studies indicate that network connectivity failures are primarily due to Border Gateway Protocol (BGP) faults. As a result, nodes in the same ISP or AS tend to fail together. Therefore, improving routing locality will also help in fault isolation. To improve the SG for building a large-scale sensor overlay network, we propose HSG, the hierarchy-aware extension of SG. In HSG, we extended the routing tables of SG to include an extra entry called H-Entry, which can reflect Internet hierarchical properties. In the routing process, H-Entry is used with high priority for improving latency and traffic locality. We conducted extensive simulations with near practical scenarios, and the results show that HSG can improve both latency and locality with little overhead than SG.

Besides the topology mismatch, another inherent problem of SG is that the overlay nodes will suffer badly from flash crowds. Flash crowds can be characterized by a dramatic increase in requests for a service over a relatively short period of time[9]. The damage caused by flash crowds to sensor overlay networks is quite serious mainly for two reasons. The first reason lies in that a single node does not have as much computing and storage resource as high-end servers and clusters. The second reason lies in the key order preserving property of SG. If flash crowds happen, it is likely that several neighboring nodes would become hot-spot simultaneously. Consider that if key represents the position of the sensing resource, when an earthquake happens, all the nodes around the epicenter will become hot-spot. Our method to alleviate flash crowds for sensor overlay networks is based on an observation that when flash crowds happen, most of nodes are spare except the hot-spot nodes. In our approach, when a node identifies flash crowds, it sends sampling messages with specified TTL to collect the spare node information. After obtaining enough spare node information, it requires each of the qualified spare nodes to create virtual nodes with the same key as itself. The virtual nodes are then inserted into the overlay network as if they were normal nodes. According to the key order preserving property, all the virtual nodes are inserted around the hot-spot node in the overlay topology. The virtual nodes run the same processes and provide the same service to the users as the hot-spot node does. With this approach, not only the query service load of the hot-spot node, but also the query routing load of the nearby nodes of the hot-spot can be significantly distributed.

The rest of the paper is divided as follows: Section **2** gives a brief introduction of Skip Graph. Section **3** and Section **4** represent HSG and the virtual node-based flash crowds alleviation method respectively. In the last section, we conclude with a look at the future work.

2 A brief introduction of SG

In this section, we give a brief introduction of SG. SG is a distributed data structure for P2P applications to search for keys. Each node in SG has two fields: key and membership vector. Let $m(u)$ denote the membership vector of node u. The elements of $m(u)$ belong to a finite alphabet set π. We denote by p the inverse of the size of the alphabet; i.e., $p = \frac{1}{|\pi|}$. Theoretically, $m(u)$ is an infinite word over π; but in practice, only a prefix of length O(logN) is needed, where N is the total node number. There are multiple levels in SG, and nodes are grouped into increasingly smaller doubly lists within each successively higher level. The levels are ranging from 0. In every list, nodes are lexicographically sorted in their keys. At level 0, all nodes belong to a single. At level 1, nodes are separated into $\frac{1}{p}$ lists. Similarly, all nodes in a list of level i are separated into $\frac{1}{p}$ lists at level $(i + 1)$. The membership vector determines which lists a peer belongs to at each level. Denote the l-length prefix of $m(u)$ as $m(u) \uparrow l$, then two nodes u and v belong to the same list at level i iff $m(u) \uparrow l = m(v) \uparrow l$. The search algorithm of SG is as follows. At each node, the value of the query is compared to the node's key in the level where the query has arrived, and a decision is made

according to whether the query is to be sent to the left or right. It is then compared to the key of the next node. If it does not exceed the value of the query, the query is sent to that node. If the key of the next node exceeds the value of the query, the level is decremented by one and the process of comparing with the keys of neighboring nodes continues until the condition is satisfied. Figure 1 shows an example of SG with $p = 1/2$, which is the most common case in practice. In the figure, the rectangles represent nodes, and the number inside the rectangle is key. The binary sequence below each node is the node's membership vector, the prefix of which is used to group nodes into lists in a given level. Consider that we aim to retrieve key 61 from the leftmost node which holds key 9. As 61 is larger than 9, the query is sent to the right. We look for the destination of the retrieval at the 9's highest level (Level 3). Since no link exists, the level is lowered by 1 (i.e., level 2). In level 2, the key held by the right neighbor holding key 50 is referenced. Since 50 is smaller than 61, the query is sent to the 50. This process is repeated until node 61 is reached.

