WIDE Technical-Report in 2008

A Role-Based Peer-to-Peer Approach to Application-Oriented Measurement Platforms

wide-tr-mawi-ntaprole-00.pdf

WIDE Project : http://www.wide.ad.jp/

If you have any comments on this document, please contact to ad@wide.ad.jp

A Role-Based Peer-to-Peer Approach to Application-Oriented Measurement Platforms

Kenji Masui and Youki Kadobayashi

January 7, 2008

Abstract

The importance of large-scale measurement infrastructures for grasping the global state of the Internet is recently strongly emphasized. However, a fundamental analysis of these infrastructures has not yet been conducted. In this paper, we highlight the formation of measurement networks and provide a first look at measurement activities performed on those networks. We also propose a scheme for constructing a measurement network, which divides the measurement agent's roles into core agent and stub agent. This scheme entails only simple adjustment for changing the formation of the measurement network. Through the transition from a centralized system to hybrid and pure peer-to-peer networks, we visualize the flow of measurement procedures and explore the factors that have an influence on the overall performance of measurement systems.

Key words: peer-to-peer network, network measurement platform

1 Introduction

Large-scale network systems that include an overlay network application and a distributed computing environment have not yet fully succeeded in obtaining network characteristics on the Internet. Network characteristics are necessary information for sustaining and scaling the services of these systems. For example, an IP-level topology and round-trip time (RTT) information between two nodes are used as the metrics of the proximity among overlay nodes, and overlay network applications perform their optimization procedures based on these metrics. However, due to the complicated procedures of measurement and data processing, network characteristics are not yet widely utilized by the applications.

Given this situation, application-oriented measurement services [1, 2, 3] have been appearing. In these systems, measurement procedures are typically packaged into one independent network service, and applications need only issue a request to the systems in order to obtain network characteristics. Monitoring nodes are located in multiple administrative domains, and the systems manage and control them to obtain the requested data. These systems have emerged as a way for the applications to grasp the global state of the Internet.

This tendency has also brought a radical shift in the architecture of measurement systems. Traditional measurement infrastructures generally prepare a central management plane. In such systems, respective monitoring nodes perform measurement independently or according to the decision made by a central control plane. Then the system aggregates collected data in the central storage. This scheme worked well within the statistical observation of the Internet for mid-to-long-term. In the case of application-oriented measurement, measurement targets (e. g., nodes and links) disperse widely and change dynamically depending on the structure of application networks. In addition, collecting a large number of network characteristics with one control point causes a heavy load on specific nodes. For these reasons, the architecture of measurement systems has become more decentralized, and measurement methodologies performed in a decentralized manner have been studied. Some of these measurement methodologies are called "cooperative measurement," [4, 5] in which a monitoring node shares collected network characteristics and/or communicates with other agents for more efficient collection of network characteristics.

Though a variety of measurement systems has been proposed, and these systems focus on the sorts of network characteristics that they can collect, their architectures are yet to be sufficiently explored. We do understand the superficial indices, such as the capability and efficiency of measurement methodologies, that are implemented on the systems; however, we do not know which aspects of the architectures bring such results and how they influence actual deployment. This problem cannot be left unsolved before a large-scale measurement service is widely deployed, because such analysis could reveal fundamental drawbacks and advantages of the measurement infrastructures.

In this paper, we focus on the structure of a measurement network as one aspect of the architecture. The measurement network is a network in which measurement procedures are performed and measurement-related information is managed. One of the structures focused on is a centralized structure, where a specific management node manages all of the management information and controls the other monitoring nodes. Another is a pure peer-to-peer structure, where all of the monitoring nodes take partial charge of the management node's tasks. Moreover, we propose a hybrid structure, where some of the nodes work as management nodes and the others work as ordinary monitoring nodes. We explain how the respective measurement networks and the monitoring nodes in the networks actually work, and we investigate their basic characteristics related to their responsiveness and load distribution.

The rest of this paper is organized as follows: Section 2 describes measurement network models including a centralized model, a pure peer-to-peer model and our proposed hybrid model. We describe the experiment in Section 3, and in Section 4, we look into the experimental results and investigate the basic characteristics of the respective measurement networks. Section 5 presents a discussion on application-oriented measurement services and the formation of their networks, based on evaluation in the previous section. We refer to related work in Section 6, and finally conclude this paper in Section 7.

2 Measurement Network Models

In this section, we first define the components of a measurement network and how they work and interact with other entities. In Section 2.2, we describe two existing models of measurement networks — centralized and pure peer-to-peer models. We also refer to a hybrid measurement network model in the same section. Finally in Section 2.3, we propose a methodology to allow shifting a measurement network between these models, and we describe its implementation on an actual measurement system.

