第 XXIII 部

Integrated Distributed Environment with Overlay Network

第 23 部
Integrated Distributed Environment with Overlay Network

第 1 章 Activities of IDEON WG in FY2006

1.1 Introduction

IDEON (Integrated Distributed Environment with Overlay Network) is a working group of researchers who try to approach realization of the integrated distributed environment (IDE) through construction of overlay networks (ON).

IDEON focuses on research, development and operation of overlay networks as an infrastructure to realize free and creative rendezvous, location and routing. IDEON members are encouraged to actually live on the infrastructure to verify their designs.

Research topics of IDEON include, but not limited to, the following:

- Application-layer multicast
- Operable distributed hash tables
- Self-sustained trust management for distributed autonomous systems

For the detailed plans of specific projects, please refer to the web pages of each project.

- `http://member.wide.ad.jp/wg/ideon/?en%2FProjects` (English)
- `http://member.wide.ad.jp/wg/ideon/?ja%2FProjects` (Japanese)

Activities in IDEON can be outlined as follows:

$$\{understand \rightarrow build \rightarrow try\}* \rightarrow deploy \mid live$$

where '$*$' denotes repetition and '$|$' denotes concurrency.

Currently, all our development projects are in *deploy* phase. For example, it is our goal that the number of users of *wija* (messaging platform we have developed) and its major plug-in *i*-WAT will reach over 20,000 people by the end of March 2007.

1.2 Summary of Activities

This fiscal year, we have held two international workshops on dependability and sustainability of P2P systems:

- DAS-P2P 2006 (in conjunction with ARES 2006) in Vienna, Austria, April 2006
  - `http://das-p2p.wide.ad.jp/2006/index.html`
- DAS-P2P 2007 (in conjunction with SAINT 2007) in Hiroshima, Japan, January 2007
  - `http://das-p2p.wide.ad.jp/`

We have investigated topics including the following towards realization of dependable and sustainable infrastructure for free and creative rendezvous, location and routing:

- Local production, local consumption P2P architecture[207]
- P2P economics[244]
- Large-scale distributed measurement[137]
- Studies on routing algorithms for structured overlays[222]
- DHT-DNS hybrid naming system[57]

Table 1.1. Glossary for IDEON

Free	Having no restriction whatsoever as to with which peers one can communicate.
Creative	Being able to select a set of peers according to one's objectives, requirements, needs and contexts so that communication becomes most valuable for the participants.
Rendezvous	To identify such a peer.
Location	To locate such a peer in the overlay networks by the acquired identifier.
Routing	To deliver a message to such a peer on the acquired location.
Overlay Network	An application-specific virtual network of peers over the IP network to realize rendezvous, location and routing over an appropriate abstraction of entities.

1.3 Glossary

Table 1.1 shows the glossary for IDEON.

第 2 章 Local Production, Local Consumption Peer-to-Peer Architecture for a Dependable and Sustainable Social Infrastructure

This chapter attempts to put existing works of P2P designs into the perspective of the five-layer architecture model to realize LPLC (Local Production, Local Consumption), and proposes future research directions toward integration of P2P studies for actualization of a dependable and sustainable social infrastructure.

2.1 Introduction

Designs of P2P systems are characterized by their usage of overlay networks such that participants can potentially take symmetrical roles. This implies distribution of authorities, not only preventing introduction of single points of failure, but also possibly assuring the level of autonomy for self-organization, where any subsystem can spontaneously start, maintain itself, or recover from its failures.

The philosophy of Local Production, Local Consumption (LPLC), a useful concept for constructing a self-organizing system, states that what consumed locally should be produced locally, and when such resources are unavailable, they should be conveyed from the nearest producers.

LPLC is originally an agricultural term to promote sustainable local economy, which can enhance mutual understanding between producers and consumers (because they live near to each other), and has positive effects on local industry because the money spent by the local consumers do not exit from the region. But it has a wide range of applications in the designs of economic systems.

In a technological context of P2P, LPLC takes the form of emphasizing locality of communication as some systems such as Plaxton Mesh[194] or Tapestry[281] address.

As a total architecture not only for efficiency of communication, LPLC necessarily shows characteristics of P2P, because it promotes autonomy of subsystems in every scale. On the other hand, it is possible to design a P2P system not to show characteristics of LPLC, as it can be designed without considering the locality of communication. Therefore, LPLC is a strengthened concept of P2P.

2.2 Rationales for LPLC P2P

The concept of LPLC has quite an engineering value, and it makes more sense to design LPLC systems than just P2P. The principle is to use the nearest resource among multiple candidates if they are considered the same. Resources include atoms (physical goods), bits (information) and time slots (labors or CPU time) as illustrated in Figure 2.1.

Purchasing atoms Let us consider the case of purchasing oranges online.

We cannot evaluate the values of the oranges easily because we cannot take them on our hands or taste them. Trust with the venders or producers (for the sake of arguments, they are called *producers* altogether henceforth) is also a problem; it is possible that the producers do not send the goods to us after receiving money for them.

Perhaps if there are predecessors who have made transactions with the producers before, they could provide useful information. A reputation system will be useful.

Suppose we could successfully evaluate the qualities of the oranges. If there are several producers with the same grade of oranges, it makes more sense if we purchase them from the one nearest to us in the logistic network. Then the oranges will be fresher, and possibly cheaper because of smaller transportation cost. Moreover, if a road is closed somewhere because of some disaster, it is more probable that we can still purchase oranges from the nearest producer. LPLC is a rational

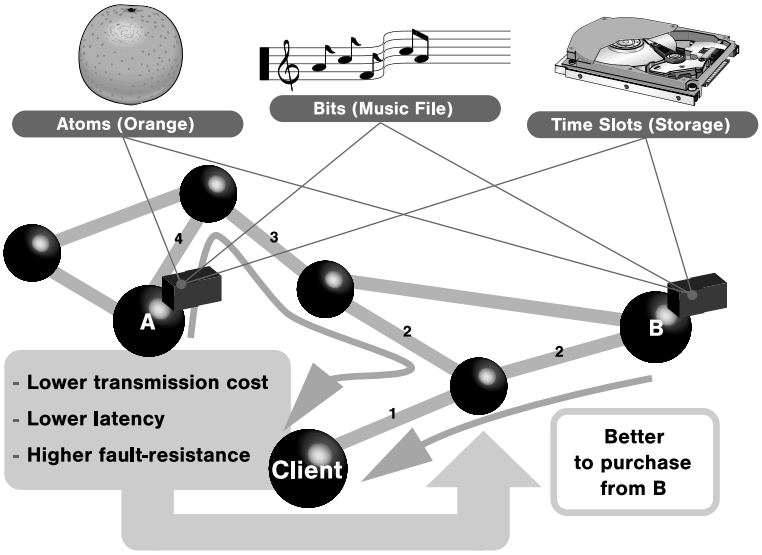


Fig. 2.1. Purchasing goods or services on a network

strategy in an economic sense.

Note that this is always a question of a distance within a (logistic) network, which does not necessarily equal the physical distance.

