Ying Zhang

On the Correlation of Geographic and Network Proximity

-Evaluation of Regional Delay Distributions from Real-World Internet Data

Master thesis based on the examination and study regulations for the
Master of Engineering degree programme
Information Engineering
at the Department of Information and Electrical Engineering
of the Faculty of Engineering and Computer Science
of the University of Applied Sciences Hamburg

Supervising examiner: Prof. Dr. Schmidt
Second examiner: Prof. Dr. rer. nat. Renz

Day of delivery November 9th 2006

Ying Zhang

Title of the Master Thesis

On the Correlation of Geographic and Network Proximity
-Evaluation of Regional Delay Distributions from Real-World Internet Data

Keywords

Internet edge scan, location mapping, hop distance, round-trip-time, traceroute, IP source routing

Abstract

This work analyzes and evaluates edge distance distributions in various regions. We use a commercial geolocation database to cluster IP ranges based on geographic location such as a city. We use *traceroute* program to collect packet forwarding path and round-trip-time of each intermediate node. The transit node is determined for each pair of destination hosts and an upper bound of the node distance is estimated. The last effort of this work is to validate the results which are obtained from a single origin. Source of *traceroute* probes are varied. The results from different locations are compared to obtain the best estimation of network distance of each pair.

Ying Zhang

Thema der Masterarbeit

Zur Korrelation von geographischer Nähe und Netzwerkdistanz
- Evaluierung regionaler Verzögerungsverteilungen auf der Basis realer Internet Daten

Stichworte

Internet Rand Vermessung, Ortsauflösung, Hop Distanz, Laufzeitmessung, *traceroute*, IP Source Routing

Kurzzusammenfassung

Diese Arbeit analysiert und evaluiert die Verteilungen der Netzwerkdistanzen in verschiedenen Regionen. Wir benutzen eine kommerzielle Ortsdatenbank für die Ortung und räumliche Bündelung von IP Adressbereichen auf einer Skala von z.B. Städten. Wir benutzen das *Traceroute*-Werkzeug, um Routingpfade und Laufzeiten für jeden Vermittlungsknoten zu ermitteln. Für ein jeweiliges Paar von Zielknoten ermitteln wir den letzten gemeinsamen Transitrouter und hierdurch eine obere Schranke für Knotendistanz im Netzwerk. Schließlich versucht die Arbeit, die von einer einzelnen Quelle aus ermittelten Daten zu validieren. Hierzu werden die Quellen der *Traceroute*-Verfolgung variiert und in der gemeinsamen Betrachtung versucht, die besten Schätzungen für die paarweisen Netzwerkdistanzen zu gewinnen.

Contents

1. Introduction

1.1 Motivation

The upgrading from IPv4 to the next generation Internet is under way. Mobility support of IP layer is greatly improved in IPv6 networks. Johnson et al. [12] proposed seamless mobility support in IPv6 by designing a new protocol mobile IPv6 (MIPv6). It enables a mobile node to maintain its connectivity to Internet when a handover is performed. Handovers produce packet loss, delay and jitter while real-time applications such as VoIP (Voice over IP) and VCoIP (Video Conferencing over IP) require restrictively high quality of handover performance.

This work is initially motivated by handover performance in mobile networks. Mobile node has a home address, which is permanent, and a unicast routable remote address. The mobile node is assigned the care-of address when it is visiting a foreign network. The mobile node sends a Binding Updates to its home agent (A router on a mobile node's home link with which the mobile node has registered its current care-of address.) and correspondent node (communication partner) in order to associate the home address with the care-of address. Suppose, in a mobile network, a mobile node is away from home and remains connected to the Internet. The user of the mobile node walks along the road. The movement will eventually cause the mobile node leaving the previous foreign network and probably entering a new foreign network, i.e. service provider is changed. The mobile node needs to acquire a new care-of address in the new foreign network. After obtaining a new care-of address, the mobile node needs to send Binding Updates to its home agent and correspondent node. In this way, a handover is performed.

Due to Binding Updates procedures, handover performance of mobile IPv6 is strongly topology dependent. The handover process is composed of geometry independent local handoff, the Layer2 link switching, the IP readdressing and the geometry dependent Binding Updates. Handover time can be represented by the following equation [8]:

$$t_{handoff} = t_{L2} + t_{local-IP} + t_{BU}$$

Where t_{L2} denotes the Layer 2 handoff duration,
$t_{local\text{-}IP}$ the time for local IP reconfiguration,
and t_{BU} the Binding Update time.

The handover latency resulting from standard Mobile IPv6 procedures is often unacceptable to real-time applications. Koodli [13] has proposed for fast handovers in IPv6,

which improves handover latency. A handover is initiated by previous access router, which sends Handover Initiate message to new access router and the latter replies with Handover Acknowledge message. The mobile node sends a Fast Bind Update to previous access router after configuring a new care-of address on the new link. The previous access router replies with Fast Binding Acknowledgement. And then, the mobile node sends a Fast Neighbor Advertisement to announce itself to the new access router. Arriving or buffered packets can be forwarded to the mobile node immediately. The newly arriving packets will be forwarded to the mobile node by new access router and the handover is done. Schmidt and Wählisch [8] calculated the performance decreases originating from different handover schemes. As they found out, either reactive (HMIPv6) or predictive handover (FMIPv6) is performed, the additional arrival delay time during handover depends on the transition time of a packet from one access router to the other access router and the transition time from mobile node to both access routers. The distances from mobile node to both access routers are assumed to be small, since they most likely are connected through a local wireless link. Therefore, predictive and reactive handovers mainly depend on the network distance (e.g. hop distance, round-trip-time) of access routers.

In the real-world, when a mobile node switches from one access router to another access router, the geographic distance between these two access routers are close. However, performance of handover depends on complexity of network topology between two access routers. For example, how many hop counts are between two access routers and how much is the round-trip-time? Figure 1-1 shows a simple analytical mode of handover in mobile networks. A mobile node moves from access router 1 to access router 2. The step size of geographic distance is considered to be small. The network distance l_1 and l_2 to home agent and correspondent node must be regarded as significantly large. The network distance between access routers l_3 is a parameter to characterize handover performance. It represents somewhat the step size of network distance of handover.

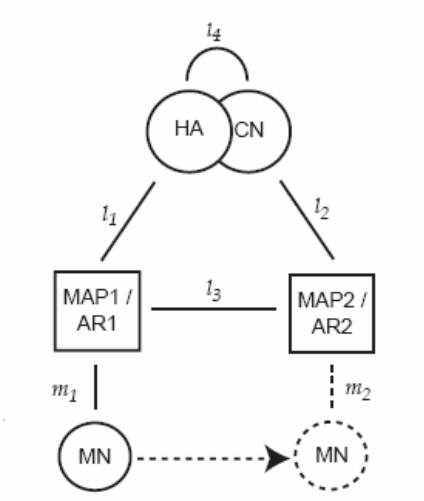

Figure 1-1 a simple analytic model of handover[①]

[①] This figure is taken from [8] on page 129.

Since network distance of access routers is a critical part of handover and since handovers in real-world mobility scenario can only be performed geographically neighboring networks, we explore the network distance of access routers within geographic vicinity from real-world Internet data. There are two basic metrics to measure network distance: hop count and round-trip-time. When a handover is performed, the probability of access routers being N hops apart (where N is a natural number) affects the performance of handover. The hop distances of intra-network access routers obviously range between zero and a few hops, but edge distance distributions at inter-network traversals are rather unpredictable. We collect real-world inter-network data to evaluate edge distance distributions in different regions such as Europe, Asia and USA. However, hop count cannot solely represent routing complexity because it depends also on other metrics such as bandwidth, reliability and cost. In fact, handover performance is mainly affected by temporal aspects. Hop count is not an immediate temporal metric and the more realistic temporal measure is the round-trip-time.

1.2 Related work

Since IP addresses are location-independent, there has been much work on the problem of mapping IP addresses to geographic locations. One of the latest studies was done by Padamanahan and Subramanian [2]. They studied geographic properties of the Internet and IP to location mapping techniques. They have developed three techniques for location mapping. First technique, *GeoTrack*, tries to infer location based on the DNS names of the target host or other nearby network nodes. The second technique, *GeoPing*, uses network delay measurements made from geographically distributed locations to infer the coordinates of the target hosts. The third technique, *GeoCluster*, combines partial IP-to-location mapping information with BGP prefix information to infer the location of the host of interest.

Ronda and Bila [3] simplified the geolocation problem through indirection. They proposed finding the geolocation of a nearby host, called a landmark, instead of the target address. They have implemented a geolocator tool, *Appono*, which takes the target IP address as argument and returns the city name, country, latitude and longitude of the IP address.

There are also efforts outside of the research community. Many companies have developed their proprietary location mapping techniques and sell their commercial geolocation databases.

Internet mapping has been studies for over ten years. These studies focus on characterizing and delineating Internet topology and performance. CAIDA [15] is pioneer, who record and measure Internet data continuously more than ten years. They collect data on Internet nodes and links to create a graph-like map of parts of the Internet. The recorded data can be used to analyze the performance of Internet nodes.

Cheswick, Burch and Branigan [6] have been collecting and recording routing paths from a test host to each of over 90,000 registered networks on the Internet since August, 1998. They used *traceroute*-style packets only from their test host to map outgoing paths. Of course, this project has its limitations since many Internet routes are asymmetric and the incoming path is often different from the outgoing path.

Mercator project [7] uses hop-limited probes in the same manner as traceroute, to infer an Internet map at the route-level. It uses "informed random address probing" to explore the addressable IP address space when determining router adjacencies, uses source-route capable routers whenever possible to discover cross-links. After running for three weeks in the summer of 1999, Mercator had discovered nearly 150,000 interfaces and nearly 200,000 links.

Another mapping project taken by CAIDA [15] has done a study of network connectivity in the Asia-Pacific region. It mainly focused on network latency and performance, autonomous system, country peering and third party transit. They use *skitter*, which uses a methodology similar to that of traceroute with ICMP echo request packets rather than UDP packets, to measure the forward IP path and round-trip-time to about 2,000 destinations in the Asia-Pacific region. The collected data suggested that minimum round-trip delays are correlated with physical distance between hosts. From their measurements, they also concluded that U.S. networks provide transit for 71.4% of the total *skitter* path that neither originate nor end in the U.S.

Paxson [14] analyzed 40,000 end-to-end route measurements using repeated *traceroutes* among 37 Internet sites in November-December of 1994 and 1995. He examined routing behavior including routing pathologies, stability and path symmetry. He found that Internet paths tended to be heavily dominated by a single prevalent route, and about two thirds of the source-destination pairs used path that persisted for days or weeks.

1.3 Overview of the project

Performance of handover depends on routing complexity between two access routers. From the point of view of geography, neighboring access routers are located in geographic vicinity. However, from perspective of network, the routing complexity between two neighboring access routers is quite unpredictable. For example, the packets are routed across autonomous systems (AS) or Internet Service Providers (ISP). In this work, we explore edge distance distributions of geographic proximity. We scan and collect real-world Internet data from geographically separated cities. We record forwarding path and round-trip-time of each end host and evaluate regional delay distributions from these data. Especially, we analyze distributions of hop distance and round-trip-time in various regions.

Although the motivation comes from IPv6 networks, topologies are in general similar for IPv4 and IPv6 and they may deviate in detail. Nowadays, IPv6 networks are not widely

deployed and no database offers IP to geolocation mapping based on IPv6. In addition, IPv6 routing is not supported by the test location and we cannot perform IPv6 traceroute from it. Therefore, instead of measuring IPv6 networks, we measure IPv4 networks.

This work can be divided into three parts:

- Cluster IP ranges according to geographic location

Since we want to find out regional delay distributions within geographic vicinity such as a city, we initially identify clusters of IP ranges. But IP addresses contain no geographic information and we need additional aids to cluster IP addresses. As mentioned in the previous section, there are many IP to location mapping techniques, but they are host-based method and they don't have full set of IPv4 allocation. However, commercial geolocation databases offer almost the full set of allocated IPv4 addresses and their corresponding geographic locations. In order to cluster IP ranges based on geographic locations, the best way is to use a geolocation database and do a filtering on it. For example, if we want to get a full set of IP addresses assigned for Hamburg, we can use a clustering strategy to filter out all IP ranges which belong to Hamburg.

- Internet edge scan from a single source

The most common way to acquire routing path is to use a traceroute-style tool. We use *traceroute* program [23] to map outgoing paths to various destinations. We collect packet forwarding path and round-trip-time of each intermediate node along the path. All the destination hosts are chosen from the same IP cluster, namely, they are located in the same city.

For each pair of end hosts, the interesting part is the divergence of both routing paths and the conjunct part is stripped off since it is irrelevant to pairwise hop distance. The coinciding hop closest to the destinations is identified as transit node, interconnecting two edge nodes. Under the assumption of symmetric routing at Internet border areas, an upper bound for edge node distances can be derived from a single source.

The experiment is based on four cities (Hamburg, Berlin, San Francisco and Shanghai) which are geographically disparate and we can analyze behaviors in different continents. We use subset of their IP cluster (e.g. 100, 200, 500 or 1000) to study the distribution of edge distance and round-trip delay in a geographic proximity.

- Validate and countercheck results

If the probes are sent only from one single source, the results might not be the optimal ones. The resulting network topology is tree-like map and only limited connectivity information can be obtained. In addition, the assumption of symmetric routing might cause inaccuracies because Internet routes are often asymmetric. Therefore, the last

effort of this work is to validate the results which are obtained from a single host. We perform traceroute from other locations which is a service offered by many websites. Varying source of traceroute can offer a better insight of network topology. If there are enough appropriate source locations of traceroute, the neighboring connectivity of a router can be determined. Thus, the optimal edge distance can be approximated.

The other method to validate results is to use IP source routing. If source routing is activated, sender can specify the hops that a packet must travel through. Traceroute can be performed to one destination via the access router of the other destination. If a route exists between them, this route will be a better approximation of optimal edge distance.

1.4 Organization of the report

This report is organized as follows. Chapter 2 introduces the methodology and tools which are used in this project. In Chapter 3, we will present the solution and concept to solve the problem. We will present the results in chapter 4. In chapter 5, we will analyze and evaluate our data and results. Finally, we give the conclusion in chapter 6.

2. Methodology and tools

2.1 Hop distance and RTT

There are two basic metrics to measure routing distance: hop distance and round-trip-time (RTT). One router along the path from source to destination is called a hop. Hop distance counts how many hops are along the path. RTT, as its name implies, is the time the reply is received minus the time the request was sent. This relationship is shown in Figure 2-1.