2.1 Components of the Measurement Network

A measurement network is a network in which measurement procedures are performed according to predefined sequences. Here we define the entities that appear in a measurement network and its operation.

The first entity is a "monitoring node," which performs measurement procedures in order to collect network characteristics. The second entity is a "management node," which is responsible for coordinating other entities so that the intended measurement can be performed. For example, the management node inspects and updates "management information," such as the list of monitoring nodes, and commands some of the monitoring nodes to perform measurement procedures. Collectively, we call a system that is composed of management information and management nodes a "control plane." A control plane is, so to speak, an entity where decisions for measurement procedures are made. "Control messages" are exchanged among the monitoring and the management nodes to achieve the intended measurement features. The control messages include a measurement command to the monitoring nodes and the node list in the measurement network, but do not contain the network traffic derived from the measurement procedures themselves. We note that one physical node may simultaneously play the roles of both management and monitoring. Figure 1 shows the relationship among the entities described in this paragraph.

2.2 Three Types of Models

Existing measurement networks are categorized mainly into two models — centralized and pure peer-to-peer models. In the centralized model, one management node or a cluster of replica nodes manages all of the management information and issues control messages to the monitoring nodes. On the other hand, in the pure peer-to-peer model, all of the nodes take the roles of both monitoring and measurement. Therefore each node has to maintain the measurement network and has also to perform the necessary measurement procedures. The merit of the centralized model is that the responsibilities of the respective nodes are clear, and it is easy to follow the sequence of measurement operations. At the

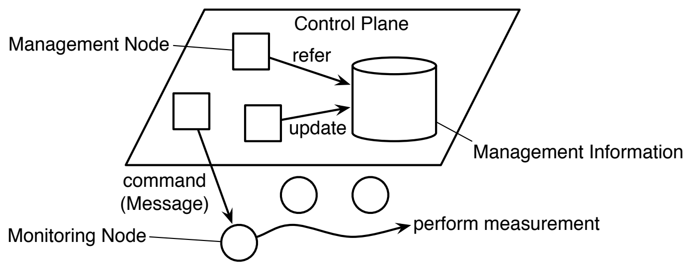

Figure 1: Components in the measurement network and relationships among them.

same time, a central management node has to tolerate a heavy load caused by all the management operations, otherwise the measurement system will not function. In the pure peer-to-peer model, we can distribute such loads to all nodes; hence this model is considered appropriate for a large-scale measurement system. However, a frequent change in the state of the measurement network, such as nodes joining and leaving, will result in poor stability of the control plane. These trade-offs are also discussed as a general problem existing between centralized and peer-to-peer systems.

As a middle course between these models, we now consider a hybrid measurement network model. In the hybrid model, management operations are divided among some management nodes, while other nodes behave as monitoring nodes. The difference between the hybrid model and the centralized model is that, in the hybrid model, multiple management nodes each perform a different management operation, whereas the management operations are not clearly divided in the centralized model even if there are multiple management nodes. By adopting this model, we can expect to moderate both the load concentration and the instability of the measurement network, which are the problems in the first two models. This model is similar to that of the Kazaa [6] network, in which stable nodes (called "super nodes") construct an overlay network in a peer-to-peer manner, and ordinary nodes join the overlay network through the super nodes.

2.3 N-TAP and its Extension

N-TAP[1] [7, 8] is a distributed measurement infrastructure that provides an application-oriented measurement service. In N-TAP, a program called an "N-TAP agent" performs measurement procedures. The N-TAP agents also construct a pure peer-to-peer measurement network (called the "N-TAP network") that is based on the technique of Chord [9]. In the context of Section 2.1, the N-TAP agent corresponds to a node that works as both a monitoring node and

[1] Available at http://www.n-tap.net/ .

a management node. The management information in N-TAP is stored and shared in a shared database that the N-TAP agents construct upon their peer-to-peer network. Besides the nodes (agents) list being maintained in the shared database as the management information, collected network characteristics are also stored in the same database so that the agents can share them in cooperative measurement. The N-TAP agents decide measurement tactics according to the "local-first and remote-last" rule, which improves the responsiveness to measurement requests from applications. In order to create a situation of a hybrid measurement network on an actual measurement system, we made some modifications on N-TAP.