Purchasing time slots Let us then consider purchasing some storage service from another computer on a P2P overlay network. As it is the case for purchasing oranges, the quality of the services cannot be predetermined (availability of peers, performance of disks, etc.). Trust is also a problem because the peers might not give access to the storage after receiving the fee.

Perhaps if there are predecessors who have made transactions with the peers before, they could provide useful information. Again, a reputation system will be useful.

Suppose we could determine the quality of the storage services. Then it makes more sense if we purchase the service from the computer nearest in the communication path among the ones who provide the same level of services, as it will have less delay, less affected by faults because it passes through smaller number of failure points, and if the network is divided, more certain to be able to use the service.

Again, LPLC is more economical, and provides more survivability.

Purchasing bits Purchasing bits is another story because bits can be copied exactly at low cost, making the concept of *producers* ambiguous.

However, if there are the same music files on different paths from a consumer, for example, it will make more sense to access the nearest copy for the reasons we have just investigated. Some lookup systems, such as Tapestry, are designed to exploit this type of economy.

LPLC as a fundamental LPLC is also a fundamental concept underlying many of the hot topics of networking: *Ubiquitous computing* is about amenities being always provided by near-by entities. *Location-based information system* is about selecting entities by vicinity to provide information. *Sensor network* makes sensor information particularly useful for near-by entities. *Mobility* means that a sustained image of services is provided by a group of near-by entities even when the consumers move. *Ad-hoc networking* creates a field of information with near-by entities.

2.3 Five-layer architecture model for LPLC

The question now is how we design LPLC P2P systems. Another question is how close the P2P research community is to actualization of LPLC.

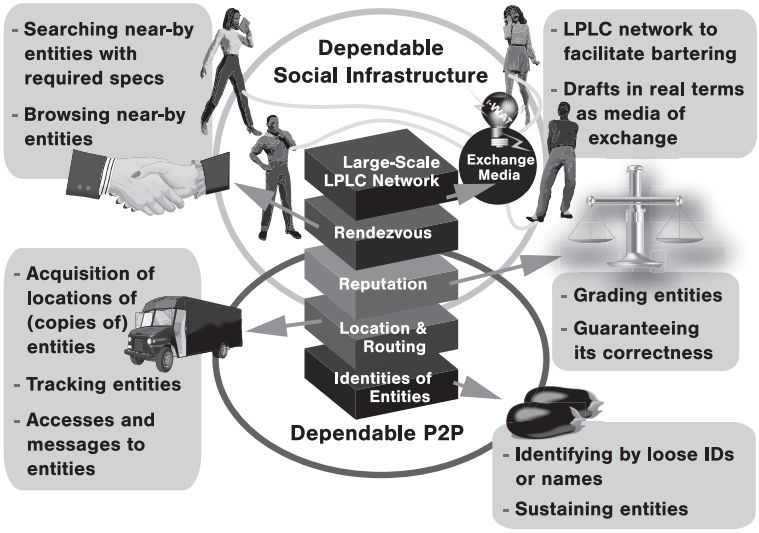


Fig. 2.2. Five-layer architecture model

Table 2.1. Five-layers and existing works

Layers	Existing research works
Large-scale LPLC network	BitTorrent[42], Samsara[46], MojoNation[11], Karma[260], Ripple[79], PPay[277], *i*-WAT[203]
Rendezvous	Peer Group Rendezvous[56]
Reputation	EigenTrust[116], Stamp trading[157]
Location and routing	Chord[239], Tapestry[281], etc.
Identities of entities	Freenet[41], Ecosystem of Naming Systems[58]

We have made the five-layer architecture model (Figure 2.2), into which we have attempted to put existing works of P2P designs as shown in Table 2.1.

The model is designed in a top-down way from the uppermost layer of *large-scale LPLC network*.

Some economic system is necessary to make a system usable in reality if we share time slots such as storage or bandwidth over a P2P overlay network. For example, BitTorrent[42] has a mechanism of choking the download bandwidth to peers who do not provide upload bandwidth.

In order for those facilities for purchasing or exchanging goods or services to work, one needs to determine what to purchase or exchange, by searching near-by entities satisfying required specifications (*rendezvous*). For such searches to work, grading entities in a distributed manner and guaranteeing its correctness are required (*reputation*).

For the reputation systems to work, as well as other basic communications and conveyance, we need to locate entities and forward messages to them (*location and routing*). To do so, we need the most fundamental mechanisms of identifying entities, assuring anonymity of them if necessary (*identities of entities*).

Currently, the focuses of the P2P research community are mostly on *location and routing* and *identities of entities*, into which many existing works fall.

We believe that we need to shift our focuses to the upper layers, and at the same time, redesign the lower layers based on the requirements from the upper layers.

To proceed, we need to determine how the uppermost layer is designed.

2.4 Designs of the exchange media

An LPLC network requires exchange media, otherwise it always has to require an unrealistic condition of *double coincidence of wants*[1]. The

1　A situation in which A wants what B can provide, and B wants what A can provide, as it is the case for the mechanism of BitTorrent. This is still useful in solving specific economic problems in P2P.

question we need to ask ourselves is how we design digital exchange media while bits can be easily copied (even if they are protected, people will find a way). An answer may be that some bits will not be copied easily (people will not want to), such as bits as a proof of debt of the holder.

Commodity money in LPLC network Autonomous economy, where participants can generate exchange media themselves, is not a new concept. In old Japan, rice was a common value being used as a medium of exchange (*commodity money*). In P2P overlay networks, such common values are rather found easily.

Samsara[46] is a fair P2P storage infrastructure in which each peer that requests storage of another must agree to hold a *claim*, or incompressible space, in proportion to their consumption. *Claims* can be forwarded along the chain of nodes that requests storage of another, eliminating themselves when cycles are found.

We would argue that *claims* can be forwarded in exchange with services other than storage, making them commodity money in the context of P2P (services and *claims* go the same direction). This is a possibility we would like to pursue, although some notable problems, particularly the efficiency of storage and bandwidth, may arise.

Drafts in real terms If rice is a common value, promise of a specific amount of rice of a specific grade can become a guarantee believable enough to work as money. This is the concept of *drafts in real terms*, which was a exchange medium used in Japan in around 12th–13th century. If we are to receive rice of the same grade in exchange of such a medium, the nearer the producer is, the cheaper the transactional cost is. Therefore this medium connotes the concept of LPLC.

i-WAT[203] is a protocol which can implement such *drafts in real terms* in an electronic way as described in section 3.2.

By issuing a digital *ticket* promising a certain grade of storage service, a peer in a P2P overlay network can purchase a service from another. Those *tickets* can be forwarded along the chain of nodes in exchange with services (services and *tickets* go opposite directions), eliminating themselves when they return to their issuers. This is another possibility we would like to pursue; actually, we have been experimenting with usages of *i*-WAT for several years.

2.5 Conclusions

LPLC, a strengthened concept of P2P, is a good philosophy for designing sustainable, cooperative (with nature), efficient, dependable social infrastructure that can support our lives.