Under certain circumstances, RTT infers geographic distance. For example, Padmanabhan and Subramanian [2] suggest that 90% of nodes within 5 ms RTT are located within a radius of 50 km and 90% of nodes within 10 ms RTT are located within a radius of 300 km. Hop distance can also give hints on geographic distance. A destination which is five hops away, is much more likely (but not necessarily) to be closer than any destination at a distance at 25 hops. But hop distance and RTT might be erroneous in some situations such as network congestion or circuitous routing, which causes the packets taking another routing path or delay time being unreasonably large. In addition hops in terms of RTT may be far apart from each other, especially when routing overlay techniques such as Multiprotocol Label Switching (MPLS) are used.

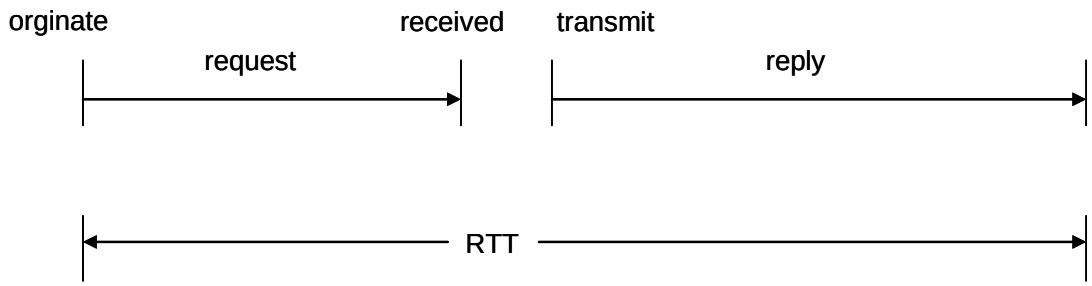

Figure 2-1 Round-trip-time[②]

In this work, these two metrics are chosen as basic metrics for the measurements. They are recorded for each pair of end-hosts. In addition, the forwarding path and RTT of each intermediate router are collected for future use.

[②] This diagram is taken from TCP/IP Illustrated, Volume 1, The protocols, page 75. [1]

2.2 *Traceroute* utility

The *traceroute* program, written by Van Jacobson, has been one of the most popular tools for acquiring a network topology so far. *Traceroute* infers an IP route between a source and a destination. Although there are no guarantees that two consecutive IP datagrams from the same source to the same destination follow the same route, most of the time they do.

The *traceroute* program is available on most computers which support networking. For example, command *tracert* is used on Microsoft Windows while Unix-style operating systems use command *traceroute*.

2.2.1 How *traceroute* works

Traceroute uses UDP datagram and the TTL (time-to-live) field in the IPv4 header. The TTL field is an 8-bit field that the sender initializes to certain value. The purpose of TTL field is to prevent datagram from ending up in infinite loop. When a router gets an IP datagram whose TTL is either 0 or 1, it must not forward the datagram. Instead the router throws away the datagram and sends back to the originating host an ICMP "time exceeded" message.

Traceroute sends an IP datagram with a TTL of 1 to the destination host. The first router to handle the datagram decrements the TTL, discards the datagram, and sends back the ICMP time exceeded. This identifies the first router in the path. *Traceroute* then sends a datagram with TTL of 2, and we find the IP address of the second router. This continues until the datagram reaches the destination host.

Traceroute sends UDP datagrams to the destination host. The destination UDP port starts usually at 33434 and is incremented by one each time a datagram is sent. *Traceroute* choose a large UDP port number to make it improbable that an application at the destination is using this port. This causes the destination host's UDP module to generate an ICMP "port unreachable" error when the datagram arrives. All *traceroute* needs to do this differentiate between the received ICMP messages (time exceeded versus port unreachable) to know when it reaches the destination.

2.2.2 Functionality of *traceroute*

The operating system of the test-bed machine is SuSe Linux 9.2. The integrated *traceroute* program on this version of Linux does not support *traceroute* with ICMP. Therefore, an additional *traceroute* program, which was written by Lawrence Berkeley National Laboratory [23], is installed on the machine. The version of *traceroute* used in this work is 1.4a12.

Usage and possible options of *traceroute*[3] is shown in Figure 2-2.

```
Usage: traceroute [-dFInrvx] [-g gateway] [-i iface] [-f first_ttl]
        [-m max_ttl] [-p port] [-q nqueries] [-s src_addr] [-t tos]
        [-w waittime] [-z pausemsecs] host [packetlen]
```

Figure 2-2 synopsis of *traceroute*

- Queries

The *traceroute* program can specify number of times to send a query for a given hop. By default, most *traceroute* programs send three queries. In this work, it is set to one query due to two reasons: first, this can save time. Three queries means that the time spends on *traceroute* is at most three times longer than one query. Second, there are some strange outputs of traceroute information of Shanghai. This phenomenon will be discussed in more detail in section 3.6.3.

- Query timeout

Query timeout means how many seconds to wait for a response to each query sent. If unspecified, it defaults to 5 seconds. It is set to 2 seconds for the experiment. Although the waiting time is reduced to a value less than half of the original waiting time, it is still a conservative value. For example, the originating host is located in Hamburg. We can say that Shanghai might be an extremely faraway destination since packets originated in Europe sometimes travel to Asia via USA, which means packets may travel around the earth. However, the round-trip-time between originating machine in Hamburg and target host in Shanghai is still less than one second. Two experiments are made in order to compare the results from default value with modified value. The results are the same, that is, the packets take the same routing path no matter the query timeout value is 2 seconds or 5 seconds.

- Maximum TTL

Maximum TTL means maximum number of hops that the *traceroute* program will try before giving up. The default value 30 is not modified since it is large enough for almost all destinations.

- Use ICMP

If use ICMP option is activated, the *traceroute* program performs probes with ICMP echo request rather than UDP. By default, this experiment uses ICMP to perform *traceroute* first, in case of an unsuccessful *traceroute*, and then try UDP instead.

[3] The traceroute command on Windows systems is “tracert” and the format different from the format on Linux systems. In addition, Linux systems use UDP datagram by default while Windows systems use ICMP. Linux is chosen as test-bed operating system; therefore, the discussions are based on Linux systems.

- Source routing

The *traceroute* program can use loose source routing as well. However, source routing can be used by malicious attack so source routed packets are either ignored or discarded by most routers (See section 2.5). Therefore, this feature of *traceroute* is not feasible in this experiment, although it is a quite good way to estimate lower bound of edge distance.

Figure 2-3 shows an example of *traceroute* output.

```
traceroute to 62.206.221.211 (62.206.221.211), 30 hops max, 52 byte packets
 1  141.22.64.1 (141.22.64.1)  1.051 ms  0.248 ms  0.222 ms
 2  141.22.4.121 (141.22.4.121)  0.737 ms  0.677 ms  0.667 ms
 3  xr-ham1-ge9-4.x-win.dfn.de (188.1.47.57)  3.846 ms  4.179 ms  3.474 ms
 4 ar-bremen1-te2-1.x-win.dfn.de (188.1.144.90) 21.845 ms 21.245 ms 21.852
ms
 5  xr-han1-te2-1.x-win.dfn.de (188.1.144.85)  10.228 ms  7.262 ms  8.185 ms
 6  xr-fra1-te3-4.x-win.dfn.de (188.1.145.126)  21.670 ms  21.166 ms  21.862
ms
 7  nap.de-cix.de.bmcag.net (80.81.192.208)  21.794 ms  21.328 ms  21.752 ms
 8  ge-0-4-0-0000.bnx-ham01.bmcag.net (194.140.111.58)  33.711 ms  34.010 ms
33.393 ms
 9  d463d7c1.datahighways.de (212.99.215.193)  58.032 ms  55.810 ms  56.135
ms
10  62.206.221.211 (62.206.221.211)  57.725 ms  57.478 ms  57.709 ms
```

Figure 2-3 *traceroute* output

The first line of output gives the name and IP address of the destination. Each numbered line begins with TTL, followed by the name (if reverse DNS lookup is successful) of the host or router and its IP address within brackets. For each TTL value, three datagrams are sent. For each returned ICMP message, round-trip-time is calculated and printed. The round-trip-times are calculated by the *traceroute* program on the sending host. They are the total RTTs from the *traceroute* program to that router. If the per-hop time is needed, then one has to subtract the value printed from TTL N for the value printed for TTL N+1.

2.3 IP to Geolocation mapping

In the recent years, geography has more and more impact on Internet. However, IP addresses do not contain geographic information. There are some classic methods to locate IP addresses but the accuracy is unpredicted. Existing commercial geolocation databases

map ranges of IP addresses to geographic locations. Their accuracy also needs to be verified.

2.3.1 What is IP2Geo?

IP2Geo means mapping an IP address to a geographic location. However, IP addresses are location-independent and they contain no information of physical location. Hence, finding the geographic location of an IP address is a challenge.

2.3.2 Location mapping

There are several approaches to map an IP address to its physical location. They are:

- Use reverse DNS lookup.

  The name of a router might be helpful to find out its location. For example, the IP address of `188.1.144.90` is translated into '`ar-bremen1-te2-1.x-win.dfn.de`' (See Figure 2-3). From the DNS name of this router, we can conclude that this router is located in Bremen, Germany.

  The accuracy of this method depends on the configuration of DNS server. Since there is no standard naming conventions for DNS servers, it might be inconsistent among different ISPs. The previous example is easy to recognize because it contains full name of a city. However, many ISPs uses abbreviation of cities, e.g., Subramanian [2] has found out that there are at least 12 different codes associated with city of Chicago. Therefore, this approach is not so reliable; of course, we can obtain some hints from it.

- Use WHOIS [20] records.

  The most widely used one is to query WHOIS servers. It contains administrative contact information for all domains. But it is not highly reliable. For example, some organizations are spread across different geographic regions and WHOIS records are often at the level of an organization. In addition, the domain information is filled in during registration time. If the domain name registrators do not insist on keeping the database accurate and current, the database might be inaccurate.

- Use *traceroute*. (see section 2.2)

  Tool *traceroute* offers path information from source to destination. It lists routers which packets flow through and names of intermediate routers if the reverse DNS lookup is successful. These DNS names might be helpful to locate the host.

  In Figure 2-3, the traceroute output shows that a packet from source to destination travels through Hamburg, Bremen, Hanover, Frankfurt and then back to Hamburg. Although the DNS names of the last two routers do not contain any location information, one can infer that the destination is not far away from Hamburg.

Besides these classic methods, there are also efforts from research community. Padamanahan and Subramanian [2] have developed three techniques for mapping IP addresses to geographic locations. First technique, *GeoTrack*, tries to infer location based on the DNS names of the target host or other nearby network nodes. The second technique, *GeoPing*, uses network delay measurements made from geographically distributed locations to infer the coordinates of the target hosts. The third technique, *GeoCluster*, combines partial IP-to-location mapping information with BGP prefix information to infer the location of the host of interest. Ronda and Nilton [3] have implemented a command line geolocator tool, *Appono*. They propose finding the geolocation of a nearby host instead of the target itself. They have simplified location mapping problem by introducing a layer of indirection.

2.3.3 IP2Geo Databases

Nowadays, there are numerous services offering location mapping, such as MaxMind [16], NetGeo [17], IP2Location™ [21], IP2Country [18] etc. Some of them are commercial while others are publicly available. However, we can not obtain full set of IP2Geo databases. We collect some famous ones and choose most preferable one as our base IP2Geo database.

In order to differentiate various IP2Geo Databases, twenty worldwide IP addresses are randomly chosen. This is only a rough comparison by using free demos from these databases. We can not say which DB is the best and which one is the worst. The results also depend on the granularity of these free demos. Of course, from the point of view of statistics, twenty is really a small value to judge the quality of these databases. All of the following judgments are based on these free online demos.

NetGeo [17] is a geographic database from CAIDA [15]. It is a collection of Perl scripts used to map IP addresses and AS numbers to geographic locations. But this database has not been actively maintained for several years and this will not change in the foreseeable future. Therefore, the query results from this database are very inaccurate, especially for recently allocated or re-assigned IP addresses.

IP2Location™ [21] is one of the products of Hexasoft Development Sdn. Bhd. ("HDSB"). After more than two years of intensive research, data collection and development, they delivered the IP2Location™ locator services for the Internet community. This database is one of the most accurate one among these databases. But in some cases, it does not offer information of region and city.

Geobytes, Inc [19] also provides an IP address lookup service to assist users in locating the geographic location of an IP address. In general, this database is accurate. However, the size of this DB is not large enough; some of the testing IP addresses can not be located.

MaxMind [16] provides its geolocation technology through their product GeoIP®. This

database is also one of the most accurate databases except for small deviations. Therefore, this database is the most preferable one, since IP clustering discussed later is based on IP2Geo database. The accuracy of IP2Geo database influences directly the results of this experiment. In addition, it offers most features of an IP address, e.g., country, city, ISP, zip code, longitude/latitude, organization etc.

Detailed comparison is listed in Appendix A.

2.4 Network scanner

Scanning is important for network discovery and maintenance. For example, a network administrator can scan the internal network to check the status of LAN while a network researcher may scan a large block of Internet for studying characters of the Internet. With the increasing demands of scanning, various scanning tools have been coming out. However, these tools may scan different layers of OSI layer model. For example, some of them scan application layer while others scan IP layer or transport layer.

Although, there are numerous scanning tools nowadays, Nmap ("Network Mapper") [22] is still one of the most famous scanning tools. It is a free and open source utility for network exploration or security auditing. It can perform host scanning, port scanning (both TCP and UDP), OS detection and more. It has various scanning options to satisfy different requirements. In addition, Nmap is very portable: most operating systems are supported, including Linux, Microsoft Windows, FreeBSD, OpenBSD, Solaris, IRIX, Mac OS X, HP-UX, NetBSD, Sun OS, Amiga, and more.

Nmap offers diverse scanning options and one can combine these wide varieties of options to customize scan needs. Of course, what makes a host interesting depends greatly on the scan purposes. Network administrators may only be interested in hosts running a certain service, while security auditors may care about every single device with an IP address. An administrator may be comfortable using just an ICMP ping to locate hosts on his internal network, while a penetration tester may detect active hosts as well as port status of these hosts.

This work studies delay distributions of Internet edge distance (they would usually be hosts), therefore, scan is a great part of this work. Nmap is the most preferable network scanner for this work due to the following reasons:

- Free

  Nmap is available for free download [22]. Some other scanning tools are commercial and it costs time and money to order it. In addition, Nmap comes with full source code that one can modify it to meet specific requirements and redistribute it.

- Native with Linux

  This work is done thoroughly on Linux. Linux is far and away the most popular

platform for running Nmap. In one user survey, 86% said that Linux was at least one of the platforms on which they run Nmap.