The key idea of the extension to N-TAP is the division of the agent's roles into core agent and stub agent. The core agent, which corresponds to the management node, constructs the measurement overlay network, called the N-TAP network, as conventional agents did: it maintains its own routing table in the Chord ring and stores some of the shared data in a local database as a part of the shared database. The core agent also performs measurement as a monitoring node if necessary. The stub agent, which is equivalent to the monitoring node, does not perform the operations related to the construction of the N-TAP network. For joining the N-TAP network, the stub agent inserts its information in the shared agent list so that other agents can find it. It performs measurement only when a core agent sends a request to it or when it knows that the measurement procedures that are requested directly from applications should be done by itself. In the case that the stub agent needs to do the operations related to the N-TAP network, it sends a request to one of core agents, and the core agent responds to the request. For example, suppose that a stub agent wants to find a core agent that is responsible for a specified ID in the Chord ring so as to retrieve a shared data entry that has this ID; the stub agent asks a core agent to find the responsible agent, and the core agent performs the procedure of finding it. After the core agent obtains a result, it sends the result to the stub agent. In this way, even a stub agent, which does not perform the management procedures for the N-TAP network, can know the state of the N-TAP network.

By adopting the scheme of core and stub agents, we can also easily form the centralized and pure peer-to-peer measurement networks. Figure 2 shows the transitions of measurement networks according to the allocation of the respective numbers of core and stub agents. Now we have N agents, and C of N agents are assigned as core agents; i.e., $S\,(= N - C)$ agents are stub agents. The N-TAP network where $C = 1$ is equivalent to a centralized measurement network because all of the management information is concentrated in one core agent. If we take the value of $C = N$, all of the agents are core agents; therefore the N-TAP network in this situation is a pure peer-to-peer network, which is same as the conventional N-TAP network. In case of $C = i\,(2 \leq i \leq N - 1)$, we can regard the N-TAP network as the hybrid measurement network.

As described in this section, we can now have three types of measurement networks on the actual measurement system. In the following sections, we investigate the basic characteristics of these measurement networks.

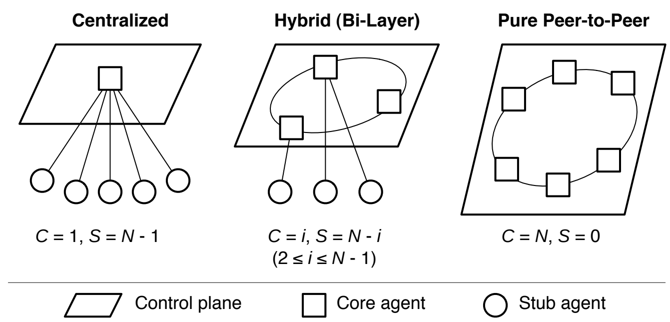

Figure 2: Measurement network formations with the scheme of core and stub agents ($N = 6, i = 3$).

3 Experiment

For this experiment, we used 128 homogeneous nodes in StarBED [10], which is a large-scale network experiment facility. Each node had an Intel Pentium III 1 GHz CPU, 512 MB memory and a 30 GB ATA hard drive. These nodes were connected through 100 Mbps Ethernet links in the same network. The Debian GNU/Linux operating system with the 2.6-series kernel was installed on each node.

We had one N-TAP agent run on each node; therefore we constructed a measurement network with 128 agents (i. e., $N = 128$). An N-TAP ID, which puts an agent in the Chord ring, was randomly assigned to each agent with no overlaps. The reason we chose random IDs was to distribute the load derived from maintaining the N-TAP network among the core agents in the hybrid and pure peer-to-peer measurement networks. After the N-TAP network was constructed, we ran a client program on one node that is in the same experimental network and did not have an agent. The program sequentially issued 2000 requests to one of the core agents for the RTT information between two randomly chosen experimental nodes. The program also issued the same number of the requests to one of the stub agents if the N-TAP network had stub agents. The request messages were exchanged based on the XML-RPC protocol between an agent and the client program. We note that an N-TAP agent usually tries to reuse RTT data previously collected and stored in the shared database if a client program specifies the request on the freshness of the RTT data. However, for simplicity in this experiment, we forced the agents not to reuse the RTT data but to perform the actual measurement. The agents logged their operations with time stamps, and N-TAP related packets were captured on the nodes, so we were able to analyze the behavior of the measurement network. We selected the values of 1, 2, 4, 8, 16, 32, 64 and 128 for C (the number of core agents) to shift a measurement network from the centralized one to the decentralized one. For convenience, we numbered the respective agents from 1 to 128 according to the following rules: (a) The first agent was a core agent that accepted and processed the above requests. (b) If there were other core agents, they were numbered from 2 to C. (c) If there were one or more stub agents, we set a stub agent that accepted and processed the above requests as the 128th agent. (d) If there were other stub agents, they were numbered from $C + 1$ to 127. Also note that the 128th stub agent was configured to issue a request related to the N-TAP network to the first core agent.