We have made the five-layer architecture model to realize LPLC, and attempted to put existing research works in the areas of P2P so that we can set forth research agendas towards actualization of LPLC.

We introduced two candidates for the designs of exchange media in LPLC networks that can coexist: *claims* for storage, originally an idea from Samsara, and *i*-WAT *tickets* as *drafts in real terms*. We will investigate those possibilities further in the future.

第 3 章 Peer-to-Peer Economics for Post Catastrophic Recovery

This chapter proposes use of a distributed autonomous economic medium to support recovery of communities after catastrophic events.

3.1 Introduction

On December 26, 2004, a tsunami swept the coastlines of Southeast Asian countries, killing an unprecedented number of people, taking means of life away from millions.

Recovery from such a catastrophe requires a lot of money. That is the reason why many appeals were made for fund-raising after the disaster. But the problem may lie in scarcity of the medium itself.

Complementary currencies, or alternative forms of monetary medium, have been proposed and tested in real life to achieve an autonomous, sustainable local economy even in short of money. Many of these currencies fall into the category of MCS[213] (Mutual Credit System), in which participants freely credit one another, and the tradings are recorded in a single accounting system. These currencies can potentially help the disaster-affected economy to recover, because they impose smaller budgetary constraints.

There have already been efforts to pursue such possibilities, including those from *ccTsunami*[28], an open forum on the Internet for discussion and implementation of programs to support the tsunami-affected places.

Such programs need to be carefully designed not to impose excessive overhead or the communities' dependencies on others. We believe that requirements are as follows: 1) We need to build a mechanism so that anyone in the world can transfer funds to someone in an affected place safely and with certainty, 2) Such a mechanism, in a long term, should help the local economy to stand independently, and 3) It should require the smallest overhead as possible.

This section proposes an alternative to today's ways for raising funds or facilitation by MCS currencies — we will propose an alternative economics using *i*-WAT[203], an electronic descendant of the WAT System[268].

Sustainability of MCS has been in question because of their high operational cost. A simulation[205] has shown that growing the number of free-riders in MCS has a paradoxical effect of increasing "welfare" of the community. Since there is no pressure to stop the growth of the bad users, it is difficult to sustain the soundness of the system without strong interventions from the operators of the system. The same simulation indicated that *i*-WAT users can spontaneously sustain barter relationships even in the presence of free-riders by natural evasive actions to avoid risks, which makes it ideal for economics in post-catastrophic recovery, where autonomy is particularly important.

3.2 WAT/*i*-WAT currency system

The WAT System uses a *WAT ticket*, a physical sheet of paper resembling a bill of exchange, as the medium of exchange. A lifecycle of a WAT ticket involves three stages of trading (*the WAT Core*) as illustrated in Figure 3.1:

Issuing A *drawer* issues a WAT ticket by writing on an empty form the name of the provider (*lender*) of the goods or service, the amount of debt[2], the present date, and the drawer's signature. The drawer gives the ticket to the lender, and in return obtains some goods or service.

Circulation The person to whom the WAT ticket was given can become a *user*, and use it for another trading. To do so, the user writes the name of the recipient, as well as their own, on the reverse side of the ticket. The recipient will become a new user, repeating which the WAT ticket circulates among people.

Redemption The WAT ticket is invalidated when it returns to the drawer.

In case the drawer fails to meet their promise on the ticket, the lender assumes the responsibility for the debt. If the lender fails, the next user takes over. The responsibility follows the chain of endorsements (*security rule*). The longer the chain is, the more firmly backed up the ticket is.

i-WAT is a translation of the WAT Core onto the Internet. In *i*-WAT, messages signed in OpenPGP (*i-WAT messages*) are used to implement transfers of an electronic WAT ticket (*i-WAT ticket*). An *i*-WAT ticket contains a unique number, amount of debt and public key user IDs of the drawer, users and recipients. Endorsements are realized by nesting PGP signatures over canonical XML expressions.

Upon translating the WAT Core onto the digital

2 Typically in the unit kWh, which represents cost of producing electricity from natural energy sources.

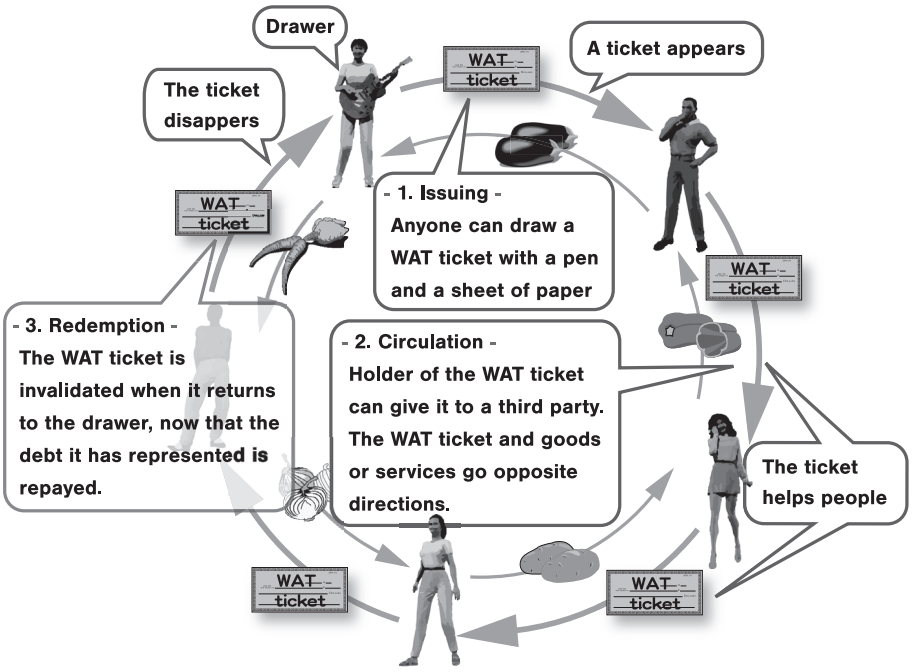


Fig. 3.1. The WAT Core

networks, we have made the following changes from the state machine of a WAT ticket: 1) Trades need to be asynchronously performed. Intermediate states, such as waiting for acceptance or approval, are introduced, and 2) Double-spending needs to be prohibited. The drawer is made responsible for guaranteeing that the circulating ticket is not a fraud. This means that every trade has to be approved by the drawer of the involved ticket.

The semantics of this design and the trust model of *i*-WAT are discussed in detail in [204].

3.3 Post-catastrophic recovery

Model The model is illustrated in Figure 3.2. It assumes the presence of an NGO installed at the disaster-affected place which facilitates reconstruction of the community. Its main function is three-hold: 1) Safely transfer funds to people in need in return with WAT tickets they issue, 2) Employ people for reconstruction work with the WAT tickets, and 3) Collect funds in hard currencies such as US dollar, euro or yen from the rest of the world in return with *i*-WAT tickets corresponding to the WAT tickets in their possession.