3 Hierarchy-aware Skip Graph (HSG) for effective sensing resource retrieval[3]

3.1 Hierarchical structure of nodes

As stated in Section **1**, the hierarchical structure is efficient to improve both latency and locality. An important issue in HSG is how to create the clusters and determine to which clusters a new node should be added. In this work, we clustered nodes in a natural manner to reflect the inherited hierarchy of the Internet. Figure 2 shows a simple example. "NTT-EAST" and "NTT-WEST" (two ASes in Japan) are clusters in the lowest layer (leaves), and they form a larger cluster "Japan". In the top layer (root), "Japan", "America" and some other countries form the "World" layer. A node belonging to "NTT-EAST" also belongs to the "Japan" and the "World" clusters. Nodes located in the same cluster are near each other in physical space. Based on this hierarchical structure, we define clustering distance to reflect the physical distance between peers. The clustering distance between two peers is defined as the maximum depth from the leaves where they are located to their lowest common parent.

3.2 System design of HSG

To reflect the hierarchical properties in SG, we extend the routing table of SG to make it contain two kinds of

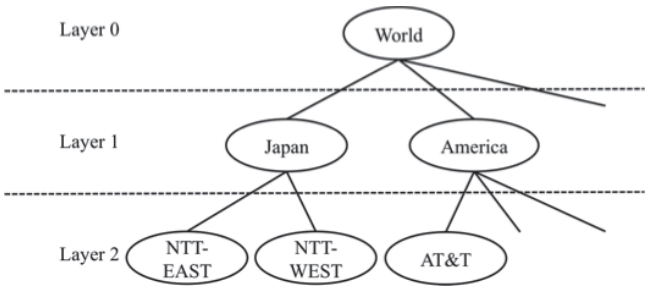

Fig. 2 The hierarchical structure of the Internet

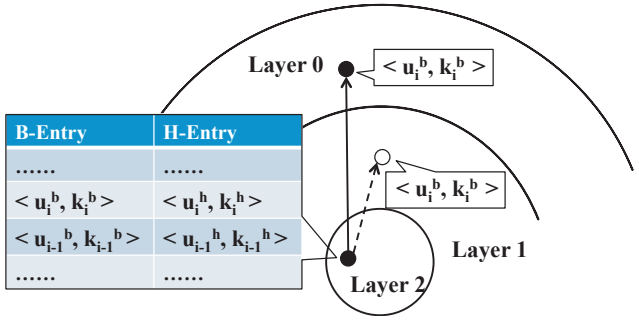

Fig. 3 Routing principle in HSG

routing table entries: B-Entries and H-Entries. B-Entries were extended from the basic SG routing table entries by adding the clustering distance from this node to the remote node. In any side of a given level of the HSG routing table, if the B-Entry is not empty, then there could be one or multiple H-Entries. The remote nodes referenced by the H-Entries are better candidates to forward messages than those in the B-Entry. The pointers in H-Entries must satisfy the requirements of both clustering distance and key values. We explain the requirements in Fig. 3.

Assume a node u with key k, its level i B-Entry references remote node u_i^b with key k_i^b, its level (i – 1) B-Entry points at remote node u_{i-1}^b with key k_{i-1}^b, and its level i H-Entry points at remote node u_i^h with key k_i^h. The level i H-Entry must satisfy the following two requirements. First, $k_{i-1}^b < k_i^h < k_i^b$, which guarantees the average search hop count is maintained O(logN). Second, the clustering distance from u to u_i^h must be less than the distance from u to u_i^b. In this example, the distance from u to u_i^h is 2 and 3 from u to u_i^b. If we assume $k_{i-1}^b < k_i^h < k_i^b$, the pointer in H-Entry should point at u_i^h. In this figure, we only show the routing table entries in the right side of N. The left side is similar to the key value requirements being the opposite. The routing process in HSG is similar to standard SG, except that H-Entries are used with higher priority than B-Entries.