The procedures carried out by a core agent when it accepted an RTT measurement request are as follows (see [7, 8] for detailed operational flows on an N-TAP agent):

1. The core agent searches the source node in the requested RTT measurement. In this procedure, the core agent issues a request to find a core agent that is responsible for storing the data entries on the source node in the shared database. After it finds a responsible agent, it asks the agent to send the information on the source node (for instance, whether the source node is alive or not).

2. If the source node is alive (this condition is always true in this experiment), the core agent asks the source node to measure the RTT. Then the source node sends the measurement result to the core agent.

3. On receiving the result, the core agent responds to a client program with this result.

4. The core agent stores the collected RTT information in the shared database. It finds another core agent, one that is responsible for storing this data entry, and sends the entry to the responsible agent.

In the case of a stub agent, a control message to find a responsible agent was always sent to a specific core agent because the stub agent did not have a routing table in the N-TAP network but only knew the core agent that bridged between the N-TAP network and the stub agent itself. Apart from this messaging manner, the stub agent behaved in a same way as a core agent.

After the experiment, we confirmed that no measurement error had occurred and that all of the N-TAP related packets had been correctly captured during the experiment. The evaluation carried out in the following section is based on the recorded behavior of the agents after the measurement network became stable, i. e., no change in the agents' routing tables were made.

4 Evaluation

In this section, we investigate the basic characteristics of the respective measurement networks shown in Section 2. Our focus is the load distribution and the responsiveness to a measurement request in measurement networks.

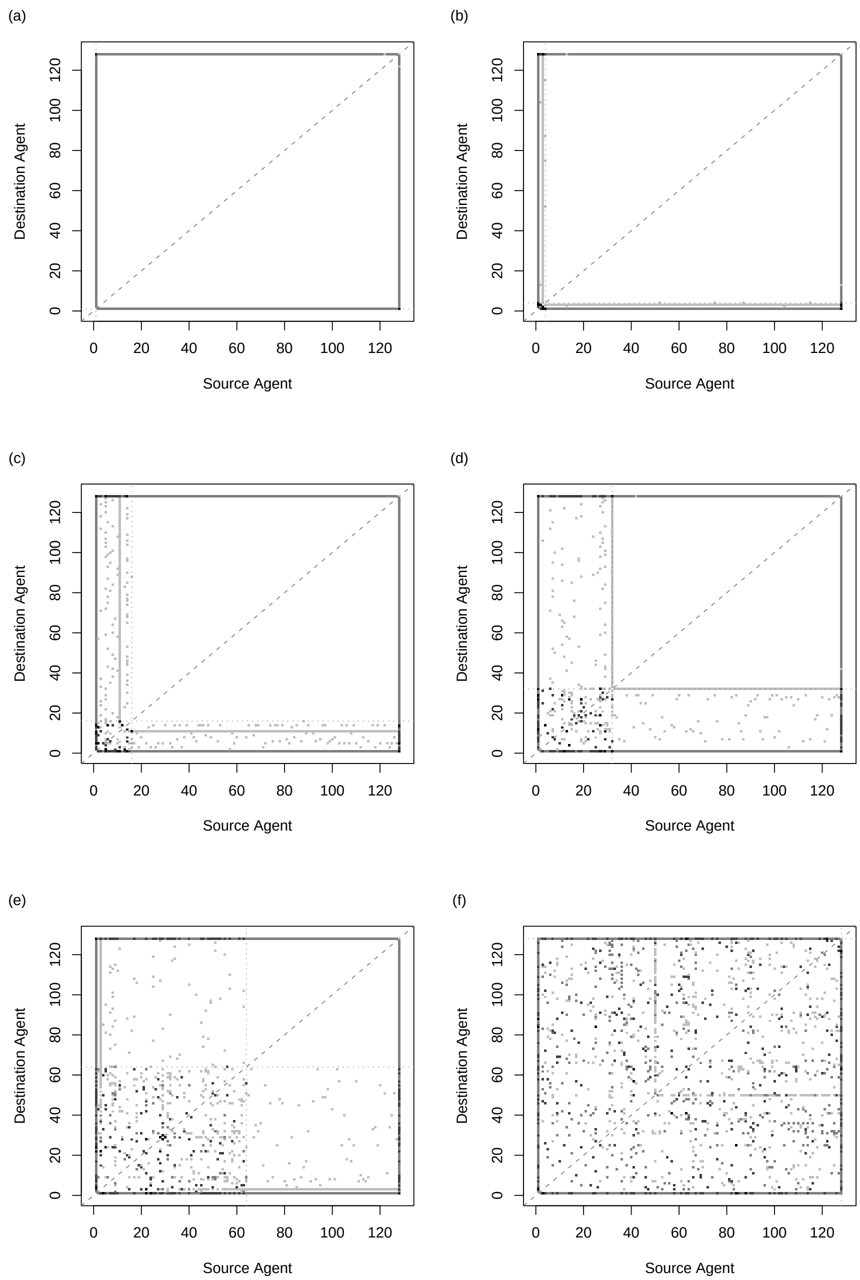

Figure 3: Distribution of exchanged messages among 128 N-TAP agents where (a) $C = 1$, (b) $C = 4$, (c) $C = 16$, (d) $C = 32$, (e) $C = 64$, and (f) $C = 128$.