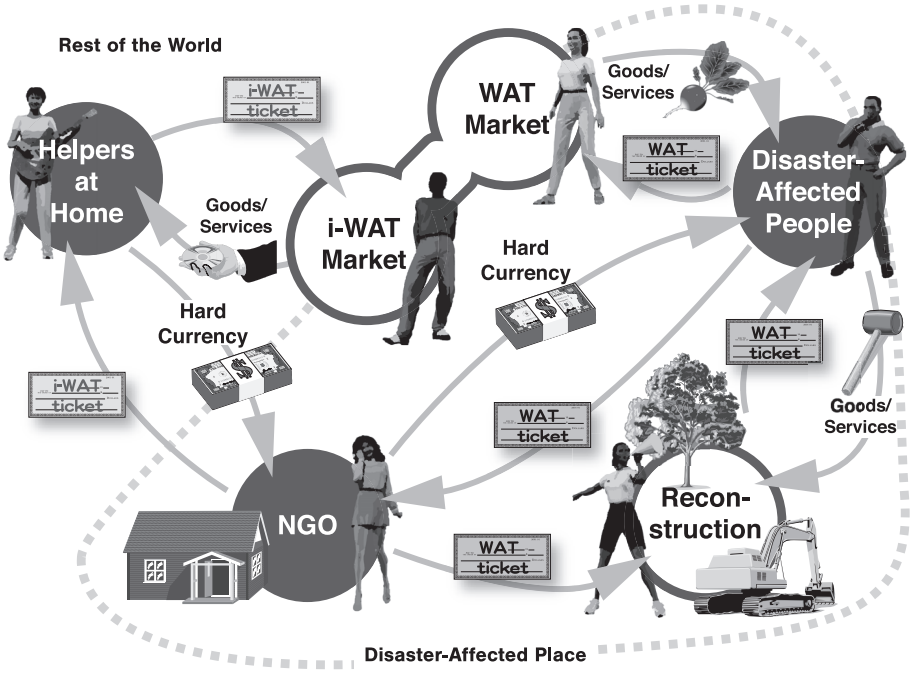
The WAT tickets represent the debt of the disaster-affected people, which can be reduced over time using the mechanism of ROT (Reduction Over Time[206]). Helpers in the rest of the world can assist this reduction by deferring the transfer of *i*-WAT tickets they have obtained.

Economics for the disaster-affected place The principle is that the local people work for the local people, creating few dependencies on the outer economics.

The people acquire means for life such as fishing instruments by issuing WAT tickets. By issuing the tickets, they promise that they will provide some goods or services when the tickets return to them in future.

In case the merchants do not accept WAT tickets, they can give the tickets to the NGO to receive funds in hard currency in return. The money is only transferred in exchange with some WAT tickets issued and presented by the people in the community. By knowing that there is a way to exchange WAT tickets with money, it will become easier for people to accept WAT tickets.

The NGO, by using the WAT tickets they



* WAT and *i*-WAT markets are connected by entities which issue one type of tickets by promising another (*exchange points*). The NGO in this figure is an example of such entities.

Fig. 3.2. Post-catastrophic recovery model

have obtained, can employ people in the affected place and its vicinities to work for restoring the infrastructure.

It is also possible to use *i*-WAT instead of WAT in the disaster-affected place where communication infrastructure has presumably been destroyed. Our reference implementation of *i*-WAT currently uses XMPP (Extensible Messaging and Presence Protocol), requiring rather consistent connectivity, but we have been investigating the possibility of applying CCS (Content Cruising System)[107] as a transport layer for conveying *i*-WAT messages, which will enable people to use portable wireless devices with ad-hoc channels where no Internet-connectivity is available.

Economics for the rest of the world The NGO issues *i*-WAT tickets which can be purchased on the Internet by the rest of the world. To do so, the NGO will need consistent connectivity with moderate bandwidth.

Those *i*-WAT tickets promise to give WAT tickets issued by the disaster-affected people. If someone in the rest of the world purchases one of those tickets, it means that they help some particular person, household or enterprise.

It can be seen that the NGO purchases the WAT tickets issued by people using the money the organization obtained in exchange with their *i*-WAT tickets.

Expected consequences The disaster-affected community in some day will be reconstructed. As the situation improves, people there will be able to pay back to the rest of the world with their working and their products. Their debt will have been considerably decreased with the help of the holders of corresponding *i*-WAT tickets by then.

The helpers in the rest of the world can use the *i*-WAT tickets they have purchased for trades on the Internet. Since each ticket represents debt owed by a disaster-affected person, household or enterprise, WAT/*i*-WAT unites the disaster-affected community with the rest of the world.

If people regard the NGO as an overhead, they can invent ways to get around it. Gradually,

the NGO will complete its role as a medium for spreading the idea of economics based on WAT/*i*-WAT. Possibly they will realize that the economics can sustain without hard currency — perhaps it will be the birth of a new autonomous economy where everyone can participate spontaneously.

3.4 Conclusions and future work

This section explained the use of WAT/*i*-WAT to support recovery of communities after catastrophic events. It proposed a model in which everyone in the world can help each other as peers. We are in the hope that those people unfortunately affected by disasters will consider WAT/*i*-WAT as an option for helping themselves.

For this to happen, we need to make an environment where the concept of WAT/*i*-WAT is readily accepted by the general public. We will start by spreading the idea that those barter currencies can help, through more experiments and applications on daily interactions among people.

第 4 章 N-TAP: A platform of large-scale distributed measurement for overlay network applications

To sustain a large-scale overlay network, knowledge about network characteristics is indispensable.

This chapter presents N-TAP, a distributed measurement infrastructure. N-TAP itself forms a measurement overlay network on which nodes can cooperate in measurement activities.

4.1 Introduction

In this section, network characteristics denotes the information that characterizes a network and its components such as nodes and links, with round-trip time (RTT) and IP topology being instances of network characteristics. For example, in the case of application layer multicast (ALM)[34, 35, 109], topology information enables an application to generate a distribution tree with the following purposes: preserving alternative routes in case of node failure, redistributing contents among proximal nodes, and so on. As P2P overlay applications become more popular, interest in discovering and exploiting large-scale network characteristics continues to grow.

Meanwhile, the measurement methodology for collecting network characteristics is shifting toward a distributed manner because the measurement capability of a single node is limited and the centralized aggregation of the results from isolated monitoring nodes is not often to scale, especially in the case of large-scale measurement. In distributed measurement, monitoring nodes utilize the data that other nodes collected, or communicate with others for cooperative measurement, then perform analysis and estimation to calculate network characteristics.

We propose a novel distributed measurement platform named N-TAP, which is equipped with shared storage for collected network characteristics.

Our motivation for proposing N-TAP is to enable application developers to handle network characteristics easily and to reduce the measurement cost on the Internet by abstracting measurement procedures into one independent service. Application developers today need to write lines of code for measurement, and this is often a significant burden on them. Moreover, this situation is a waste of resources (CPU, network bandwidth, etc.) because some types of required network characteristics are common among applications and portions of collected data may be reusable with other applications. It is also reported[161] that the measurement traffic, which is generated from experiments on overlay networks, consumes considerable amounts of bandwidth in PlanetLab[191]. If applications can obtain network characteristics from an integrated service, and if collected data can be shared and reused within the service, it will be possible to reduce the impact of these problems.