- Powerful

Since we collect data from real-world Internet, the amount of scan work is very heavy. Nmap has been used to scan huge networks of hundreds of thousands of machines, so it can handle with this work. Basically, this work only need the scanning results of layer 3 (IP layer) and layer 4 (transport layer) of OSI layer model, but Nmap offers scanning of application layer such as service detection as well.

2.5 IP source routing

The term routing refers to selecting paths in a network along which to send data. Normally IP routing is dynamic with each router making a decision about which next hop router to send the datagram to. Applications have no control of this and are normally not concerned with it.

Although the routing decision is automatically made by routers, in the Internet Protocol (IP), there is an option field of IPv4 header which offers manual selection of routing path the packet should take through the network. This is called “source routing”. Source routing can be activated in option field of IP header and it enables the sender to specify the route. There are two forms of source routing:

- Strict source routing

In strict source routing, the sender specifies the exact route the packet must take. This is virtually never used.

- Loose source routing

The more common form is loose source record route (LSRR), in which the sender gives one or more hops that the packet must go through. For example, as depicted in, Figure 2-4, Host1 can specify that all the packets destined to Host2 must travel via Router B. Then, all the packets originated from Host1 are diverted to Router B.

Figure 2-4 shows how IP source routing works. Host1 wants to send a datagram to Host2, specifying a source route of Router B. In the option field of IPv4 header, first entry will be filled with IP address of Router B. Host1 takes the source route list, removes the first entry (Router B), moves all the remaining entries left by one entry and places the original destination address as the final entry in the list. The destination address of the datagram is now Router B. Router A checks the destination address of the datagram and finds it is not its own address and forwards the datagram as normal to Router B. Router B finds its own IP address in the destination address of the datagram and the next address in the list (Host2) becomes the destination address of the datagram. The IP address of outgoing interface of Router B replaces the source address (Host1). Router C checks the destination address of

the datagram and does not find a match. Then it forwards the datagram as normal to Host2, which is the final destination.

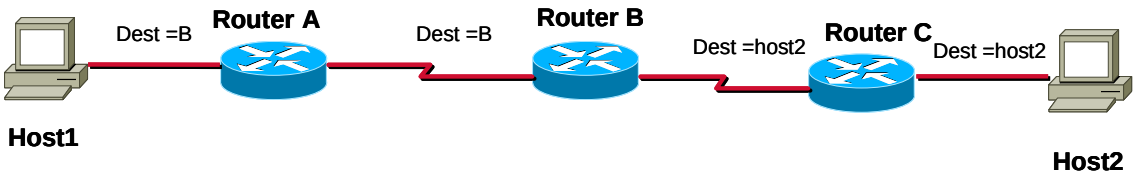

Figure 2-4 IP source routing

Source routing can be helpful to map a network. The *traceroute* program offers one option that specifies source routing (See section 2.2.2). IP source routing can be used to determine routing path between two end-systems, if it really exists between them. To consider the scenario in Figure 2-5, there exists one routing path between the seventh hop of host1 and the eighth hop of host2. If source routing is used in the *traceroute* program, we can specify the last hop of one host as an intermediate router of the other host. For example, *traceroute* is performed to host2 via the seventh hop of host1. The datagram travels to the seventh hop of host1 and then to the eighth hop of host2. Without source routing, the link between the seventh hop of host1 and the eighth hop of host2 is invisible to the source. Source routing offers a better option to estimate edge distance.

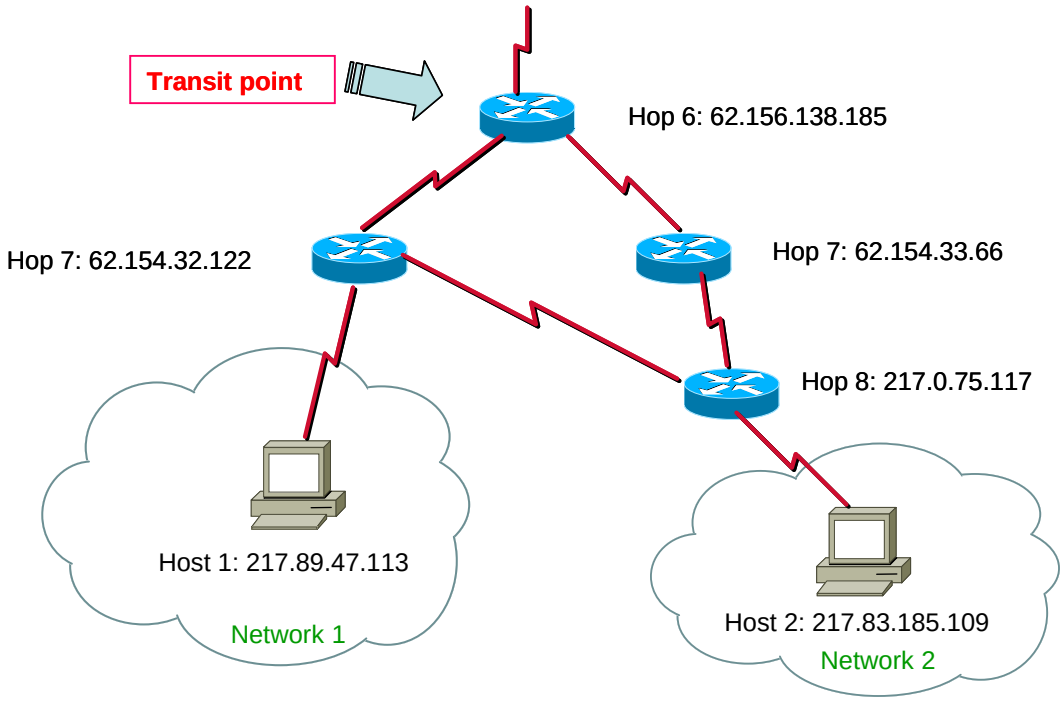

Figure 2-5 scenario of using source routing

However, source routing can be used for hacking purposes. Some machines are on the

Internet, but they may be using a private IP address such as 192.168.1.100. They are not reachable from the Internet directly. For example, to consider the scenario in Figure 2-4, Host1 is located in a local area network (LAN) and Router A is the default gateway of Host1. For Host1, Router A is a trusted machine. To suppose on Router A, IP source routing is not disabled. If Host2 is an attacker who wants to obtain data on Host1, he will try to intercept packets on Router A and modify option field of IP header. And then, he sends source routed packets to Host1 via Router A. Host1 receives the packets and "thinks" that the packets are originated in Router A. In this way, the attacker accesses Host1.

In consideration of security issues, many routers either ignore IPv4 option fields or discard packets carrying IPv4 options. In the real Internet world, only a few of routers perform source routing. We plan to get some complementary counterchecks from source routing, but unfortunately source-routed packets are discarded by HAW Hamburg (campus of test location). So we discard this method.

2.6 One vs. multi-origins routing paths

If all *traceroute* data are sent from the same location, it is a tree-like map rather a complete network map. If *traceroute* is performed from varying sources, the neighboring connectivity of a certain router can be determined. For example, if source1 performs *traceroute* to host1 and host2, the transit node is determined as transit node1. While source2 performs *traceroute* to those two hosts as well, but the transit node is totally a different one. From viewpoint of source2, the hop distance between host1 and host2 is one hop more than that of source1. Therefore, results from one location are not optimal. In order to obtain better estimations of edge distance, varying source of *traceroute* is necessary.

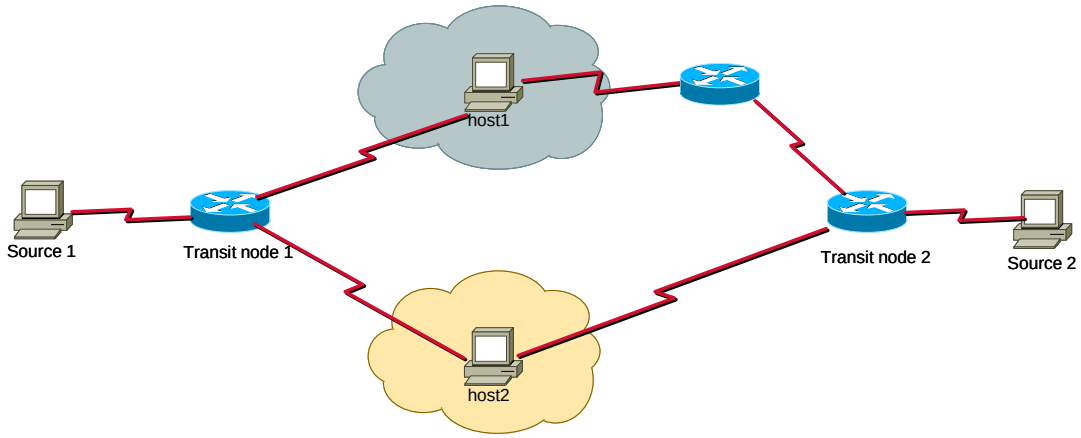

Figure 2-6 *traceroute* from various locations

If source routing is not applicable, it is necessary to compare the results from different locations. Therefore, the lowest number of hop distance between two hosts is the finest estimation and packets between these two hosts would probably be routed along that path.

[24] offers an exhaustive list of websites which people can perform *traceroute* from. Some of them are located in campus while others are offered by companies. In Germany, networks of universities are connected to the same backbone network. For example, the biggest peering point of Germany is located in Frankfurt, but if a packet originates in HAW Hamburg and destines to University of Hamburg, it will not be routed via Frankfurt. Since we want to analyze edge distance distributions at inter-network routing and the test machine is located in campus, choose a website offered by companies will be more meaningful.

3. Internet edge scan

3.1 IP clustering based on geographic locations

Commercial geolocation databases offer a light way to cluster IP ranges since they map ranges of IP addresses to geographic locations rather than single IP address. After the accuracy of these commercial geolocation databases are verified (comparison is listed in Appendix A), GeoIP from MaxMind is the most preferable one. Therefore, IP clustering is done based on this database.

3.1.1 GeoIP® from MaxMind

GeoIP®, as mentioned in previous section, is one of the commercial products of MaxMind. It provides geolocation data and services in real-time. When an IP address is given, GeoIP can determine which country, region, city, zip code, area code it belongs to. Furthermore, GeoIP can provide information such as longitude/latitude, connection speed, ISP, organization name, domain name and DMA code.

GeoIP is available in two formats: binary and Comma Separated Value (CSV).

- Binary format

  The binary format is a highly optimized database that supports fast lookups using open source API code. GeoIP has numerous open source APIs. Most popular programming languages are supported by GeoIP: C, Perl, PHP, Apache, Java, Python, C#, Ruby, VB.NET, and Pascal.

- CSV format

  The CSV file generally contains IP Address range and geographic data for all publicly assigned IPv4 addresses. Usually, file size of this format is large, therefore, MaxMind recommends using binary format with one of the APIs. The CSV format is recommended when the data is imported into a SQL database.

  The CSV format has two files: one file contains location ID, ISO 3166 Country Code, region name, city name, longitude/latitude, etc.; the other one contains location ID, start IP number and end IP number.

However, we need to cluster IP ranges according to geographic locations. This is only possible with the CSV format of the database. Each city has been assigned one or more location IDs. Different cities have different granularity, e.g., San Francisco has 63 location

IDs while Berlin has only one.

3.1.2 Clustering strategy

First, a manual input of country code and city name is required. There are cities having exact the same name. For example, Germany and USA both have a city called Hamburg. This can be identified by country code, which is unique. As long as location ID(s) of a city is determined, the ranges of IP numbers belong to the location ID(s) can be extracted from the database. Since we can not work directly with IP numbers, they are converted into IP addresses using the formula shown in Figure 3-1. Hence, we can get IP ranges of dotted separated form.

```
w = int ( ipnum / 16777216 ) % 256;
x = int ( ipnum / 65536    ) % 256;
y = int ( ipnum / 256      ) % 256;
z = int ( ipnum            ) % 256;

   Where % is the mod operator.
    ipnum stands for IP number.
 IP address is in the form w.x.y.z
```

Figure 3-1 translation algorithm

IP range separation was done by MaxMind. Usually, an IP range contains several to several thousand IP addresses. In some extreme cases, one IP range contains only one IP address. On the other hand, there exist IP ranges contains hundred of thousand of IP addresses. We assume that all the IP addresses in one IP range are in the same network, that is, they belong to the same company, the same ISP or a university.

The process of clustering is depicted in Figure 3-2 and can be summarized as:

1. extract location IDs of this city
2. filter out other location IDs and retain beginning IP number and ending IP number of desired location ID(s)
3. translate beginning IP number and end IP number to beginning IP address and ending IP address

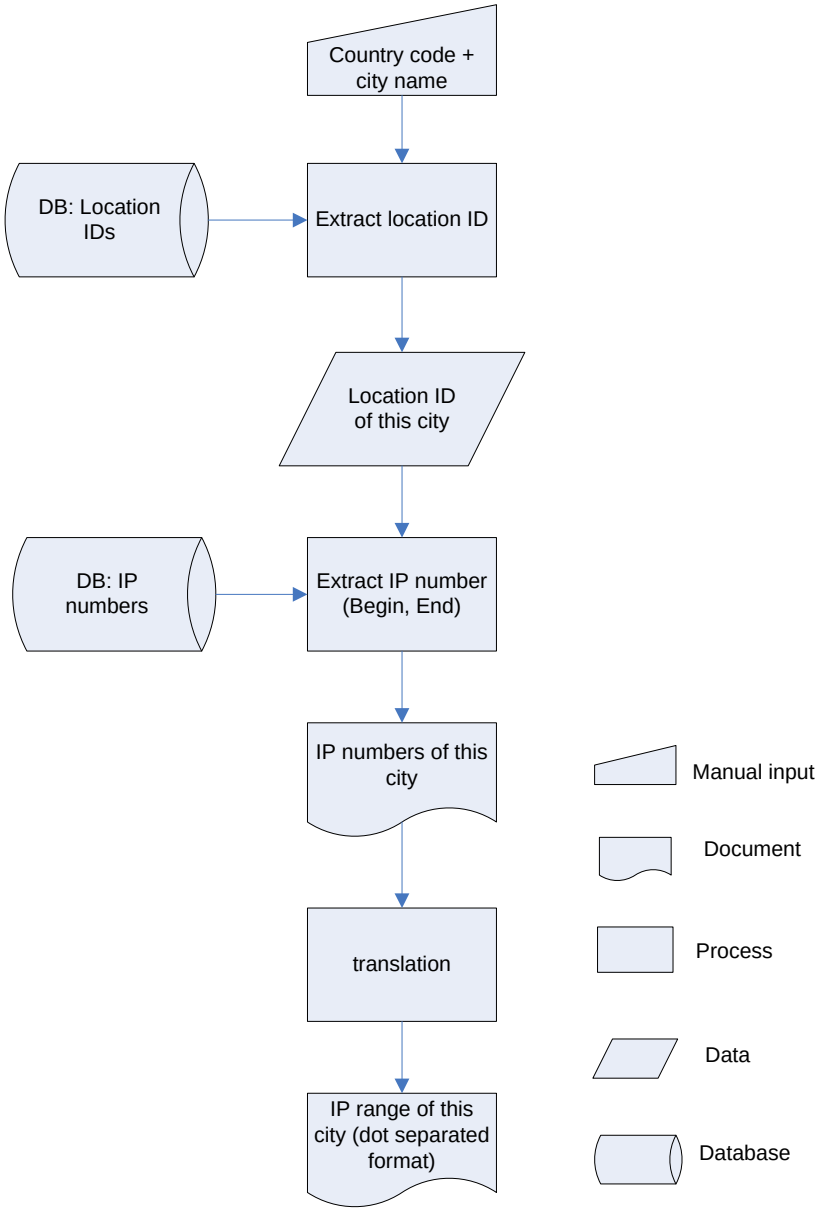

Figure 3-2 clustering IP ranges from IP2Geo database

3.2 Scan technique

Scan is one of the most important parts of the experiment, since active hosts should be discovered in each IP range and probably UDP port scan is required. As mentioned in section 2.4, scan is done by Nmap. Whether a suitable scan option or appropriate combinations of scan options are used will directly affect scan rate, correctness and accuracy of scanned results.