4.1 Load Distribution

First we investigate the flow of control messages in the N-TAP network. Since the N-TAP agents have to carry out procedures according to the control messages, we can determine the distribution of loads among the agents by seeing this flow. Figure 3 depicts the distribution of exchanged control messages among the agents. Its horizontal axis denotes the assigned numbers of source agents in control messages, and the vertical axis denotes the assigned numbers of destination agents. The colored squares in the graphs show the number of the messages by their darkness: dark gray indicates more messages were exchanged and light gray means fewer. Specifically, where we define M as the logarithm of the number of exchanged messages, we divide the zone of the values of M into four even intervals and assign four shades of gray to the respective intervals so that the zone of the largest value of M is the darkest; a white area means that no message was exchanged between the agents. The horizontal and vertical dotted lines indicate the borders between the core agents and stub agents; therefore the bottom-left area shows the messages exchanged between two core agents, the bottom-right and top-left areas are for the messages between a core agent and a stub agent, and the top-right area is for the messages between two stub agents.

In any case, we can confirm that the squares are plotted more densely in the bottom-left area than in other areas, and the grays there are mostly dark. This shows that the burden of maintaining the measurement network was concentrated on the core agents, and the stub agents were relatively freed from such tasks. Additionally, no message was exchanged between two stub agents except for the cases of involving the 128th agent. The reason why the number of the messages to/from the first and 128th agents is large is that these agents had to ask other agents to perform the RTT measurement when they accepted measurement requests. For example, these agents asked the 10th agent to obtain the RTT between the 10th agent and the other agents. Moreover, after they obtained the RTT information, the first and the 128th agents had to store the measured RTT information in the shared database, as described in Section 3.

Secondly, we look into the exact number of exchanged messages and its tendency. Figure 4 shows the total number of exchanged messages during the 4000 requests in the respective cases of the C values. We can find that the number of messages exchanged between two core agents increases proportionally as the logarithm of the number of core agents grows. This number is zero where $C = 1$ because there is only one core agent and it does not need to issue a control message to another core agent. Meanwhile, the number of messages exchanged between a core agent and a stub agent changes slightly, though it becomes zero in the case of no stub agent ($C = 128$). The number of messages exchanged between two stub agents decreases as the number of stub agents decreases. The total number of messages tends to be larger as the number of core agents increases.

From these tendencies, let us see the number of messages that an agent of each role has to process as a metric of loads. It appears that the number of

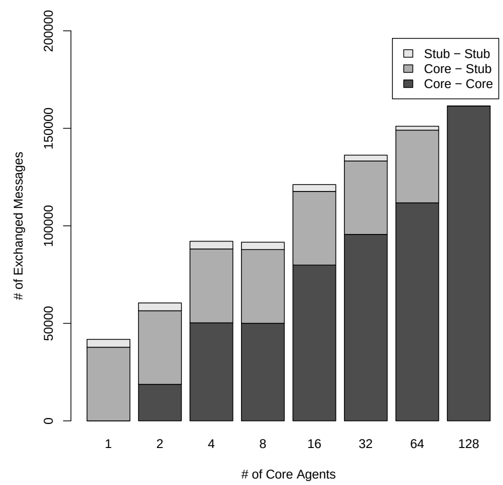

Figure 4: Number of exchanged messages.

messages that one core agent has to process is reduced when the proportion of core agents to the total number of agents is large, because the growth order of the summation of the core-core and core-stub messages is lower than that of the number of core agents. On the other hand, when this proportion of core agents is large, the number of messages that one stub agent has to process increases but is still smaller than the number of messages that one core agent has to process.

These facts indicate that the scheme of core and stub agents works just as we had intended, that is, that the loads should be distributed among the core agents, and the stub agents should have less burden. The maintainer of the measurement network can easily adjust the load distribution with the proportion of core agents as he or she intends.