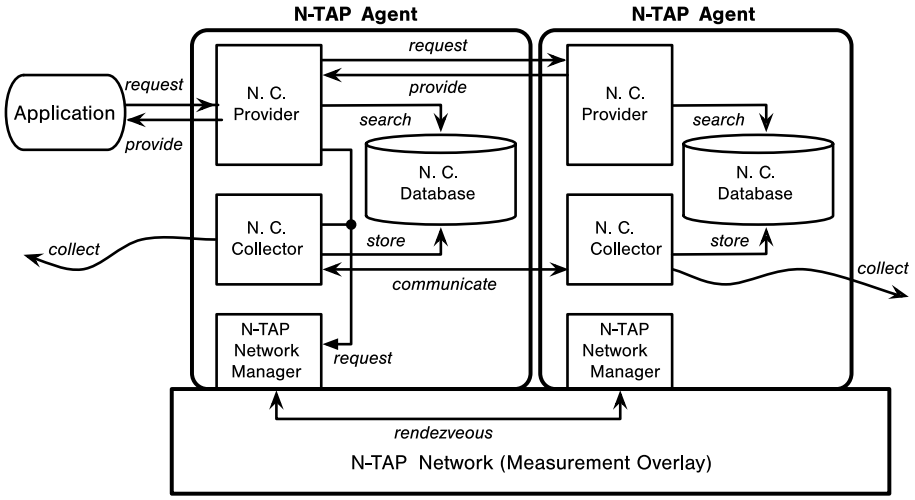


Fig. 4.1. Overall architecture of N-TAP

4.2 Requirements

We describe the requirements for N-TAP.

Cooperation N-TAP must prepare both a mechanism for communication among monitoring nodes and accessibility to the collected network characteristics on the overall system. This is because distributed measurement often involves communication among other nodes and the utilization of collected data, as stated in subsection 4.1.

Independence N-TAP must include an interface for exchanging messages with other applications to accept their requests and provide network characteristics to them. Moreover, the formats of messages and network characteristics must preliminarily be defined to ensure the portability of this system.

Decision making N-TAP has to interpret requests from applications carefully and make a decision about what action to take to collect network characteristics. For instance, topology data collected ten years ago does not represent the current topology and thus is worthless for an application that intends to reconstruct an overlay multicast tree. However, the RTT data that has been collected continuously during a week showing its periodicity would be useful for RTT-based proximity-aware node selection. As a consideration to the purpose of each application, we also need to take into account other factors such as measurement cost and response time.

4.3 Architecture and implementation

The system of N-TAP is composed of N-TAP agents, programs that run on monitoring nodes. We assume, in principle, that all the overlay application nodes, which use N-TAP, run N-TAP agents locally. However, this assumption is not necessary in cases that the required network characteristics do not require measurement from the local node.

As Figure 4.1 shows, N-TAP agent is divided into four components: a network characteristics database, a network characteristics provider, a network characteristics collector, and an N-TAP network manager. The network characteristics database is a repository of collected network characteristics. The network characteristics provider is an interface between an N-TAP agent and other applications. This component accepts the requests from the applications, and decides how to collect the requested network characteristics. The network characteristics collector performs measurement in order to collect the network characteristics requested by the network characteristics

provider. The N-TAP network manager is responsible for forming N-TAP network, the measurement overlay network among N-TAP agents. It also searches other N-TAP agents that other components (collector or provider) required for cooperation, and sets up a place for their rendezvous.

To reduce measurement costs and improve responsiveness, the decision-making process adopts the principle of locality-first. Initially, the network characteristics provider searches for network characteristics that satisfy the request from an application in the local database. If the requested data are not found in the database, it checks whether the agent can collect the data or not. If the agent can, it requests the network characteristics collector to collect the data; otherwise, it searches the data in the shared database. If the data does not exist in the shared database, by asking the N-TAP network manager, the network characteristics provider tries to find other agents that can collect the data. If one or more agents are found, the network characteristics provider makes a request to the other network characteristics provider that can collect the data. If not, it replies to the application that the requested data are unavailable. The provider that accepted the request from another agent performs the same decision-making process, but it does not forward the request to other agents and immediately replies in the case of unavailability. Through the simple steps described above, the network characteristics provider decides what to do to collect the requested network characteristics.

In our actual implementation of N-TAP, we utilize the Chord[240] technique, which is one implementation of a distributed hash table (DHT), to form an N-TAP network. N-TAP currently uses the 160-bit keys obtained from the SHA-1 cryptographic hash function. Each agent has its own agent ID that locates the agent in the Chord ring, and the length of the ID space is same as the keys.

N-TAP constructs its shared database for collected network characteristics by using the nature of Chord, for the reason that the storage cost, which will be enormous in the case of large-scale measurement, can be distributed among monitoring nodes. After an agent has collected the network characteristics, the item is stored in the node that collected the data as well as the (other) nodes that are responsible for the hash values obtained from the respective indices in the collected data, as the keys. The indices of each type of network characteristics are preliminarily defined in order to make the collected data accessible. For example, if node A collects the RTT between node A and node B, the data are stored in node A as well as the nodes that are responsible for $hash(IP(A))$ and $hash(IP(B))$ in the Chord ring, where $IP(N)$ denotes the IP address of node N and $hash(x)$ is the hash value calculated from a key x. As above, N-TAP forms the shared database where N-TAP agents can deposit and retrieve the collected data with some indices. Additionally, each agent deposits the information about itself into the shared database so that other agents can find the agent for cooperation. The agents use their IP addresses, netmasks, and fully qualified domain names (FQDNs) as the indices of the deposited information, thus the N-TAP network manager can search other agents by using these indices.

Regarding the protocol that applications use for making requests to N-TAP agents, we adopt XML-RPC[273] because of its widespread deployment and expressiveness. To request network characteristics, applications call the agent's methods for collecting the target data with specifying the type of network characteristics and certain conditions for the data. The collected data are stored with additional information such as time stamps, the ID of the agent that collected the data, its collection method, and so on. They are the criteria for judging whether the data can show the actual state of network entities, and the judgment depends on the conditions offered by an application.

Our current implementation of N-TAP contains the fundamental features presented above for meeting the requirements stated in section 4.2.

N-TAP agents now work on FreeBSD, Mac OS X, and Linux on PlanetLab. As the collection methods, current N-TAP can *ping* to measure the RTT between a monitoring node and another node, and can also *traceroute* to obtain the IP topology. Furthermore, one N-TAP agent can request other agents to *ping* or *traceroute*, and it is a primitive method of cooperative measurement from the standpoint that only one node cannot measure the RTT or the topology whose start points are not the measurement node. The implementation of existing methodologies for distributed measurement on N-TAP is still ongoing. Due to space limitations, we have skipped the details of the protocols, the format of collected data, etc. However, in future we also plan to release N-TAP software and documentation on N-TAP.