Scan procedure of this work can be split into two steps:

- Host discovery: locate an active host in an IP range.

- Port scanning: finds out an accessible port on the host found in previous step.

3.2.1 Host discovery

Host discovery is also known as ping scan, but it goes well beyond simple ICMP echo request packets. It offers arbitrary combinations of multi-port TCP SYN/ACK, UDP and ICMP probes. The goal of these probes is to solicit responses which demonstrate that an IP address is actually active (is being used by a host or network device).

The first step of the requirement is to reduce one IP range into a list of active hosts. By default, Nmap does host discovery and then performs a port scan against each host it determines as online. However, the first step of our scan is simple and no port scan is required because we need specific port scan which is performed in the second step. Nmap offers one scan option called “Ping Scan”, which tells Nmap to perform only ping scan and no further testing (e.g. port scan or OS detection) is performed. Ping Scan sends an ICMP echo request and a TCP packet to port 80 by default. It is more reliable than ping because many hosts do not reply to ping messages. When strict firewalls are placed in between the source host running Nmap and the target network, the probes send by Nmap may be dropped by those firewalls. As a result, hosts on that network could be identified as inactive.

One of the first acceleration of host discovery is to use option called “no DNS resolution”. It tells Nmap to never do reverse DNS resolution on the active IP addresses it finds. Since DNS can be slow even with Nmap’s built-in parallel stub resolver, this option can slash scanning time. In addition, DNS name is of no use in this experiment because most of active IP addresses found belong to the Internet edges (hosts) which usually do not have a DNS name.

However, some IP ranges are extremely large, e.g., a class B network contains 65536 hosts. The second acceleration method is to split IP range into smaller sub networks. Since our requirement is to find out an active host which can be traced to, the number of active hosts in a certain IP range is of no important use. Hence, we can split IP range to reduce scanning time. We limit subnet mask to 24. If the subnet mask is less than 24, then we cluster large IP range to smaller IP ranges with subnet mask greater or equal to 24. If the subnet mask of one IP range is greater or equal to 24, scan is performed directly.

By default, the *traceroute* program sends UDP datagrams with a destination port number starting at 33434. However, the *traceroute* program has also an option to specify use of ICMP echo request instead of UDP datagram. As long as a host is determined as online by Nmap, this means that the host responses to ICMP echo request. Therefore, we can use ICMP echo request to perform *traceroute* to the host. Ping is also a type of ICMP packet; therefore, if a host can be pinged, we can use ICMP to perform *traceroute* to it. This also saves scanning time since no port scanning is needed which costs one port per second. Therefore, performing traceroute by use of ICMP echo request is more preferable.

The successful ratio of *traceroute* with ICMP echo request is very high. However, it might be not successful all the time. For example, consistent data arrive or maximum number of hops exceeds. If these problems occur, *traceroute* is repeated for five times. If the problems still exist, we will use UDP probes to perform *traceroute*, which is also the content of next section.

3.2.2 Port scanning

In the situation of an unsuccessful *traceroute* with ICMP echo request, we perform port scanning. The *traceroute* program uses UDP datagram rather than TCP.

Since the *traceroute* program uses UDP ports, UDP ports of target host are scanned. UDP port scan can be activated by option called “UDP scans” and can only be executed under privileged mode. This is because it sends and receives raw packets, which requires root access on UNIX systems or administrator account on Windows systems.

Goal of UDP port scan is to find out an accessible port on the target host. Many port scanners have traditionally divided all ports into open or closed states, while Nmap is much more granular. Nmap classifies ports as six states. The six port states are: open, closed, filtered, unfiltered, open|filtered and closed|filtered. Of course, these states are not intrinsic properties of the port itself, but describe how Nmap sees them. For example, Nmap may detect port 33434/UDP as open from the same network, while identifies the same port as filtered across the Internet. This is due to firewalls. If a firewall is placed between the source host running Nmap and destination, the packets which do not obey rules of the firewall, will be discarded.

A port is accessible when it is in the state of open, closed or unfiltered. When a port is open that means an application is actively listening on it to accept TCP connections or UDP packets. If a port is accessible but no application actually listens on it, the port is in the state closed. An unfiltered port can be either an open port or a closed port since Nmap is not able to determine. A port in these three states can be used as base port to perform the *traceroute* program.

If a port is in other states (filtered, open|filtered, closed|filtered), the probes of Nmap are filtered by firewalls, router rules or host-based firewall software. Hence, a port in these three states is inaccessible for the *traceroute* program and ignored.

As mentioned above, we need scan UDP ports of the target host, which can be activated by option “UDP scans”. UDP scan works by sending an empty UDP header to every targeted port. If an ICMP port unreachable error is returned, the port is closed. Other ICMP unreachable errors mark the port as filtered. Occasionally, a service will respond with a UDP packet, proving that it is open. But most popular services run over TCP. DNS, SNMP

and DHCP (ports 53, 161/162 and 67/68 respectively) are the most common UDP services deployed nowadays. If no response is received after retransmissions, the port is classified as open|filtered.

There is one option called “Versions scan” that can be used to help differentiate the truly open ports from the filtered one. Therefore, this option is added as well to help us obtain a better scan results.

However, UDP scanning is in general slower and more difficult than TCP. Open and filtered ports rarely send any response, leaving Nmap to time out and then conduct retransmissions just in case the probe or response was lost. Closed ports are often even worse. They usually send back an ICMP port unreachable error. But many hosts limit ICMP port unreachable messages by default. For example, Linux 2.4.20 kernel limits destination unreachable message to one per second.

Of course, scanning all the ports on the target host is unnecessary, since it is time-consuming and might be regarded as malicious attack. The UDP port scan is accelerated by trying the default port (33434) of *traceroute* first. If this port is inaccessible, then scan another port range to check whether all UDP ports are blocked on the target host. Five to ten ports are recommended since a Linux-style limit of one packet per second wastes too much time and effect.

3.3 Route discovery

The network topology can be acquired by using the *traceroute* program [23]. The *traceroute* program can determine the path between an initial host and any destinations in the Internet if they are not hidden on certain purpose. The *traceroute* program lists all the hops which the packet travels through from source node to destination node.

The part of great interest to this work is the edge distances between any two end-hosts. In order to calculate hop distance between two hosts, one assumption must be made that the forwarding and returning edge routes of all packets are the same. This assumption allows one to derive the route between any two hosts based on the routes from a third machine to each of the hosts. This assumption might be certainly not true all the time on the Internet, however, without this assumption, it would not be possible to map a network topology.

For each pair of end-hosts, the transient node would firstly be determined which actually is a router. Since the *traceroute* program is performed from the same source machine, the common parts of both routing paths to end-hosts are stripped off. The transient node is determined by comparing each hop along both paths from source to destination. The last common hop is identified as the transient node between two end-hosts. This comparison should be done from the hop closest to destination host. Once the transient node is determined, the hop distance between two end-hosts would be the sum of hop distance from

transient node to one end-host and from transient node to the other end-host. This method offers an upper bound of hop distance between any two hosts. Ideally, all traffic can be routed from one host via transient node to the other host. Of course, this may not be the most preferable method. Because source routing is not applied which has been discussed in section 2.5, the shortest routing path from host1 to host2 cannot be directly determined.

In Figure 3-3 shows one scenario how to determine transient node and calculate hop distance between two hosts. The source node on which the *traceroute* program is performed is of IP address 141.22.64.9 and IP addresses of two destination hosts are 62.8.134.1 and 62.8.134.1 respectively. The packet travels to host1 through eight intermediate hops while it takes nine in-between hops to reach host2. The first five hops of both routing paths are identical. The sixth hops are disparate of both paths. However, the seventh hops are again coinciding. Based on the principle of determining the transit node, the seventh hop rather than the fifth hop is the transit node of these two end-hosts. The previous part of the routing path (from source to the sixth hop) is stripped from the *traceroute* output.

Once the transit node is determined, the calculation of hop distance between two end-hosts will be rather easy. The hop distance between host1 and the transient node is 2.The hop distance between host2 and transient node is 3. Then, these two hosts are determined to be five hops away, which is a likely upper bound of hop distance. The round-trip-time which a packet travels from host1 to host2 can be determined in the same manner.

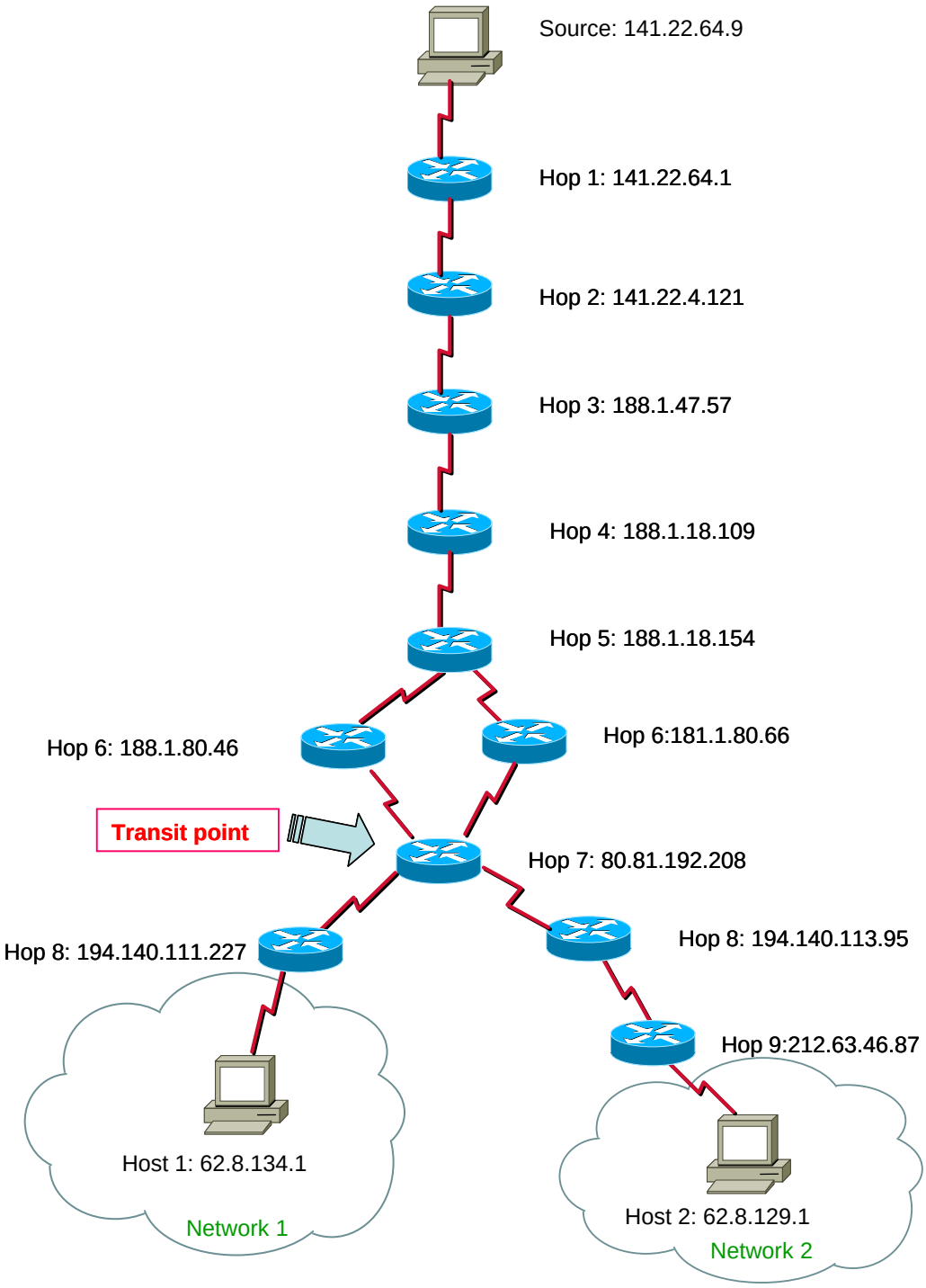

Figure 3-3 determination of transient node and hop distance

Of course, the network topology shown in Figure 3-3 is quite simple. However, even if the network topology is complicated the transit node can be determined and hop distance can be calculated in the same manner because of the assumption made before. As only seen from one *traceroute* origin, though, this distance need not be the optimal one.

3.4 Summary of procedure

The procedure of scan and edge distance calculation is depicted in Figure 3-4 and can be summarized as.

1. Read in one IP range from the input file
2. Reduce the IP range to a list of active hosts. If there is no active host, jump to step1
3. Take one IP address from the active host list
4. Perform *traceroute* to the IP address with ICMP
5. If traceroute is successful, record traceroute output and jump to step 1.
6. If jitter problem occurs, repeat step4 and step5. If maximal trail times exceed, then start UDP port scan.
7. If an accessible port is found on the active host, perform traceroute to it with UDP datagram.
8. If traceroute is successful, record traceroute output and jump to step 1.
9. If jitter problem occurs, repeat step7 and step8. If maximal trail times exceed, then jump to step 2 if there are still unexplored active hosts
10. If the current IP range is not the last one in the input file, jump to step1. Otherwise, start pairwise hop distance calculation and write log files.