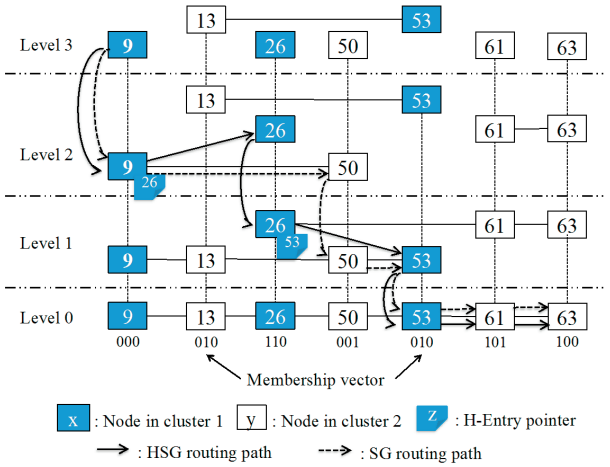

Fig. 4 A routing example of HSG

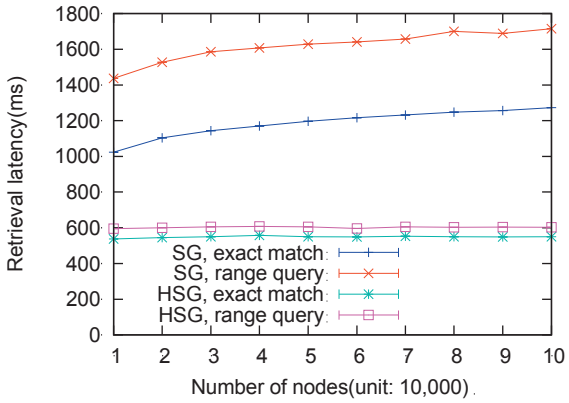

Fig. 5 Comparison of average retrieval latency

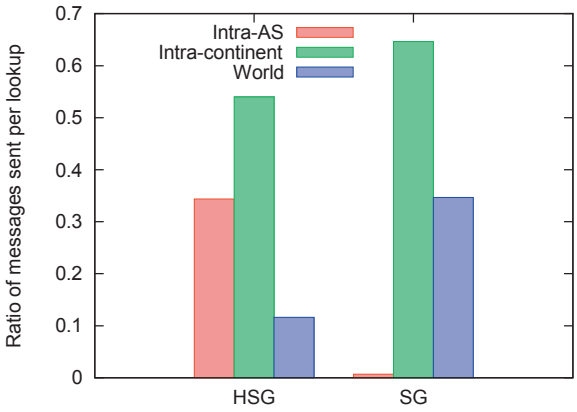

Fig. 6 Comparison of traffic locality

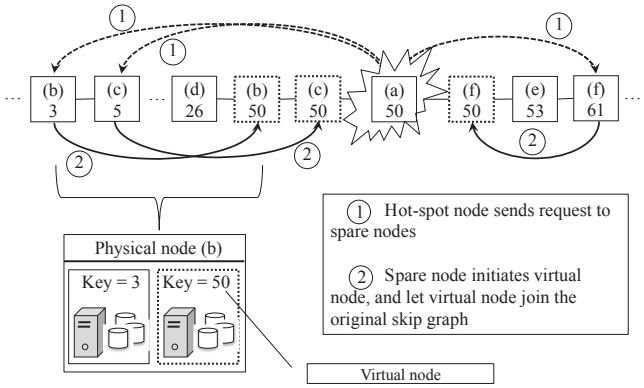

Fig. 7 Flash crowds alleviation strategy with virtual nodes

Take the query in Fig. 4 as example. In that figure, node 9 has a routing entry pointing at node 13 in level 0, a routing entry pointing at node 50 in both level 1 and level 2. With HSG, node 9 will reference node 26 with H-Entry. If node 9 searches key 61, the next hop will become node 26 instead of node 50.

3.3 Evaluation

In order to reflect the network characteristics of the Internet, we used the measurement data from PingER[10] project. PingER is one of several collaborative projects that have measurement infrastructures for monitoring Internet traffic. Using the ping command, monitoring nodes initiate transmissions to remote nodes, then measure and record the response times. In the simulations, we consider a three-tier hierarchical structure with the AS tier, continent tier, and world tier. Specifically, we considered a scenario of 7 continents, 500 ASes and nodes from 10,000 to 100,000.

In the first simulation, the number of nodes is varied from 10,000 to 100,000. For range query, the query range is fixed to 2,000. Figure 5 shows the retrieval latency of SG and HSG.