4.4 Discussion

From the aspect of our goal, the most important proposition is whether N-TAP really reduces a developers' burden and promotes the utilization of network characteristics. As stated in subsection 4.1, measurement is often a burden on overlay application developers. Our answer to reduce such burden is to prepare an independent measurement service that provides network characteristics to them. Compared to writing own codes for measurement, N-TAP will surely reduce the burden; they just need to specify what kind of network characteristics to collect and request N-TAP to collect them even if the measurement procedure that N-TAP performs is very complicated. However, such measurement framework would be worthless if N-TAP does not provide what they want; e.g., collectable network characteristics, measurement methodologies and their parameters. We need to continue the survey of such requirements and verify the proposition by actually implementing some applications that utilize N-TAP.

On the other hand, one of the challenges of distributed measurement we recognize is to explore a reasonable point of trade-off as an infrastructure for overlay network applications. There are some indices to characterize measurement methodology, but each index affects another as a trade-off. If agents attempt to extend the target where they collect network characteristics, due to the limitation of their measurement capacity they risk the possibility of losing timeliness. Timeliness is one of the most important factors because its increase will improve the reusability of collected data, which will in turn reduce the measurement cost on both agents and the overall Internet. Meanwhile, excess measurement activity often changes the primary characteristics of network entities. Therefore, we should study on a strategy of the collection.

Another challenge of N-TAP is extracting significant data from a massive amount of the collected data in a distributed manner. The entire N-TAP system can be compared to one huge shared repository for network characteristics. Each agent analyzes the data in the repository and makes the decision about what action to take in order to provide network characteristics to an application. Comparing with a centralized type of system, distributed analysis has the advantage of reduced analysis cost because each agent does not have to process all requests in N-TAP, only ones from a part of the applications in question. Meanwhile, if retrieval of the data that the agent uses for the analysis takes a long time, this makes turnaround time to the application longer. We need to verify whether N-TAP can process the request within an acceptable timeframe under such a situation.

4.5 Related work

Some has already been conducted on the methodology of distributed measurement. Vivaldi[48] is a decentralized coordinate system based on a physical mass-spring system, and can estimate the RTT between two nodes with few measurements. GNP[174], NPS[173], PIC[45], and Lighthouse[190] are also included as coordinate systems. Meanwhile, Doubletree[59] is an algorithm that reduces the cost of traceroute by

exploiting the common portion of IP topology. By deploying these methodologies on N-TAP, applications can take advantage of them and researchers will have opportunities to perform their experiments on an actual network environment.

Projects for collecting network characteristics such as CAIDA[22] and DIMES[55] also exist, and they are basically working on scientific and statistical analysis. Though their objectives are slightly different from the one of N-TAP, the fundamentals of their studies, such as the style of infrastructure and the analysis methodologies, will be informative for N-TAP, too.

4.6 Conclusions

This chapter presented the architecture and the implementation of N-TAP, a platform for large-scale distributed measurement with the consideration for overlay network applications.

We believe that such an infrastructure is indispensable for sustaining application networks and promoting their continued growth even though we need more proof from both theoretical and practical aspects. To guard against the excessive traffic derived from the activities of overlay network applications, our system will contribute toward solving such problems. Furthermore, we hope that our system aids application developers in handling network characteristics, and accelerates the studies on the methodology of distributed measurement.

Future work will involve continuing research on the validity of distributed measurement for overlay network applications on actual network environments.

第 5 章 A Comparative Study of Iterative and Recursive Lookup Styles on Structured Overlays

We have conducted a comparative study of iterative and recursive lookup styles in structured overlays. We discuss pros and cons of each lookup style with respect to delay, efficiency, and resiliency to attacks. As a result, we have clarified that one lookup style cannot satisfy all requirements and the lookup styles have contradictory characteristics.

5.1 Introduction

The methodologies for constructing and maintaining overlay networks are divided into two categories: *unstructured* and *structured* overlays. The former is represented by content-sharing application Gnutella and Winny's searching network. It does not impose any constraints upon nodes for choosing their neighbors. On the other hand, the choice is systematically defined for the latter. As an application for the latter, searching methodologies in the form of Distributed Hash Table (DHT)[197, 202, 239, 281] and multicast methodologies[26, 27] have been proposed. This section deals with structured overlays.

For structured overlays, an abstraction based on routing is possile[49]. According to this abstraction, applications such as searching and multicast can be implemented by routing and transferring messages, where identifiers are allocated for objects of search, multicast groups and overlay nodes, and nodes are searched by the range of identifiers of which they are responsible. Identifiers can for example be 160 or 128 bit numbers.

There can be variations in patterns of internode communication on structured overlays even if the routing algorithms or routes are the same. Such communication patterns include iterative and recursive lookup styles. The objective of this section is to compare those and clarify their differences.

5.2 Iterative and recursive lookup methods

Routing on a structured overlay, or searching for a responsible node, requires a specification of the target ID. In the end, the search request reaches the node responsible for the ID. If the application requires a response, the obtained route or reply

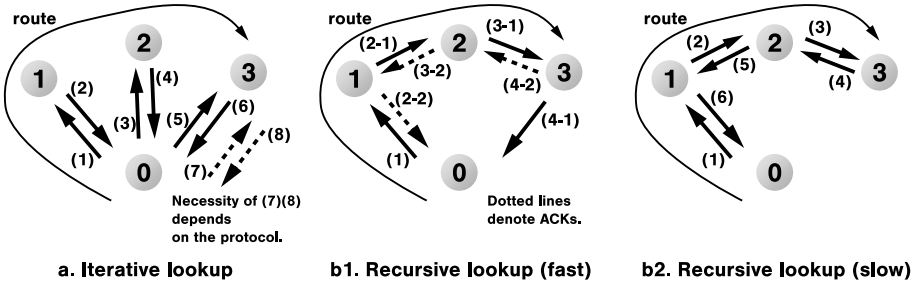


Fig. 5.1. Lookup styles (route length $n = 3$)

(some value in the case of a DHT) is returned to the requester. This search is achieved by mutual communication among nodes in the discovered route including the source and the destination.

Unlike Internet Protocol, where a packet is forwarded from the source to the destination at which the transfer completes, searching a responsible node on a structured overlay can have either of the two communications patterns: *iterative* and *recursive* lookup[198] styles[3].

In iterative lookup, the requester queries each node on the route by themselves, determining the next hops. In recursive lookup, on the other hand, each node on the route forwards the search request in a similar manner as routing on the Internet. Figure 5.1 illustrates their communication patterns.

Recursive lookup has two communication patterns depending on how the response is returned to the requester. In order to reduce the number of messages, it is effective for the target to directly return the response to the requester. But it is also meaningful to return the response by tracing the route back. In reality, both methods have been used. We distinguish these two by calling them recursive lookup (fast) and (slow), respectively.