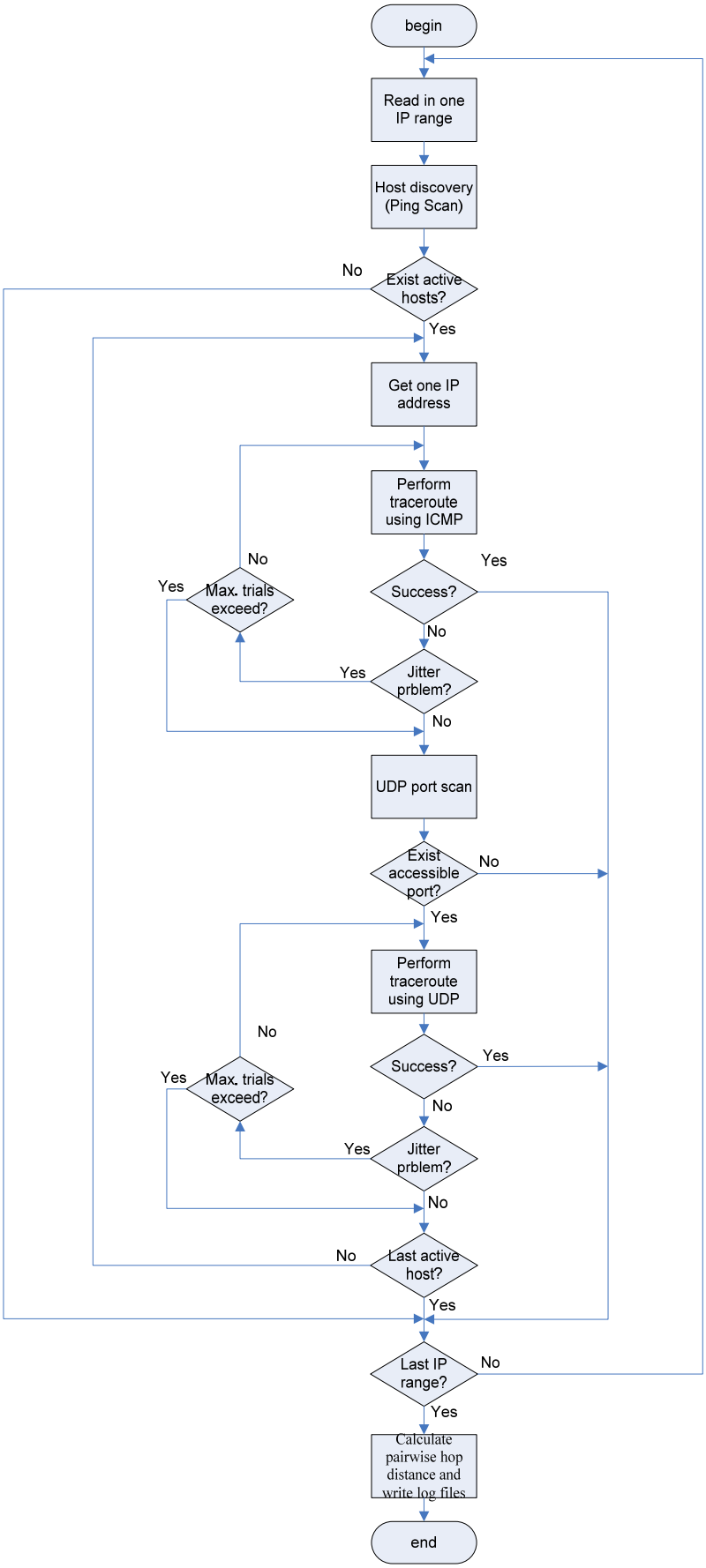

Figure 3-4 sequence diagram of scan procedure

3.5 *Traceroute* from other locations

The procedure of performing *traceroute* locally and performing traceroute from other locations are different. If *traceroute* is performed locally, the output of traceroute is handled by package of the programming language itself. If it is performed by means of a public *traceroute* server, the functionality is almost always only accessible via a Web site and thus, a request is first sent to the remote server via HTTP. Then, the server returns the result as HTML content, and the program needs to handle the *traceroute* output. The output of *traceroute* is embedded in a certain HTML tag. The first step is to strip off other information other than the *traceroute* output. Then, the *traceroute* output is parsed line by line to extract useful information such as IP address and round-trip-time of each intermediate router. The determination of the transit node and calculation of pairwise hop distance is the same as performing traceroute locally. (See section 3.3)

Of course, different servers use different way to perform traceroute such as UDP datagram or ICMP echo request packet. Not every host which appears to be up to the experimental host is also reachable from other locations. Almost all other servers send three queries to each intermediate router; therefore, we use the mean value of round-trip-time of three queries.

Comparison is done among the results from different servers and the experimental host. The lowest value of hop distance is the best estimation. Because we evaluate an upper bound in any of the measurements and the minimum of all upper bounds is still an upper bound, but supposedly a result closer to the actual value. It can also offer a better view of neighboring connectivity of the end-hosts.

3.6 Difficulties and possible inaccuracies

3.6.1 Impact of firewalls

The Internet consists of thousands of networks and millions of hosts. Due to the security reason, many Internet hosts lie behind firewalls to separate the internal network from the rest of the Internet. Firewalls can filter all traffic leaving or entering the connected networks based on many packet attributes such as source/destination IP address, source/destination port. They can filter packets also based on protocols, for example, ICMP packets. Denial of Service (DoS) attack is one of the methods by flooding the network with ICMP packets. Therefore, some firewalls do not allow ICMP packets to pass. The host discovery part in this work depends on an ICMP echo request and a TCP packet to port 80. Although it is more reliable than ping the broadcast, the ICMP might also be filtered out by firewalls.

Firewalls often have network address translation (NAT) functionality. The hosts behind a firewall commonly use “private address space”. The IP address identifies to be up may not be an end-host; instead, it might be a router which connects the internal networks and the Internet. Of course, the hosts using private network addresses can not be reached by *traceroute* program.

3.6.2 Jitter problem

Network congestions can introduce jitter in the round-trip-time measurements of *traceroute* program. Because *traceroute* program does not send UDP datagram in parallel, network condition might change between identifying hop N and hop N+1. So, round-trip-time of hop N may be greater than that of hop N+1. This situation occurs more frequently if the source and destination are more than ten hops away.

The IP layer is connectionless and uses “best effort” to deliver packets. Therefore, the routing path of incoming packets might be different from that of outgoing packets. Paxson [14] analyzed 40,000 end-to-end paths and found half of the paths measured were asymmetric.

If jitter problem occurs, the *traceroute* program is repeated for five times. It is not worth wasting time on repeating the *traceroute* program because the jitter problem might be caused by asymmetric routing rather than network congestion.

3.6.3 Problems of *traceroute*

Although *traceroute* is currently the best tool for inferring the end-to-end IP-level path, there are still many problems of traceroute. Janic and van Mieghem [10] have ascertained that 17% of the collected *traceroute* were erroneous. Some ISPs hide their routers from *traceroute* by manipulating ICMP replies. The Rocketfuel project [11] reported that, when performing *traceroute* measurements using different tools (Skitter and Rocketfuel) in the same area with the time difference of two months, Rocketfuel found roughly seven times more links, IP addresses and routers, but some routers and links were only found by Skitter.

Traceroute can also suffer from alias problem since one router can have several interfaces, with one IP address per interface. In order to obtain a router-level map, it is necessary to determine which IP address belongs to the same router.

In this work, we found a strange phenomenon of *traceroute* output. For example, Figure 3-5 lists seventh and twelfth hop of a certain *traceroute* output. By default, the *traceroute* program sends three queries to a router and obtains three RTTs if the router responds. The seventh router replies the first two queries, however, the third queries is replied by another

router. The format of the twelfth hop of *traceroute* output is disordered. The asterisk comes before the DNS name of the router. These two phenomena occur more frequently if traceroute is performed to a remote location. However, the reason is probably the alias problem observed by Janic and van Mieghem [10].

```
7  ldn-bb1-link.telia.net (80.91.249.10)  17.885 ms  17.714 ms
   adm-bb1-pos7-0-0.telia.net(213.248.65.153)  14.133 ms

12  * ae-1-100.ebr1.NewYork1.Level3.net (4.69.132.25)  95.507 ms *
```

Figure 3-5 strange traceroute output

4. Measurements and data

4.1 Destination dataset

Four cities from different continents are chosen for geographic diversity. We cluster IP ranges of each city from geolocation database. After IP ranges are extracted from geolocation database, they are sorted and then written to a file. The number of networks of each city is listed in Table 4-1. San Francisco's distribution of IP networks is fine-grained which has more than eight thousand IP ranges and each range contains less than one thousand IP addresses. Shanghai has only 763 IP ranges and is a bit coarse-grained. Some of the IP ranges of Shanghai are extremely large, that means, they contains a large number of IP addresses.

	Hamburg	Berlin	San Francisco	Shanghai
Number of networks	5118	4234	8476	763

Table 4-1 destination dataset

Since every city has around or more than thousands of IP ranges, scanning the full set of IP ranges is not possible and time-consuming. For each scan, we first reduce the full set of IP ranges to small subset of IP ranges. An array is generated which contains a list of random numbers and no duplicate one is allowed. The upper bound is the number of networks and the lower bound is 1. The size of this array can be 100, 200, 500 and 1,000. The subset is composed of IP ranges corresponding to random number in the array. For example, if the array contains number 4, and then the fourth IP range is chosen as one of the IP ranges in the subset. The number 500 is already a representative number since it covers two-third of networks in Shanghai and one-tenth of networks in Hamburg and Berlin.

4.2 Data format

The scan work is heavy and a lot data are recorded, therefore, a suitable data format must be predefined so that log files can be post processed easier later.

There are three types of files that contain data:

- Individual host-to-host pair

This file contains the information of one pair of hosts. It records the forwarding path, RTT of each intermediate router of each host, the transit node of this pair and hop distance and RTT difference of this pair. The number of this kind log file is huge. For example, if a subset of 100 is scanned and all the networks have an accessible host which *traceroute* can be performed to, as a result, the number of this kind of file will be

4,950. An example of this kind of file is listed in B.1.

- Overall matrix

This file records hop distance and RTT of all pairs. This file is organized as a matrix. The diagonal is written with "N/A". The upper right part of the diagonal contains hop distance and the lower left part of the diagonal records RTT difference. Normally, the file size is quite large. For example, if a subset of 100 is scanned, the file will contain 101 rows with one caption of networks and 100 rows. If no information of a pair exists, an "N/A" is written. An example of this kind of file is listed in B.2.

- Comparison database

This file contains hop distance of each pair from different locations. The same subset is used for different locations. The first two columns represent the number of each network. The third column is the minimum hop distance of this pair. The second column is the hop distance from the experimental host. The following columns contain hop distance from other locations. If *traceroute* is performed from one more location, an addition column is appended to the right most column and the column of minimum hop distance is updated. An example of this kind of file is listed in B.3

4.3 Data overview

Subsets of datasets are used to be scanned and to perform *traceroute* to active hosts. The number of the subsets can be 100, 200, 500 or 1000. The larger the number of subset is, the more accurate data we can obtain. For each destination, we record IP address of each intermediate router as well as round-trip-time from source to that router.

For each pair of end-hosts, we consider three variables: hop distance, round-trip-time and transit node. Hop distance and round-trip-time are two metrics to measure network distance. The transit node is different for each pair of end-hosts since they may belong to different ASes and the peering points for ASes are different.

4.3.1 Hamburg

Table 4-2 lists all scanned data of Hamburg. During two months of scanning, data of fifteen subsets have been obtained. The mean hop distances of each subset are between 10 and 12 hops and the mean hop distance of all subsets is around 10 hops. The mean RTTs of each subset range from 55 ms to 87 ms and the mean RTT of all subsets is around 70 ms. Padmanabhan and Subramanian [2] suggest that 90% of nodes within 5 ms round-trip-time are located within a radius of 50 km and 90% of nodes within 10 ms round-trip-time are located within a radius of 300 km. If we suppose two end-hosts are located symmetric to the transit node which means it takes 5 hops and 35 ms from one end-host to the transit node on

average. It suggests that most of the transit nodes are not located in Hamburg since 35 ms is much larger than the threshold value 10 ms.

Date_time	Number of networks [#]	Number of samples [#]	Mean hop distance [hops]	Mean RTT [ms]
20060726_114436	200	6105	10.355	81.206 ms
20060726_205735	200	6216	10.506	69.247 ms
20060728_202345	200	7260	10.677	56.736 ms
20060802_142704	200	8256	10.969	55.375 ms
20060803_155825	200	8385	10.496	69.100 ms
20060807_144843	200	7021	10.510	75.647 ms
20060808_202012	500	48205	10.707	75.216 ms
20060810_205630	500	47586	10.480	68.936 ms
20060811_200506	500	46971	10.442	68.054 ms
20060815_192410	500	48516	10.756	87.209 ms
20060821_192300	500	47586	10.516	80.091 ms
20060901_201950	1000	176715	10.842	62.621 ms
20060922_183115	100	1035	12.124	59.416 ms
20060926_230110	200	6903	10.996	66.187 ms
20061002_183049	500	39903	10.004	79.759 ms
Mean value			10.692	70.320 ms

Table 4-2 data overview of Hamburg

4.3.2 Berlin

Table 4-3 lists all scanned data of Berlin. Data of thirteen subsets of Berlin have been obtained. The mean hop distances of each subset are between 11 and 12 hops and the mean hop distance of all subsets is around 11 hops. The mean RTTs of each subset range from 73 ms to 140 ms and the mean RTT of all subsets is around 93 ms. It also suggests that the most transit points of each pair are not located in Berlin as well. Both cities are located in the same country, but the overall mean hop distance of Berlin is one hop more than that of Hamburg and the overall mean RTT is about 23 ms larger than that of Hamburg. The reason for that difference is that the test machine is located in Hamburg. If the traceroute probes are sent from a location in Berlin, the results might be different.

Date_time	Number of networks [#]	Number of samples [#]	Mean hop distance [hops]	Mean RTT [ms]
20060726_024053	200	6555	11.407	139.823 ms
20060726_171319	200	6441	11.574	87.435 ms
20060731_042056	200	7750	11.634	90.671 ms
20060802_150317	200	8128	11.691	101.587 ms
20060803_152848	200	7503	11.670	73.229 ms

20060807_142738	200	7260	12.039	75.253 ms
20060808_065115	500	37128	12.011	75.037 ms
20060809_041654	500	38226	11.958	74.123 ms
20060811_042452	500	38503	11.637	84.531 ms
20060811_222407	500	45451	11.462	79.852 ms
20061002_163723	200	6670	12.001	94.659 ms
20061002_232639	500	45753	12.200	108.601 ms
20061004_161750	100	2016	12.694	124.485 ms
Mean value			11.845	93.022ms

Table 4-3 data overview of Berlin

4.3.3 San Francisco

Table 4-4 lists all scanned data of San Francisco. Less data have been obtained for San Francisco and there are only seven subsets. The mean hop distances of each subset are between 24 and 26 hops and the mean hop distance of all subsets is around 25 hops. The mean RTTs of each subset range from 311 ms to 355 ms and the mean RTT of all subsets is around 325 ms. The value of mean hop distance is twice greater than that of Hamburg and Berlin. The value of mean RTT is four to five times greater than that of Hamburg and Berlin. It suggests that the transit nodes are far away from end-hosts. Because San Francisco is located in another country, the packets are routed via nationwide ISPs or worldwide ISPs. The peering points of these ISPs may be located in several cities, which are not all visible to the test machine.