We can see that HSG outperforms SG significantly in both exact match query and range query. In addition, the latency of SG increases with the number of nodes, while the latency increase of HSG can be ignored. Also note that the difference between exact match query and range query of HSG is very little. As comparison, the latency of range query of SG is much larger than the exact query. Besides latency, HSG also supervises SG in traffic locality. Figure 6 shows that the intra-AS traffic ratio of SG is nearly 0. As comparison, HSG achieves significant traffic locality.

4 Relieving flash crowds for sensor overlay with virtual nodes[4]

In this section, we introduce our method to alleviate flash crowds, which is another inherent problem for sensor overlay networks based on SG and some other range queriable P2Ps.

4.1 The virtual node-based mechanism

Figure 7 shows the basic idea of the virtual node

mechanism. Generating virtual nodes involves two steps. In the first step, the hot-spot node sends requests to several spare nodes for help. If a spare node has enough available capacity, it initiates a virtual node with the key being equal to the hot-spot node, and let the virtual node join the original SG if it were a normal node with the standard SG joining algorithm.

In Figure 7, the solid rectangles represent physical nodes, and the dashed rectangles represent virtual nodes. The number in the node is key, and the letter in brackets is the physical node identifier. The physical node "a" has original key 3. After receiving the request from the hot-spot node with key 50, physical node "a" initiates a virtual node with key 50, and let it join the original SG as a normal node. All the virtual nodes are around the hot-spot node in SG and forming a virtual node zone.

Before further discussing the effect of virtual nodes, we first explain how could the hot-spot node find the spare nodes. It is known that the SG topology is an expander which has the property that random walks over the links converges very quickly to the stationary distribution. Since the SG topology is also a regular graph, the stationary distribution is the uniform distribution. This leads to a very simple algorithm for performing random sampling: the hot-spot node sends off a sample request message with O(logN) TTL, where N is the total number of the nodes. Every node along the path selects a random neighbor link, forwarding the request message and decrementing the TTL. The node at which the TTL expires sends back the sample. In the sample, there is (N-H)/N spare nodes in average, where H is the number of hot-spot nodes. If H << N, it is quite easy for the hot-spot node to find enough spare nodes. Now we show some analytical results on the effects of introducing virtual nodes. Let M be the size of the virtual node zone, which implies that there are (M – 1) virtual nodes and one hot-spot node. Node r_i with $0 \le i \le M$ represents the ith node in the virtual node zone. Note that r_i can either be a virtual node or be the hot-spot node. Consider a search from s with $s < r_o$ in the key ordering, we have the following two results:

Theorem 1. The probability that the search from s encounters each virtual node as well as the original hot-spot node is O(1/M).

Theorem 2. Let u be a node with $s < u < r_0$ in the key ordering, and the distance between u and r_o is d . Given that d $\sim M$, the probability that a search from s to the virtual node zone passes through u is O(1/(M + d)) . The details of the proof can be found in Reference [4].

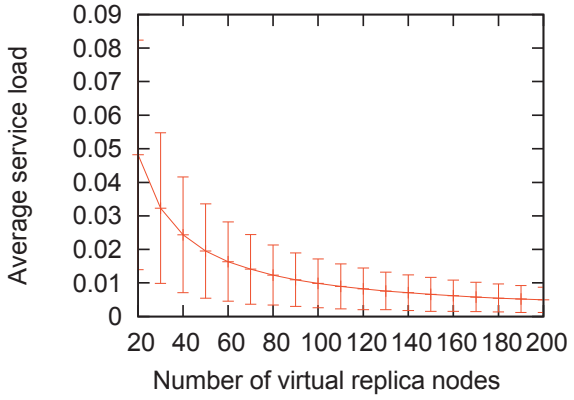

Fig. 8 Service load distribution

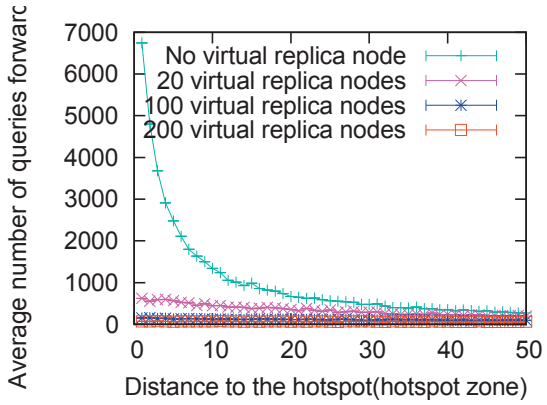

Fig. 9 Decrease in query routing load

4.2 Evaluation

In this section, we made extensive simulations to evaluate the proposed methods. In the simulations, we evaluated the query service load distribution among the virtual replica nodes and the hotspot node. We also studied whether our approach can lower the query routing load on the nodes near the virtual replica node zone. As SG is a probabilistic overlay network, we generated 200 SG with 10,000 nodes each, and all the results are the average of the 200 independent SGs.