Table 5.1 shows which implementations of structured overlay adopt which lookup styles. Table 5.2 shows the number of hops and messages for each method where the route length is n.

Table 5.1. Adopted lookup styles

Implementations	Styles
Bamboo[198]	Recursive (false)
Chord[239]	Iterative
MS Pastry	Recursive
Overlay Weaver[183]	Iterative, Recursive (fast)

5.3 Summary of analysis

We have analyzed the differences among iterative and recursive (fast/slow) lookup styles by certain technical metrics. The metrics included delay, efficiency, ease of implementation and attack-resistance.

Our discussion has made the following clear: each style has its advantages and disadvantages, and it is difficult for one sytle to satisfy all requirements. For example, if performance has a priority, recursive lookup (fast) is optimal in general. However, depending on the distribution of data and queries in DHT, it is possible that

Table 5.2. Number of hops and messages for route length n

	Iterative	Recursive (fast)	Recursive (slow)
Returns a response (e.g. DHT)			
Hops	$2n$ or $2(n+1)$	$n+1$	$2n$
Messages	Same as above	$2n+1$	$2n$
Does not return a response (e.g. message delivery)			
Hops	$2n-1$	n	n
Messages	$2n-1$	$2n$	$2n$

3 They are often called iterative/recursive *routing*, but we do not use the words because routing should denote construction of a route information.

recursive lookup (slow) is faster in some cases, because of caching effects and so on. Concurrent search, which is effective on reducing latency, can be implemented most easily for iterative lookup. As for security, recursive lookup (fast) and iterative lookup allow the nodes responsible for particular IDs to be known publicly, and may assist attacks to such nodes. However, it is difficult for malicious nodes to control routes over recursive lookup. Transparency of routing and hiding responsible nodes are contradicting requirements.

Application developers need to be aware of those contradicting properties of lookup styles, and choose an appropriate style for the requirements of the application. We hope that this research will become one of the guidelines for such designs, and be helpful for software developments with respect to structured overlays.

Acknowledgment

This research was conducted in cooperation with Mr. Daishi Kato, NEC Corporation.

第 6 章 On Scalability of DHT-DNS Hybrid Naming System

In this chapter, we describe a DHT-DNS hybrid naming system we have designed and its evaluation we have conducted on a large-scale emulation testbed.

We found a bottleneck at the mounter that limits scalability of the proposed system, which required an optimization.

6.1 Background

RFID tags and object traceability systems are among promising solutions in ubiquitous computing and networking.

In a traceability system, a naming system is essential to bind IDs and other information. Requirements for such a naming system include that it must have superior *scalability* to support lookup of billions of items in the world; which makes DHT suitable for the purpose. Another requirement is that it must be *legacy friendly*; consumers will not own dedicated clients for product traceability systems, and they expect to use web browsers and DNS.

Figure 6.1 shows the outline of the structure of EPCglobal GID-96, which is the commodity ID to be dealt with the DHT-DNS mounter.

- It consists of three parts: corporation, commodity and instance, in addition to the header.
- The top two parts are smaller and static. Which means that caching is effective for those fields.
- The *Serial Number* part corresponding to each instance is dynamically created, and caching is not effective. In most cases, it is read at most several times in one place.

Figure 6.2 shows the configuration of the name space to be used by the DHT-DNS mounter. Figure 6.3 shows the internal structure of the DHT-DNS mounter.

- All DHT nodes has authority on *Serial*

Header	General Manager	Object Class	Serial Number
8 bits	28 bits	24 bits	36 bits

Fig. 6.1. Structure of GID-96

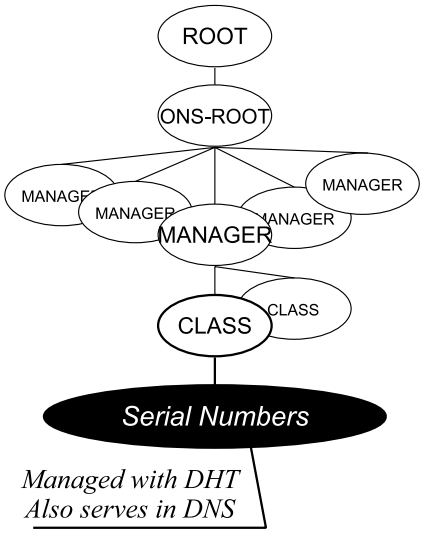


Fig. 6.2. Configuration of the name space

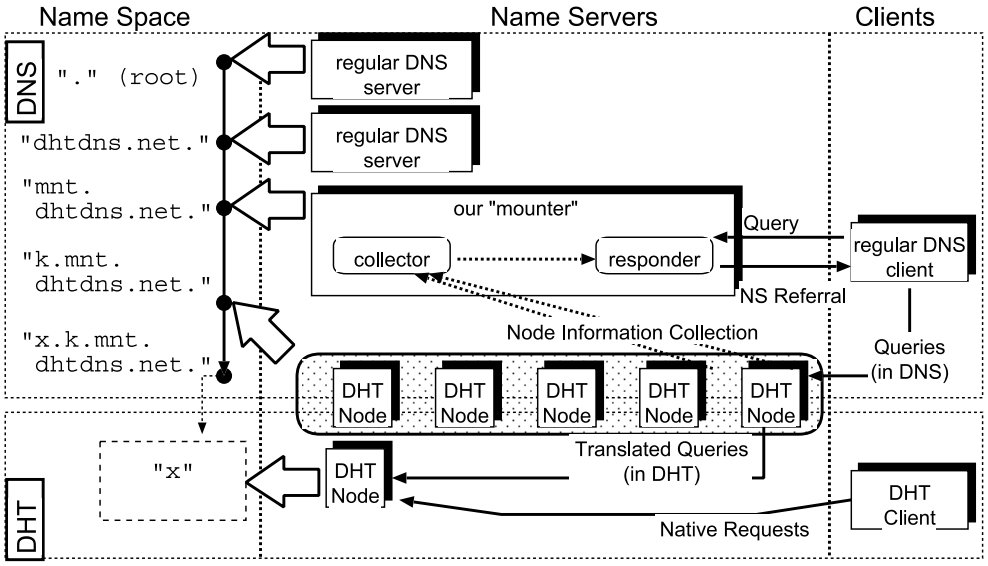


Fig. 6.3. Internal structure of the DHT-DNS mounter

Number zone.

- The zone corresponding to *Object Class* works as load balancer.
- The DNS queries to *Serial Number* zone is distributed over all DHT nodes.

6.2 Evaluation methodology

Figure 6.4 illustrates the environment used for evaluation. The environment consists of the four kinds of nodes:

- *Load clients* to impose the load. M denotes the number of load clients henceforth. Those clients generate a constant load, and therefore M can be a metric for the size of the load.
- *Root NS* is the single name server to root the DNS for this experiment. It corresponds to ONSRoot.
- *Mounter* is the device for compatibility.
- DHT nodes manage the meta information in a distributed manner. N denotes the number of DHT nodes henceforth, which can be a metric for the processing power.