Date_time	Number of networks [#]	Number of samples [#]	Mean hop distance [hops]	Mean RTT [ms]
20060726_125729	200	2926	25.151	333.011 ms
20060802_180923	200	4095	24.198	355.280 ms
20060803_195448	200	4371	24.845	311.377 ms
20060807_161808	200	4278	25.802	313.083 ms
20060809_030747	500	24310	24.972	318.268 ms
20060812_003223	500	25200	24.701	324.361 ms
20061107_202944	200	4005	22.219	320.207 ms
Mean value			24.556	325.084 ms

Table 4-4 data overview of San Francisco

4.3.4 Shanghai

Table 4-5 lists all scanned data of Shanghai. Data of eight subsets have been obtained. The mean hop distances of each subset are between 20 and 25 hops and the mean hop distance of all subsets is around 22 hops. The variance of hop distance is much larger than

that of other cities. The mean RTTs of each subset range from 619 ms to 772 ms and the mean RTT of all subsets is around 680 ms. The value of mean hop distance is twice greater than that of Hamburg and Berlin. The value of mean RTT is about ten times greater than that of Hamburg and Berlin.

Date_time	Number of networks [#]	Number of samples [#]	Mean hop distance [hops]	Mean RTT [ms]
20060727_191127	200	5671	21.224	638.116 ms
20060728_232308	200	5460	20.401	618.765 ms
20060802_174350	200	8518	23.721	712.041 ms
20060803_210942	200	9869	21.995	681.841 ms
20060807_211101	200	9870	22.702	683.841 ms
20060809_053425	500	47278	21.446	670.503 ms
20060812_084532	500	37950	20.952	665.045 ms
20060930_011959	200	6786	25.396	771.843 ms
Mean value			22.230	680.249 ms

Table 4-5 data overview of Shanghai

5. Data analysis and evaluation

5.1 Reliability of GeoIP®

The accuracy of GeoIP database has great impact on the results of the experiment. If the database offers a wrong location, the routing path is totally different because the IP address is located in another city. As a result, the estimation of edge distance will be erroneous.

In order to verify the accuracy of GeoIP database, a test is done to compare the clustered IP addresses with the query result from WHOIS database [20]. Although, the query results from WHOIS database is not the most trustworthy. If the results are the same from both databases, the location of this IP address is considered reliable.

Checking every clustered IP addresses is not feasible and time-consuming. Only one IP address is checked within one IP range. In each range, one IP address is randomly chosen. Then, query WHOIS sever with this IP address. The returned content of WHOIS query may contain address and description information of this IP address. If this content contains name of the city to be checked, it is considered to be correct.

The results of comparison are listed in Table 5-1.

	Hamburg	Berlin	San Francisco	Shanghai
Number of tested ranges	100	100	200	763
Number of identical results	87	80	116	697
Number of disparate results	13	20	84	48
Number of unallocated	0	0	0	20
Ratio of correctness	87%	80%	58%	91.35%
resource	RIPE	RIPE	ARIN	APNIC

Table 5-1 reliability of GeoIP Database

Since Shanghai has only 763 IP ranges, all of them are checked. While other cities have more than five thousand IP ranges, it is not possible to validate every IP ranges. A subset of each city is stochastically chosen. The number of these subsets is around one-fifth of the number of the full IP range. From Table 5-1, it is obvious that Shanghai has the highest ratio of correctness. Shanghai has twenty unallocated IP ranges; it is maybe that the owner of these IP addresses has not registered them to WHOIS. Hamburg and Berlin have a ratio higher than 80% which is still an acceptable value. However, San Francisco experiences with a low value of a bit over 50%. WHOIS database has it own disadvantage, since organizations or ISPs register with their headquarters. But many ISPs are spread across different geographic regions and it causes significant errors. For example, Deutsche

Telekom AG is a nationwide ISP and it assigns its IP addresses to various locations in Germany. However, within WHOIS database, only the headquarter location Nuernburg is returned.

5.2 Network transparency

The destinations of the *traceroute* probes are randomly chosen from clustered IP ranges; therefore, it is not possible to know when the destination is up and whether it is hidden from a firewall. The accessible ratio changes with day and time. For example, during weekday, most hosts from companies are up while in the weekend, most Internet traffic is from home user. If the destination host is a web server by chance, it may be always up.

If there is no accessible host in one IP range, it may be due to the following reasons: 1) there is no active host in this IP range; 2) there are active hosts in this IP range, but no host is accessible; 3) there are accessible hosts, but because of jitter problem and no *traceroute* data is obtained.

Table 5-2 lists the accessible ratio of Hamburg. More than half of IP ranges have at least one accessible host, but there is one extreme case of 46%. The average accessible ratio is approximately 60%, which is not high but still a reasonable value.

Date_time	Number of networks [#]	Number of accessible networks [#]	Number of samples [#]	Ratio of accessibility [%]
20060726_114436	200	111	6105	55.5%
20060726_205735	200	112	6216	56%
20060728_202345	200	121	7260	60.5%
20060802_142704	200	129	8256	64.5%
20060803_155825	200	130	8385	65%
20060807_144843	200	119	7021	59.5%
20060808_202012	500	311	48205	62.2%
20060810_205630	500	309	47586	61.8%
20060811_200506	500	307	46971	61.4%
20060815_192410	500	312	48516	62.4%
20060821_192300	500	309	47586	61.8%
20060901_201950	1000	595	176715	59.5%
20060922_183115	100	46	1035	46%
20060926_230110	200	119	6903	59.5%
20061002_183049	500	283	39903	56.6%
Mean value				59.48%

Table 5-2 ratio of accessible networks of Hamburg

Table 5-3 lists the accessible ratio of Berlin. All the values are around 60% and there is

no great variance. The mean value is also around 60% which is similar to Hamburg.

Date_time	Number of networks [#]	Number of accessible networks [#]	Number of samples [#]	Ratio of accessibility [%]
20060726_024053	200	115	6555	57.5%
20060726_171319	200	114	6441	57%
20060731_042056	200	125	7750	62.5%
20060802_150317	200	128	8128	64%
20060803_152848	200	123	7503	61.5%
20060807_142738	200	121	7260	60.5%
20060808_065115	500	273	37128	54.4%
20060809_041654	500	277	38226	55.4%
20060811_042452	500	278	38503	55.6%
20060811_222407	500	302	45451	60.4%
20061002_163723	200	116	6670	58%
20061002_232639	500	303	45753	60.6%
20061004_161750	100	64	2016	64%
Mean value				59.34%

Table 5-3 ratio of accessible networks of Berlin

Table 5-4 lists the accessible ratio of San Francisco. The accessibility of networks in San Francisco is poor. More than half of the IP ranges don't have an accessible host. One possible reason for this low value is that San Francisco is most fine-grained and has more than 8,000 IP ranges. Each range contains no more than 1,000 IP addresses. Because IP range clustering is done by MaxMind, we cannot know how they have done. However, if a network contains less IP address, the probability to find an active host is much lower than a IP range with a large number of IP address. The other possible reason is that more than half of IP ranges belong to Class A IP network. Some of them are rarely used in real-world such as 4.0.0.0.

Date_time	Number of networks [#]	Number of accessible networks [#]	Number of samples [#]	Ratio of accessibility [%]
20060726_125729	200	77	2926	38.5%
20060802_180923	200	91	4095	45.5%
20060803_195448	200	94	4371	47%
20060807_161808	200	93	4278	46.5%
20060809_030747	500	221	24310	44.2%
20060812_003223	500	225	25200	45%
20061107_202944	200	90	4005	45%
Mean value				44.529%

Table 5-4 ratio of accessible networks of San Francisco

Table 5-5 lists the accessible ratio of Shanghai. Shanghai has the highest ratio of

accessibility. On the contrast of San Francisco, IP ranges are coarse-grained. Some of IP ranges contain more than ten thousand of IP addresses. The probability of finding an accessible host is of course much higher than an IP range having only few IP addresses.

Date_time	Number of networks [#]	Number of accessible networks [#]	Number of samples [#]	Ratio of accessibility [%]
20060727_191127	200	107	5671	53.5%
20060728_232308	200	105	5460	52.5%
20060802_174350	200	131	8518	65.5%
20060803_210942	200	141	9870	70.5%
20060807_211101	200	141	9870	70.5%
20060809_053425	500	308	47278	61.6%
20060812_084532	500	276	37950	55.2%
20060930_011959	200	117	6786	58.5%
Mean value				60.98%

Table 5-5 ratio of accessible networks of Shanghai

5.3 Analysis of single-origin data

This section presents the results of single-origin data. There are two forms of delay distributions to be analyzed and they are hop count distributions and round-trip-time distributions. All of the data presented in this section are obtained from a single source, which locates in Hamburg. As mentioned in the previous section, number of accessible networks varies with size of subset, dates and cities; therefore, number of samples obtained of each scan is different. Relative frequency of samples is used to represent data in order to make fair comparisons. Statistical diagrams are used to show data variance of different scan results. For each city, the scale of diagrams is normalized.

5.3.1 Hop count distributions

Hop counts are discrete values, so number of each hop distance can be easily counted. Two forms of hop count distributions are depicted: one of relative frequency and the other of cumulative distribution function (CDF). We first explore hop count distributions individually of four cities and delineate character of them. We compare scan results of daytime and nighttime, varying size of subset, various dates to find out whether these factors affect scan results and how significant influence they are. Then, we make a comparison of four cities to analyze variances of different regions. Finally, we present a scan result of randomly-chosen destinations and compare with clustered destinations.

- Hamburg

Hamburg is the first city to be analyzed because the test machine is located in Hamburg. The curves of relative frequencies look quite similar. They have the following similarities: 1) most of them have two peaks. One small peak is round hop count 5 and the other large one is around hop count 11. 2) Most of them have a hollow between hop count 6 and 7. 3) After hop count 14, the values of relative frequencies go fast to zero. The ramp of right part of the curve is quite steep. Hop distance of 5 indicates that the peering point of these two networks is probably in Hamburg and the transit node might locate in Hamburg as well. Hop distance of 11 means the transit node is a bit far from both destinations. The packet might travel through two regional- or nationwide ISPs and peering point locates somewhere else rather than Hamburg. The backbone network of test machine is German Research Network, which peers with other ISPs mainly in Frankfurt, thereby hop distance 11 gains the most weights of the whole hop distances. The cumulative curves do not differ very much from each other. They reach 50% between hop count 10 and 11. All the curves are steep between hop count 3 and 15 and become almost a straight line after hop count 15.

First of all, hop count distributions of daytime and nighttime are compared, which is shown in Figure 5-1. These two experiments were done on the same day and with the same subset of IP ranges; one of the scanning parts was done during daytime while the other one was done during nighttime. These two curves are quite the same except that peak value of daytime is 1% greater than that of nighttime. As shown in Table 5-2, the number of accessible networks of daytime is one less than that of nighttime, which is not a significant value.

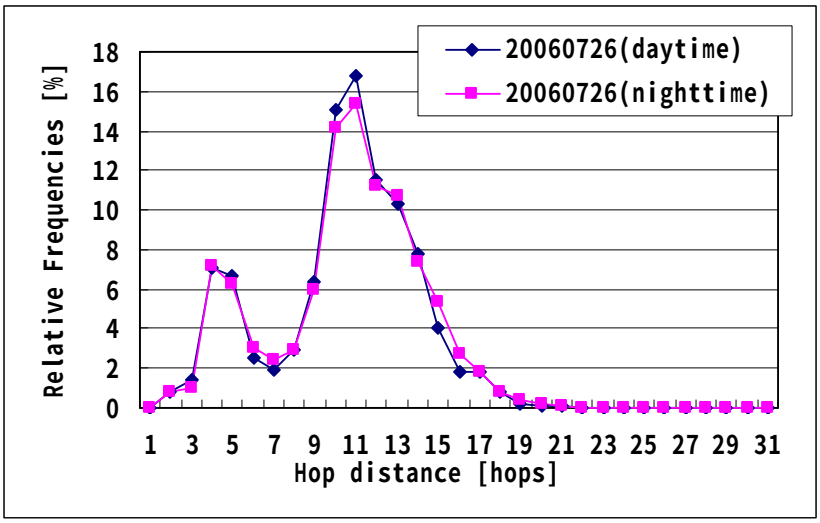

Figure 5-1 Hamburg, 200 networks, daytime vs. nighttime

Figure 5-2 compares hop distance distributions of various sizes of subsets. These four datasets were scanned on close dates. These four curves look quite dissimilar. The peak values of 100 networks and 200 networks are at the same hop distance. The curve of 100 networks shows great variances of other curves, which indicates 100 networks out of more than 5,000 networks is not a representative quantity. The shape of 500 networks and 1000

networks are much more similar than other two curves.

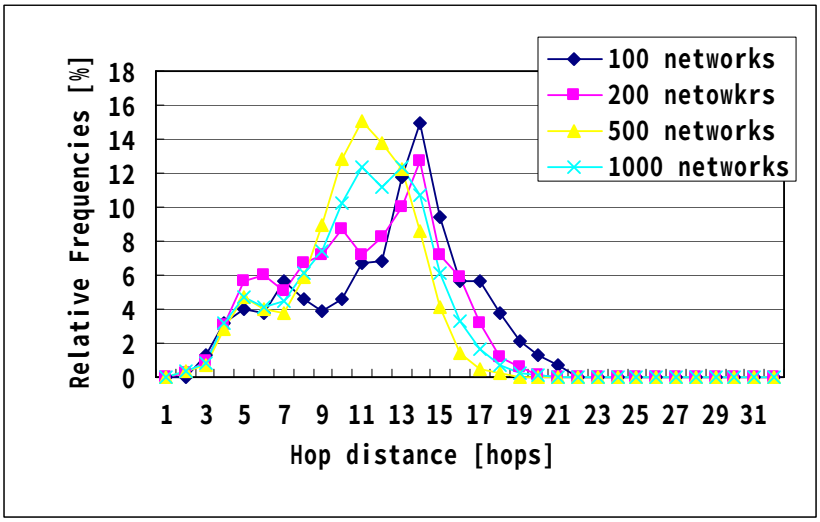

Figure 5-2 Hamburg, various sizes of subsets

Figure 5-3 compares hop count distributions on various dates. The size of all the subsets is 200. It is apparent that the variances of different dates are quite large. Only curves on date 20060726 and 20060728 show great similarity. If the first and the last date are considered, which have time difference of two months, both the peak value and shape of the curves are changed. Since Hamburg have more than 5,000 networks, a subset of 200 is comparatively a quite small value and hop count distributions depends largely on which networks have been chosen.