4.2 Responsiveness

Next we compare the responsiveness to a measurement request in the cases of a request to a core agent and to a stub agent. Responsiveness is an important factor as an application-oriented measurement service, because it has an effect on optimization procedures performed by emerging applications that need network characteristics. In Figure 5, the boxplots that represent the distribution of the turn-around time for a measurement request are depicted. The left graph represents the turn-around time in the case that a client program issued requests to a core agent, and the right graph represents the turn-around time in the case of issuing requests to a stub agent. In both graphs, the horizontal axis denotes the number of core agents and the vertical axis denotes the turn-around time for one request. In the case of sending a request to a stub agent, the boxplot

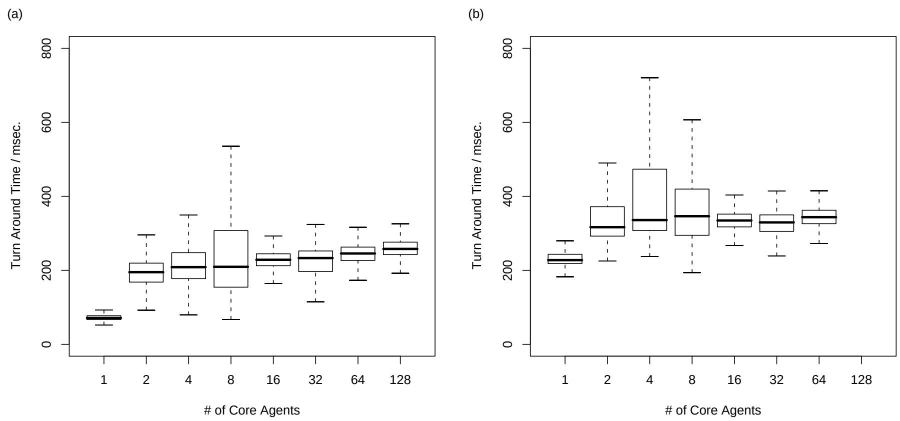

Figure 5: Turn-around time for a measurement request to (a) a core agent / (b) a stub agent.

where $C = 128$ is not given because we have no stub agent in the measurement network.

We can find that, in both cases, the turn-around time where we adopted the centralized model is shorter than the turn-around time with other models. The difference between the centralized model and other models is that the agent that accepts a request must perform the procedures for finding a responsible agent, retrieving the information on agents in the N-TAP network from the shared database, and requesting other core agents to store the collected data as an entry in the shared database. Inspecting the log files of the agents that accepted measurement requests from a client program, we calculated the mean of required time for each procedure, and the result is shown in Table 1. From this table, we see that, between the centralized model and the other models, a considerable difference of the time required for a measurement request is dominated by the time required for these procedures. The time required for finding a responsible agent is expected to increase linearly depending on the logarithm of the number of core agents. This is because, given the nature of Chord, the number of times a control message to find a responsible agent is forwarded among the core agents is proportional to the logarithm of the number of core agents. In Table 1, the required time for finding a responsible agent seems to follow this expectation. On the other hand, the required time for the database-related procedures would not significantly change while the size of local databases in respective core agents is small, and we can confirm such a tendency from the table. We also note that the distribution of the core agents' IDs also has an effect on the topology of the measurement overlay network, which results in the fluctuation of the required time, as in the above table. In this experiment, the IDs were randomly assigned;

Table 1: Required time for the procedures (in milliseconds).

The number of core agents ($= C$)	2	4	8	16	32	64	128
Find a responsible agent (core)	1.0	2.5	6.2	4.8	6.4	7.5	7.8
Retrieve from the shared DB (core)	5.5	5.3	7.6	3.5	3.4	3.0	3.0
Store in the shared DB (core)	52.7	53.7	49.4	58.2	55.5	60.5	62.6
Find a responsible agent (stub)	2.1	7.1	9.6	8.3	9.8	11.2	—
Retrieve from the shared DB (stub)	10.0	9.1	9.7	5.5	5.2	4.6	—
Store in the shared DB (stub)	56.5	58.5	55.0	60.3	56.4	61.5	—

therefore we suppose that the required time for these procedures is almost the same among the agents.

The required time for finding a responsible agent in the case of sending a request to a stub agent is longer in the order of a few milliseconds than in the case of sending a request to a core agent. This can be explained by considering that a stub agent first needs to send a control message to a core agent, while a core agent can send a message directly to the next hop's agent in its own routing table. We can suppose that this additional procedure for a stub agent increases the required time in the case of sending a request to a stub agent.

According to the discussion in this section, a measurement network with the centralized model is superior to one with the hybrid or the pure peer-to-peer model in its responsiveness to a measurement request. In this experiment, communication delay between the agents is short enough to be ignored, however, the communication delay will range approximately from tens to thousands of milliseconds when the measurement network is deployed in a wide-area network. This will have significant influence on the measurement network with the hybrid or the pure peer-to-peer model because a larger number of control messages must be exchanged through networks in these measurement networks. However, the centralized measurement network always has to struggle with load concentration at a core agent. These factors should be considered in constructing a measurement network.