Figure 6.5 shows the combinations of the parameters N and M. We have constructed the environment illustrated in Figure 6.4 for each combination in Figure 6.5 to analyze how queries are distributed. In the series of experiments, we analyzed the characteristics of DHT by a

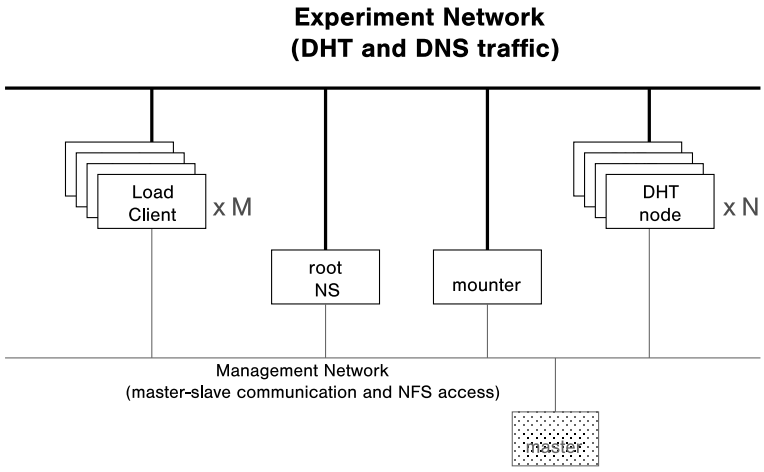


Fig. 6.4. Evaluation environment

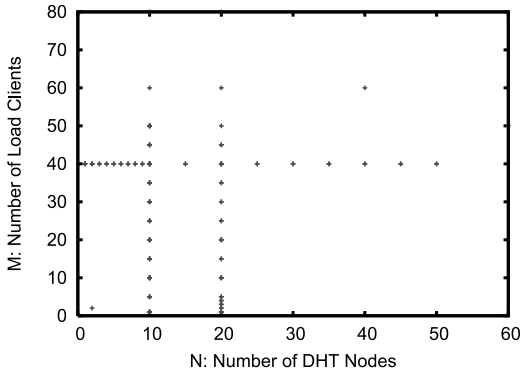


Fig. 6.5. Combinations of parameters N and M

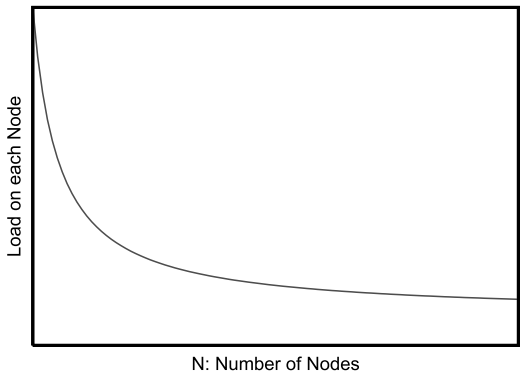


Fig. 6.6. Prediction of load distribution

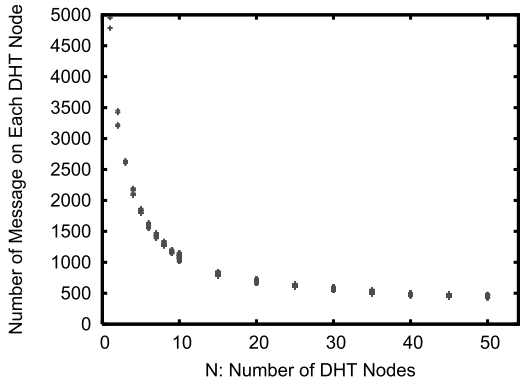


Fig. 6.7. Resulted load distribution

constant N and variable M, and the characteristics of the mounter by a constant M and variable N.

Figure 6.6 shows the expected distribution (the inverse of N) of the amount of queries for each DHT node with a constant M and variable N. Figure 6.7 shows the distribution of actual observed amount of message arrivals at each node. The expected and actual values show similar tendencies, and the load seems

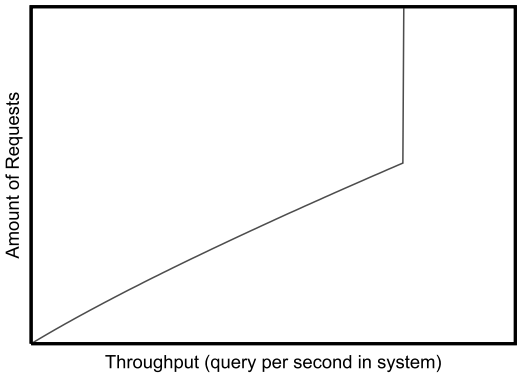


Fig. 6.8. Expected tendency of query process performance

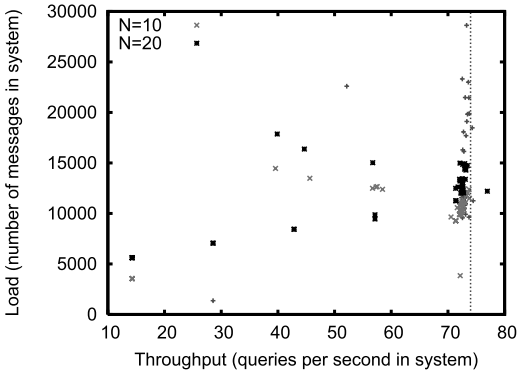


Fig. 6.9. Resulted queries and process performance

well-balanced, which suggests that load-balancing is functioning well.

Figure 6.8 shows the expected tendencies for the performance (queries per second) with a constant N and variable M. We set the horizontal axis to the performance and vertical axis to the traffic. When the amount of received messages grows gradually, it scales well to a certain degree because of the distribution characteristics of DHT, and then shows constant performance because of a bottleneck.

Figure 6.9 shows the actual observed performance, which does not contradict our expectation.

However, we discovered that our implementation is too slow for achieving sufficient load-balancing for the large-scale DHT network.

6.3 Performance optimization and its effects

We have measured the performance of the

mounter in a separate experiment, and obtained 40 qps (queries per second). This performance is insufficient, and we have optimized the implementation. By compiling the code with JIT compiler, we had about 420 qps (10 times as high). Then we rewrote the code with language C, which resulted in 1800 qps in the same settings. This result suggests that the system is capable of 3200 queries in the conditions same as the evaluation environment.

Acknowledgment

Part of this research is an outcome of the analysis of the results we obtained from using the large-scale testbed (StarBED) at Hokuriku Research Center founded and managed by NICT, in cooperation with Prof. Shinoda, Mr. Chinen and Mr. Miyachi of JAIST. We would like to express our thankfulness to them.

第 7 章　Conclusions

In this report, the following topics we have investigated this fiscal year have been described in detail:

- Local production, local consumption P2P architecture
- P2P economics
- Large-scale distributed measurement
- Studies on routing algorithms for structured overlays
- DHT-DNS hybrid naming system

Since all our development projects are in *deploy* phase now, we will continue our efforts for refining our methodologies through actual usage of our technologies in our lives.