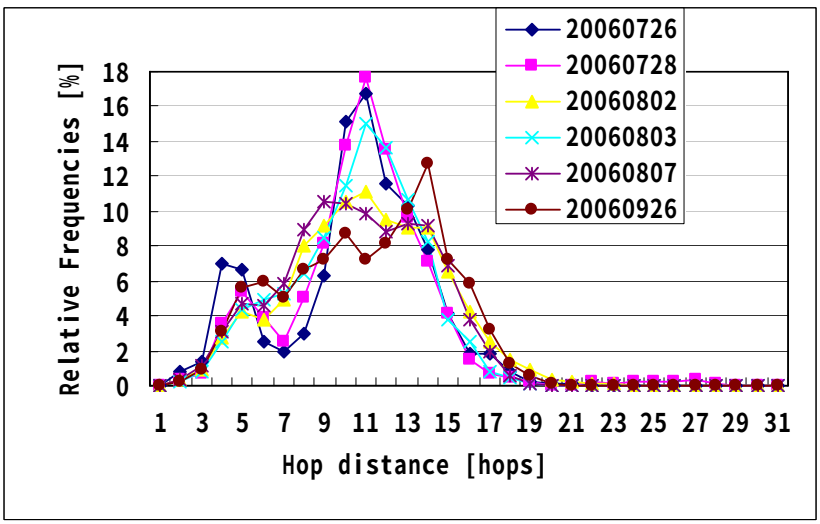

Figure 5-3 Hamburg, 200 networks on various dates

Figure 5-4 compares as well hop count distributions on various dates, but the size of subsets is 500.The shape of these curves are almost the same and hop count distributions are

much more similar than that of 200 networks. One exception is the curve on date 20061002, which is shifted by two hops to the left but retains the shape. This curve has a time difference of two months of other curves, which infers routing paths might change after a certain time interval.

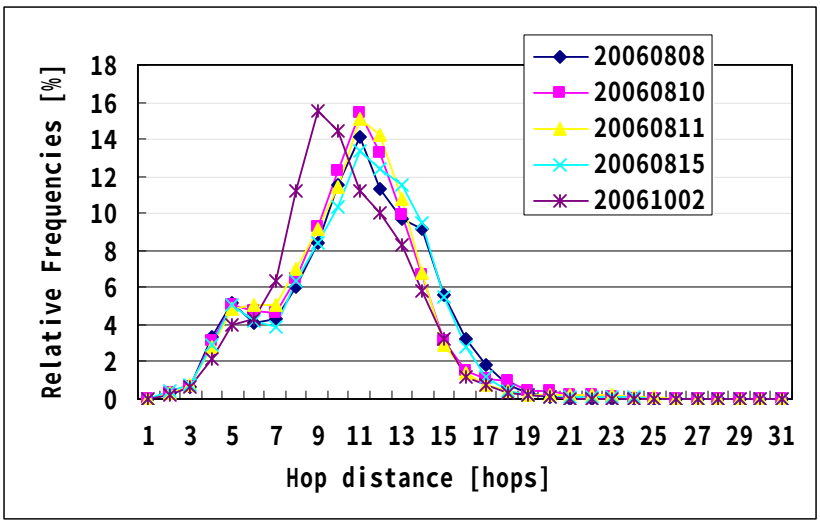

Figure 5-4 Hamburg, 500 networks on various dates

- Berlin

The second city to be analyzed is Berlin. Most curves of relative frequencies have three peaks at hop count 5, 11 and 14. Some of them are not quite obvious. As mentioned early, hop distance of 5 means that the transit node locates in Berlin while hop count of 11 and 14 indicates a transit node outside Berlin. There are two hollows at hop count 7 and 12. The shape of hop count distributions of Berlin is quite similar to that of Hamburg, because both cities locate in the same country and no globe transit providers are involved. All the cumulative curves are quite similar. They reach 50% between hop count 11 and 12. They are quite steep between hop count 4 and 16 and become plain after hop count 16.

Figure 5-5 compares hop count distributions of daytime and nighttime. These two experiments were done on the same day and with the same subset of IP ranges as well. The values of hop distances are quite the same expect for small deviations at peak area. This result is identical to that of Hamburg. The number of accessible networks of daytime is one more than that of nighttime, which is on the contrary to Hamburg.

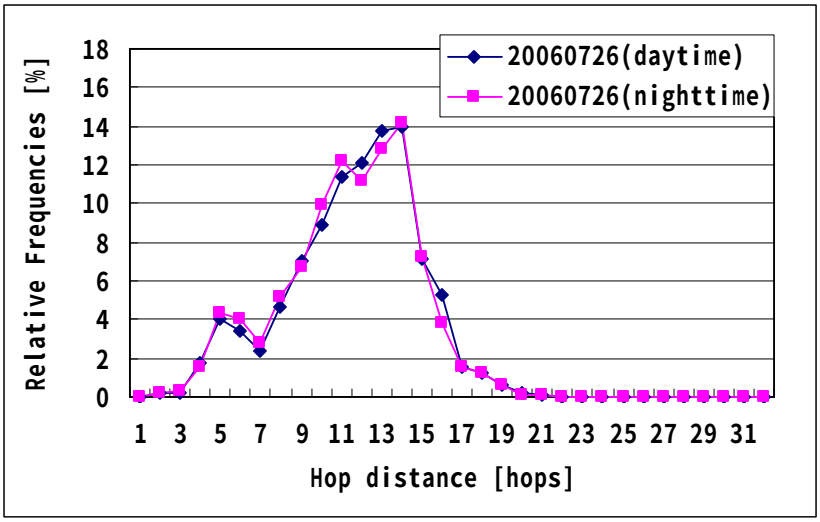

Figure 5-5 Berlin, 200 networks, daytime vs. nighttime

Figure 5-6 compares hop distance distributions of various sizes of subsets. So far, no data of 1000 networks is obtained. These three curves look similar especially for 200 networks and 500 networks. The curve of 100 networks gains fewer weights at peak area but has a larger value than the other two between hop count 17 and 19. Berlin has more than 4,000 networks and a subset of 100 networks is also not a representative one.

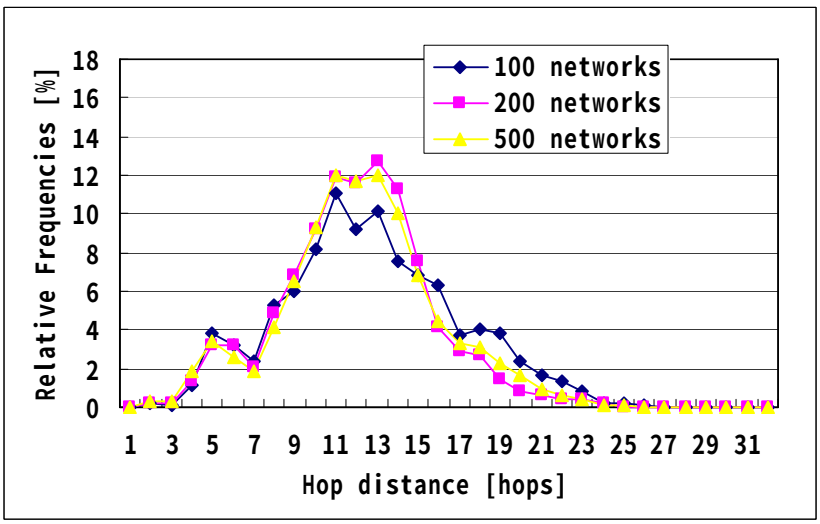

Figure 5-6 Berlin, various sizes of networks

Figure 5-7 compares hop count distributions on various dates. The size of all the subsets is 200. Except for the peak area, the other parts of the curves are similar. The hop count distributions are more stable than that of Hamburg with the same size.

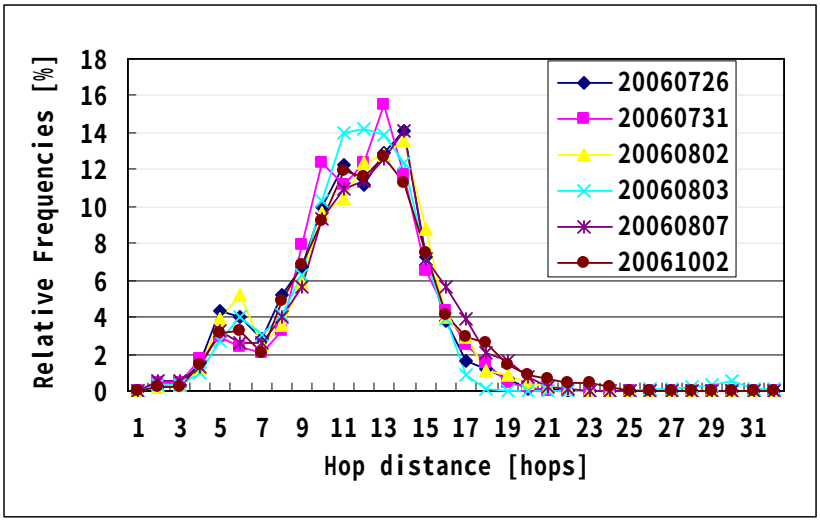

Figure 5-7 Berlin, 200 networks on various dates

Figure 5-8 compares as well hop count distributions on various dates, but the size of subsets is 500. The great variance is at the peak area. In the opposite of Hamburg, the hop count distributions of 200 networks are much stable than that of 500 networks.

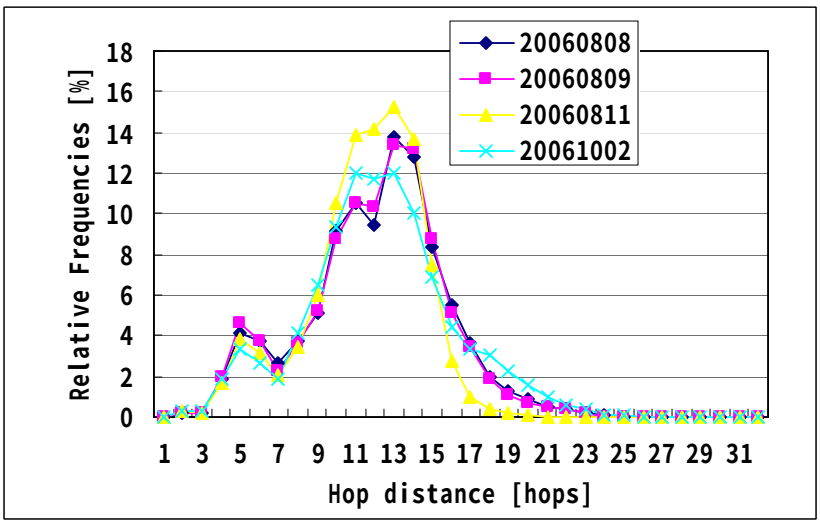

Figure 5-8 Berlin, 500 networks on various dates

- San Francisco

The third city to be explored is San Francisco, which locates in another country of the test machine. The curves of relative frequencies are more diverse than that of Hamburg and Berlin. The curves spread much wider and the peak value is around 7%, which is much lower than that of Hamburg and Berlin. The first part of the curve (between hop count 1 and 16) has a value less than 1% while the middle part (between hop count 22 and 34) has a much larger value. Since San Francisco locates in another country, nationwide and international ISPs are involved in the routing. Most curves reach their peak value at hop count 27, which infers the

transit node is still in Germany. The cumulative curves are not distinct from each other. Compared with the curves of Hamburg and Berlin, they are much flatter at the first part and the slop of middle is not so steep.

Figure 5-9 compares hop count distributions on various dates. The size of all the subsets is 200. The diagram shows great variance of hop count distributions on different dates. Only the peak areas are located closely for each curve.

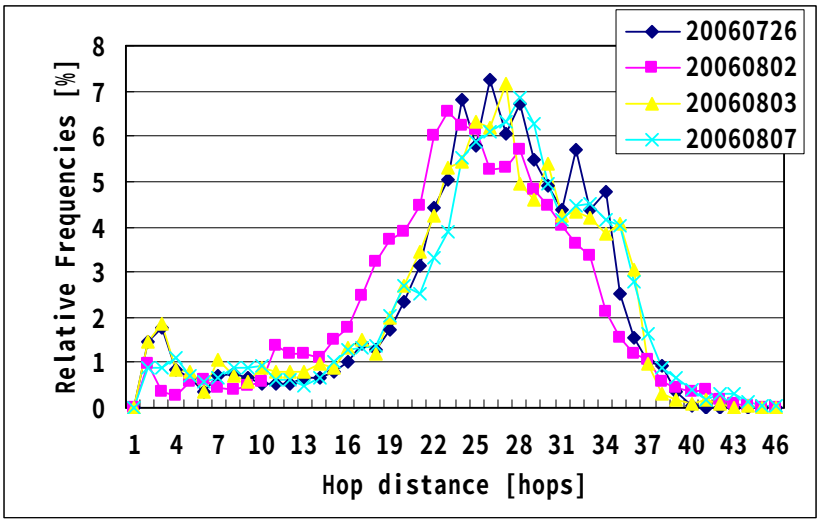

Figure 5-9 San Francisco, 200 networks, on various dates

Figure 5-10 compares as well hop count distributions on various dates, but the size of subsets is 500. These two hop count distributions do not differ very much from each other. But there are only two dates which are close to each other, it is hard to judge whether dates have influence on hop count distributions.