Figure 8 shows the load distribution among virtual replica nodes as well as the original node. The horizontal axis is the number of virtual replica nodes, and the vertical axis is the ideal load ratio of each node. For example, if the number of virtual replica nodes is 20, the ideal load ratio of each node should be 1/20. In this figure, the standard difference between the real load ratio and the idea load ratio is shown with the error bar. We can see that our method works well in service load distribution.

We then study the query routing load on the nodes near the virtual replica node zone in Fig. 9.

In the figure, the horizontal axis is the distance between the nodes and the virtual replica node zone, and the vertical axis is the number of the request forwarded by the nodes. In this simulation, randomly chosen nodes sent 10,000 requests to the virtual replica node zone. We can see that without virtual replica nodes, the node nearest to the hot-spot node forwarded around 7,000 out of the 10,000 requests, while with only 20 virtual replica nodes, the number of requests forwarded by the nearby nodes of the virtual replica node zone decreases dramatically.

5 Conclusion and future work

In this work, we presented sensor overlay network based on SG, a kind of range queriable P2P, to integrate sensing resources distributed all over the world, and enable effective and complex queries. Aiming at realizing world-wide IoT systems, we addressed two inherent and critical problems of range queriable P2Ps: the mismatch between logical and physical topologies, and the flash crowds of queries. Simulations showed that our methods could reduce the retrieval latency by nearly 50%, and the effect of flash crowds can also be alleviated effectively. In the future, we are planning to validate our methods with field experiments.

References

1 NwGN project: http://www.nict.go.jp/en/nrh/nwgn/nwgn-randd-projects.html
2 Y. Teranishi, "PIAX: Toward a Framework for Sensor Overlay Network," in Proc. CCNC'09, 2009
3 X. Shao, M. Jibiki, Y. Teranishi, and N. Nishinaga, "A Low Cost Hierarchy-Awareness Extension of Skip Graph for World-Wide Range Retrievals," in Proc. COMPSAC'14, 2014
4 X. Shao, M. Jibiki, Y. Teranishi, and N. Nishinaga, "A Virtual Node-based Flash Crowds Alleviation Method for Sensor Overlay Networks," to be presented in Proc. COMPSAC'15, 2015
5 R. Banno, S. Takeuchi, M. Takemoto, and T. Kawano, "A Distributed Topic-Based Pub/Sub Method for Exhaust Data Streams towards Scalable Event-Driven Systems," in Proc. COMPSAC'14, 2014
6 J. Aspnes and G. Shah, "Skip Graphs," ACM Transactions on Algorithms, Vol.3, No.4, pp.1–20, 2007
7 M. Aazam and E. N. Huh, "Smart Gateway Based Communication for Cloud of Things," in Proc. ISSNIP'14, 2014
8 F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing andIts Role in the Internet of Things," in Proc. ACM MCC'12, 2012
9 J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash Crowds and Denial of Service Attacks: Characterization and Implications for CDNs and Web Sites," in Proc. WWW'02, 2002
10 PingER project: http://www-iepm.slac.stanford.edu/pinger/

Xun Shao, Ph.D.

Researcher, New Generation Network Laboratory, Network Research Headquaters
Distributed Computing, Overlay Network, Sensor Network

Masahiro JIBIKI, Ph.D.

Research Expert, New Generation Network Laboratory, Network Research Headquaters
New Generation Network, Information Centric Network, Very Large-scale Information Sharing Network

Yuuichi TERANISHI, Ph.D.

Research Manager, New Generation Network Laboratory, Network Research Headquarters
Ubiquitous Computing, Overlay Network, Multimedia, Database, Mobile

Nozomu NISHINAGA, Ph.D.

Director of New Generation Network Laboratory, Network Research Headquarters
New-Generation Network