5 Discussion

So far we have described the trade-offs among measurement networks with three different models based on the agents' behavior in respective networks. The centralized measurement network can get the best responsiveness in exchange for the heavy loads, which may bring a decrease in responsiveness. In the hybrid measurement network, we can select multiple core agents according to our purposes, and the processing loads can be distributed among the core agents. The load on one core agent will be the minimum on an average in the case of the pure peer-to-peer measurement network. However, in the hybrid and pure peer-to-peer measurement networks, the responsiveness will go down depending on the size of the control planes of these networks.

The ease of adjusting the formation of a measurement network will be important in the actual deployment of a measurement service. In this paper, we first proposed the scheme of core and stub agents in a measurement network. With this scheme, we can easily shift the measurement network among the centralized network, the hybrid network and the pure peer-to-peer network by adjusting the proportion of core and stub agents. In the case that we can control a measurement network (e. g., when we monitor network facilities with such measurement systems), administrators should design the measurement network to meet their requirements. They will benefit from the ease of adjustment to the measurement network. In the case that we cannot know beforehand what types of agents will join a measurement network, we cannot create a clear plan for constructing the network. One of the cases is that the agents run on the same nodes as the applications (an overlay network application, etc.), whose nodes will arbitrarily join and leave. Even in such cases, role-based adjustment will work with the application nodes. For example, in order to improve the responsiveness to a measurement request, we would choose agents that are connected with a high-speed link and have high performance as core agents. Other metrics, like the continuous running time of nodes, will also be helpful in constructing the desired measurement network.

Focusing on the application-oriented measurement service, quick responsiveness to a measurement request is indispensable in a measurement system. To improve the responsiveness in a hybrid or a pure peer-to-peer measurement network, some possible refinements of a measurement system can be pointed out. One is to let an agent cache the results of finding a responsible agent so as to decrease the number of exchanged control messages. From the results in Section 4.2, in a large-scale core network, we can expect that the required time for finding a responsible agent will become dominant in the turn-around time for a measurement request. Caching the results of this procedure will improve the responsiveness, but the agents will need to handle the inconsistency between the cache and the actual topology of a measurement network, and we will pay a waiting time penalty when such inconsistency occurs. Moreover, as described before, choosing core agents based on the capability of agents will also be effective. In the case of choosing core agents dynamically, we will also have to handle the migration of key-value pairs in a distributed hash table (DHT), which is expected to be a considerable burden.

6 Prior Work

Some application-oriented measurement systems have been proposed. The S^3 [2]'s network is similar to our hybrid measurement network in terms of having multiple roles for the entities in its network. On the other hand, considering that these entities are connected in a peer-to-peer manner, the S^3 network can be regarded as a pure peer-to-peer measurement network. iPlane [3] forms a centralized measurement network and provides a variety of network characteristics including an

IP-level topology, packet loss rate and available bandwidth. pMeasure [1] leverages the technique of Pastry [11] to form its own pure peer-to-peer measurement network and manage monitoring nodes in this network. In application-oriented measurement, the responsiveness to a measurement request is emphasized. To improve the responsiveness in these systems, an inference algorithm for network characteristics is sometimes utilized instead of performing actual measurement procedures. For example, iPlane estimates the RTT between two nodes based on an AS path. Alternatively, research efforts have produced effective measurement methodologies in large-scale networks called "cooperative measurement." As one example of the cooperative measurement methodologies, Vivaldi [4] lets us calculate the RTT between two nodes from their locations and distance in Euclidean space. Some researchers have adopted an approach of optimizing overlay networks for a specific measurement purpose. For example, MIND [12] focuses on the indexing and query processing in order to make its overlay network suitable for the distributed monitoring of anomalous traffic.

Other measurement infrastructures, e. g., DIMES [13] and NETI@home [14], whose main purpose is the statistical analysis of network characteristics, basically construct centralized measurement networks. They aggregate the collected data to a central server for performing their own analysis. These infrastructures do not need to consider responsiveness as strictly as application-oriented measurement services do. Hence the simple formation of a centralized measurement network seems to be suitable for analyzing the collected data.

In a hybrid peer-to-peer network, each overlay node is assigned one or more node roles and is managed in a hierarchical structure as described already in this paper. Kazaa [6], a peer-to-peer file sharing application, utilizes this scheme to connect between its unstructured peer-to-peer network and ordinary nodes. Though the details of its protocol and structure are not officially unveiled, some measurement-based work [15, 16] has already been done. The extension to N-TAP that we have added in this paper is unique in applying this scheme to a structured measurement overlay network in which measurement procedures different from the ones of ordinary file sharing applications are performed.

7 Conclusions

Analysis of the behavior and characteristics of measurement networks was an unexplored field. In this paper, we proposed a methodology for constructing a measurement network, which can easily change its network formation, alternating between centralized, hybrid and pure peer-to-peer models. By adopting this scheme and modifying an existing measurement agent, we investigated the operational flow in each of the measurement networks. As a result, we were able to confirm that exchanging control messages through networks has an appreciable effect on the turn-around time for a measurement request in the hybrid and pure peer-to-peer measurement networks. At the same time, the processing loads were successfully distributed among core agents in these networks. The consideration of such trade-offs is important in constructing a desired measurement network.

More measurement networks of a decentralized type will appear, and their importance will grow in the future, as large-scale network services and emerging applications are developed in the Internet. In further research and development of the N-TAP project, we aim to construct a practical measurement network that can provide network characteristics indispensable for these applications.

References

[1] Wenli Liu, Raouf Boutaba, and James Won-Ki Hong. pMeasure: A Tool for Measuring the Internet. In *Proceedings of the 2nd IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services (E2EMON'04)*, October 2004.

[2] Praveen Yalagandula, Puneet Sharma, Sujata Banerjee, Sujoy Basu, and Sung-Ju Lee. S^3: A Scalable Sensing Service for Monitoring Large Networked Systems. In *Proceedings of the ACM SIGCOMM Workshop on Internet Network Management (INM'06)*, September 2006.

[3] Harsha V. Madhyastha, Tomas Isdal, Michael Piatek, Colin Dixon, Thomas Anderson, Arvind Krishnamurthy, and Arun Venkataramani. iPlane: An Information Plane for Distributed Services. In *Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI '06)*, November 2006.

[4] Frank Dabek, Russ Cox, Frans Kaahoek, and Robert Morris. Vivaldi: A Decentralized Network Coordinate System. In *Proceedings of ACM SIGCOMM 2004*, August 2004.

[5] Benoit Donnet, Philippe Raoult, Timur Friedman, and Mark Crovella. Efficient Algorithms for Large-Scale Topology Discovery. In *Proceedings of ACM SIGMETRICS 2005*, June 2005.

[6] Kazaa. `http://www.kazaa.com/`.

[7] Kenji Masui and Youki Kadobayashi. N-TAP: A Platform of Large-Scale Distributed Measurement for Overlay Network Applications. In *Proceedings of the Second International Workshop on Dependable and Sustainable Peer-to-Peer Systems (DAS-P2P 2007)*, January 2007.

[8] Kenji Masui and Youki Kadobayashi. Bridging the Gap between PAMs and Overlay Networks: a Framework-Oriented Approach. In *Proceedings of the Eighth Passive and Active Measurement Conference (PAM 2007)*, volume 4427 of *Lecture Notes in Computer Science (LNCS)*, pages 265–268. Springer, April 2007.

[9] Ion Stoica, Robert Morris, David Liben-Nowell, David Karger, M. Frans Kaashoek, Frank Dabek, and Hari Balakrishnan. Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications. *IEEE Transactions on Networking (TON)*, 11(1):17–32, February 2003.

[10] Toshiyuki Miyachi, Kenichi Chinen, and Yoichi Shinoda. StarBED and SpringOS: Large-scale General Purpose Network Testbed and Supporting Software. In *Proceedings of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS 2006)*, October 2006.

[11] Antony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, November 2001.

[12] Xin Li, Fang Bian, Hui Zhang, Christophe Diot, Ramesh Govindan, Wei Hong, and Gianluca Iannaccone. MIND: A Distributed Multi-Dimensional Indexing System for Network Diagnosis. In *Proceedings of IEEE INFOCOM 2006*, April 2006.

[13] DIMES. `http://www.netdimes.org/`.

[14] Charles Robert Simpson, Jr. and George F. Riley. NETI@home: A Distributed Approach to Collecting End-to-End Network Performance Measurements. In *Proceedings of the 5th Passive and Active Measurement Workshop (PAM 2004)*, April 2004.

[15] Nathaniel Leibowitz, Matei Ripeanu, and Adam Wierzbicki. Deconstructing the Kazaa Network. In *Proceedings of the 3rd IEEE Workshop on Internet Applications (WIAPP'03)*, June 2003.

[16] Jian Liang, Rakesh Kumar, and Keith W. Ross. The KaZaA Overlay: A Measurement Study. *Computer Networks (Special Issue on Overlays)*, 49(6), October 2005.