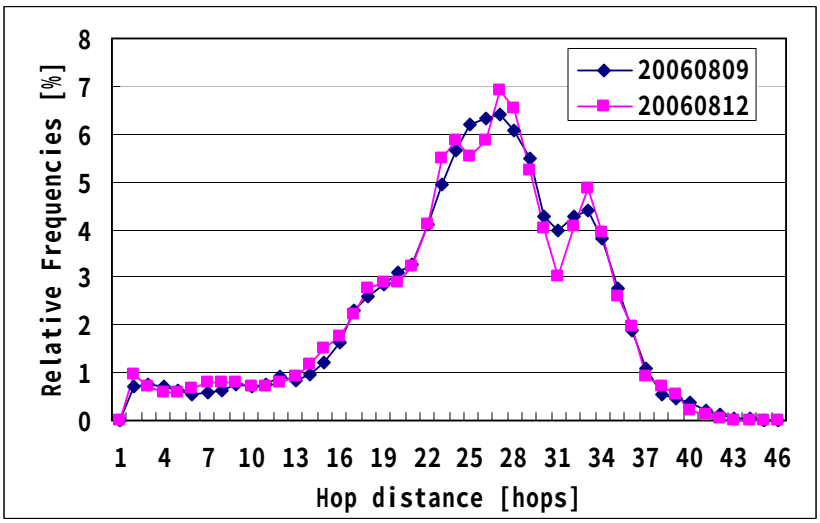

Figure 5-10 San Francisco, 500 networks, on various dates

- Shanghai

The last city to be analyzed is Shanghai. Most curves of relative frequencies have two peaks and their values do not differ very much from each other. One peak value is at hop count 15 and the other one is at hop count 27. Hop count of 27 infers that the transit node is far away from both destinations. Shanghai is the most remote destination of Hamburg, since most packets originating in Europe and destined to China travel via USA. Hop count of 27 means there are 14 hops from the transit node to both destinations on average, which probably manifests the transit node is USA or even in Germany. The cumulative curves are the most diverse among four cities. The middle part of curve is not so smooth but quite winding compared with other cities.

Figure 5-11 compares hop count distributions on various dates. The size of all the subsets is 200. These curves are not quite similar. Especially the hop count distribution on date 20060930, with time difference of two months, it varied a lot from other curves. This great variance shows the locations of the transit nodes are quite disparate.

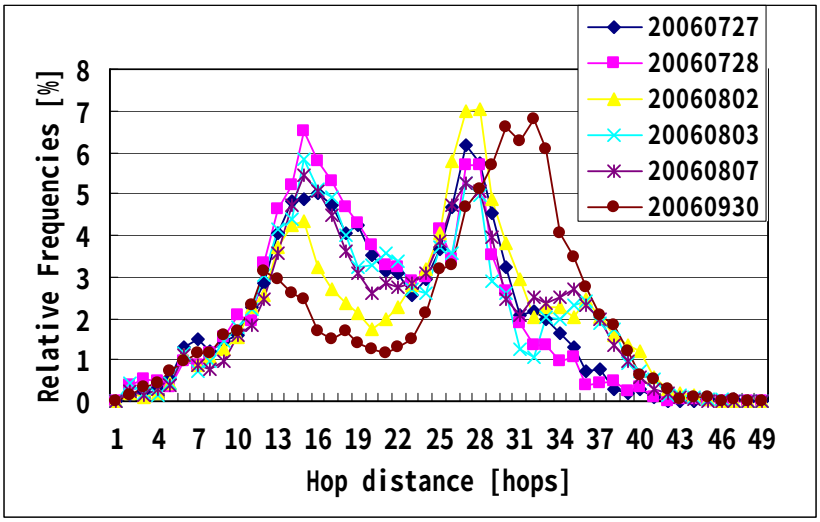

Figure 5-11 Shanghai, 200 networks, on various dates

Figure 5-12 compares as well hop count distributions on various dates, but the size of subsets is 500. These two hop count distributions have slight difference from each other. Because two hop count distributions were obtained from two close dates, it is also hard to judge whether dates have influence on hop count distributions.

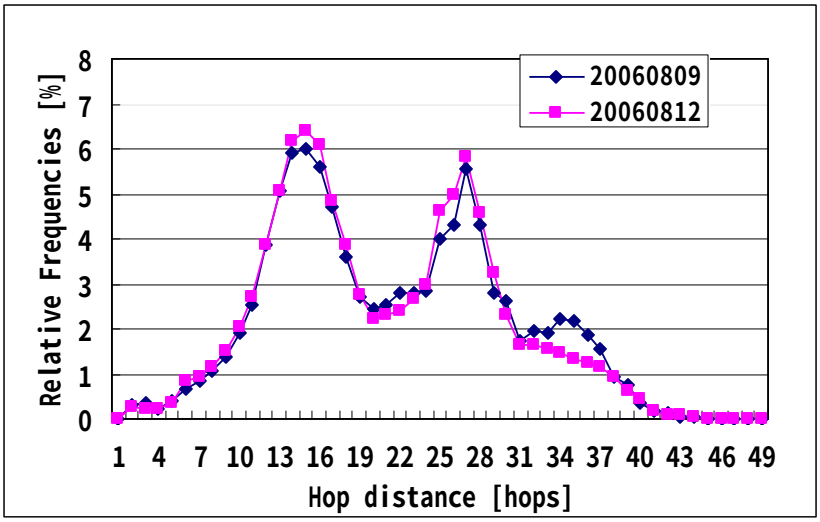

Figure 5-12 Shanghai, 500 networks, on various dates

- Comparison of four cities

Figure 5-13 compares hop count distributions of these four cities. The size of all the subsets is 200. The curves of Berlin and Hamburg are narrow and the peak values are high, while the curves of San Francisco and Shanghai spread much wider. Compared with Berlin and Hamburg, the peak values of San Francisco and Shanghai are "right-shifted" because the transit node locates no near to both destinations.

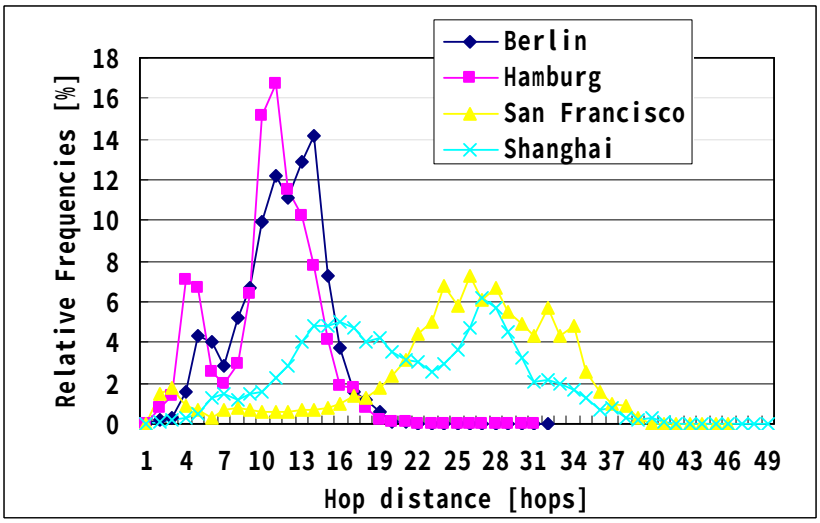

Figure 5-13 four cities, 200 networks

Figure 5-14 compares as well hop count distributions of these four cities, but the size of all the subsets is 500. The situation is the same as 200 networks. The curves of Berlin and Hamburg show great similarity in shape and peak value. The curves of San Francisco and Shanghai spread much wider and they are dissimilar to other cities.

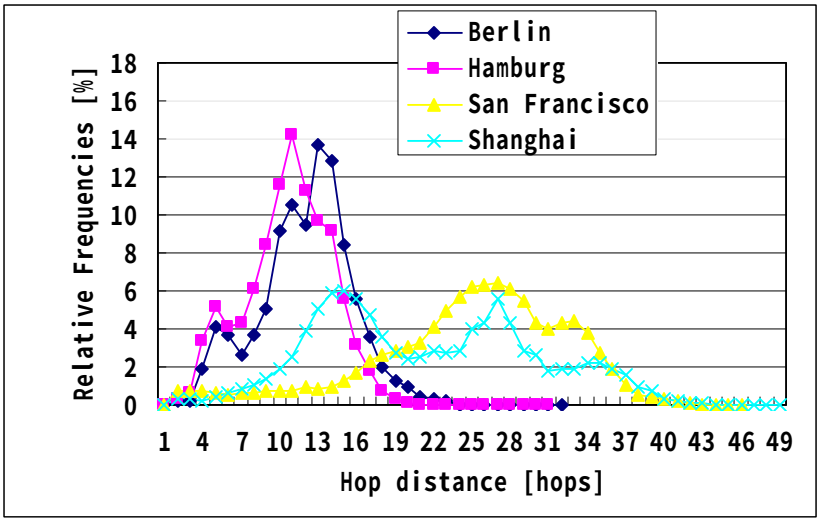

Figure 5-14 four cities, 500 networks

- Random destinations

Figure 5-15 shows hop count distribution and cumulative distribution function of a random-destination dataset of size 200. A random-destination dataset means the 200 IP ranges are chosen randomly from MaxMind GeoIP database. The locations of these IP ranges are unknown. The mean hop distance of this dataset is 21.229, which is much higher than that of Hamburg and Berlin and close to that of San Francisco and Shanghai.

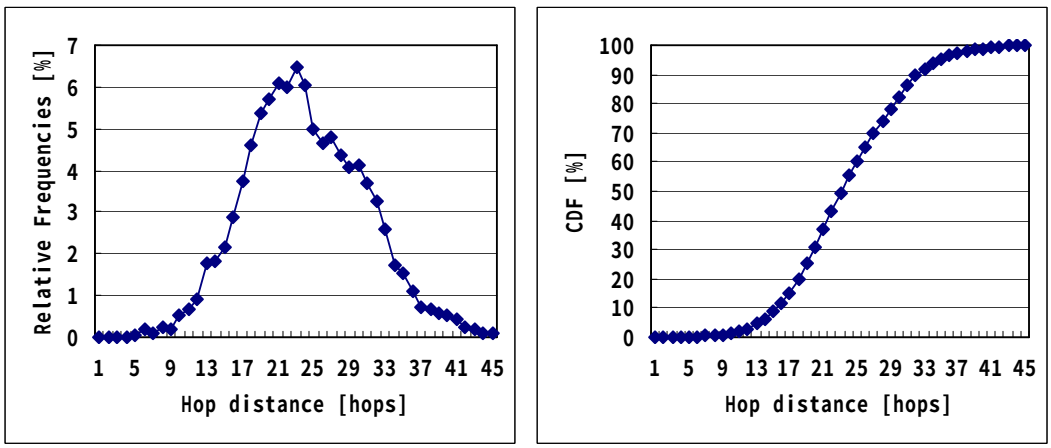

Figure 5-15 random, 200 networks

If the hop distance distribution of random-destination dataset is compared with that of four cities, it is much more similar to hop distance distributions of San Francisco and Shanghai which spread wider. The transit nodes for each pair of random-destinations are ideally uniformly distributed all over the world, so the similarity of random-destination dataset and San Francisco/Shanghai indicates the locations of the transit nodes are widely distributed, e.g. in different countries/cities. Therefore, the data obtained for San Francisco/Shanghai from the test location are to some extent very imprecise.

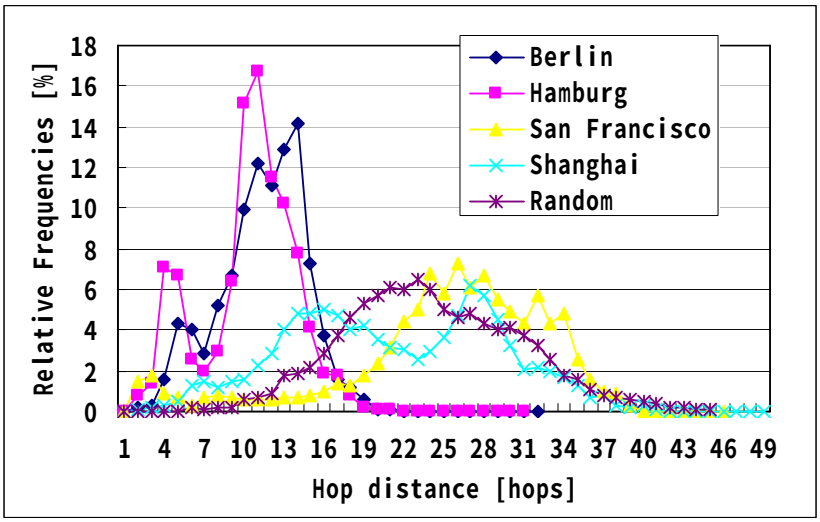

Figure 5-16 random vs. four cities, 200 networks

- Summary

In this section, we compare different scan results of four cities. From the results of Hamburg and Berlin, we find time during a day does not influence much on hop count distributions. The size of subsets effect the scan results as well, because a small quantity is not representative if it is less than one-fifth of the full set. The hop count distributions depend on dates as well, since network topology might be changed during a long time interval, for example, several months.

Four cities show great variances of hop count distributions except Hamburg and Berlin have great similarities than other twos. Hamburg and Berlin are in the same country as the test machine, thereby packets travel only through customer or regional ISPs and maybe sometimes national ISPs. The situation of San Francisco and Shanghai are quite different, since they are located in another country. We find some of the transit nodes belong to German Research Network, which still locate in Hamburg. Therefore, these data are quite meaningless. Of course, two hosts located in San Francisco will not exchange their packets through a router in Hamburg. One possible reason is that the some routers along the routing do not answer to *traceroute* probes; therefore, no IP address is recorded for those routers. The determination of transit node depends largely on IP address, if a router does not answer, which is likely to be the transit node, and then the closer transit node is lost.

All the diagrams of hop count distributions of four cities are shown in Appendix C. The diagrams on the left part represent hop distance versus relative frequencies while the right part shows the cumulative distribution functions.

5.3.2 Round-trip-time distributions

Since round-trip-times are not discrete values, we use delay bins to classify measurement results. They are 0-10 ms, 11-20 ms etc. Each bin contains an equal time interval. Round-trip-time is a critical metric to measure network distance as CAIDA people conclude [4]: “hop count is not a representative metric for expressing Internet connectivity geographically.” Usually, these curves have a long tail, which might be caused by network congestion or jitter problem. We truncate long tails and focus on most critical part of the round –trip-time distributions. The comparison procedure is the same as hop count distributions. We first explore round-trip-time distributions individually of four cities and describe character of them. We compare scan results of daytime and nighttime, varying size of subset, various dates to find out whether these factors affect round-trip-time and how significant influence they are. Then, we make a comparison of four cities to analyze variances of different regions. Finally, we present a scan result of randomly-chosen destinations and compare with clustered destinations.

- Hamburg

The first city to be explored is Hamburg. The round-trip-time distributions of Hamburg look like “a mirror” of hop count distributions. Most of the curves have two peaks: one around 35 ms (delay bin of 31-40 ms) and the other one at 85 ms (delay bin of 81-90 ms).

Figure 5-17 compares round-trip-time distributions of daytime and nighttime. These two experiments were done based on the same subset of networks. The variance of round-trip-time distributions is much greater than that of hop count distributions. The curve of nighttime gains much more weights on lower value of round-trip-time than that of daytime. The diagram infers that time during a day influences the round-trip-time distributions and on average the round-trip-time during daytime is higher than during nighttime. Form the result of Hamburg, time during a day does influence much on hop count distributions.

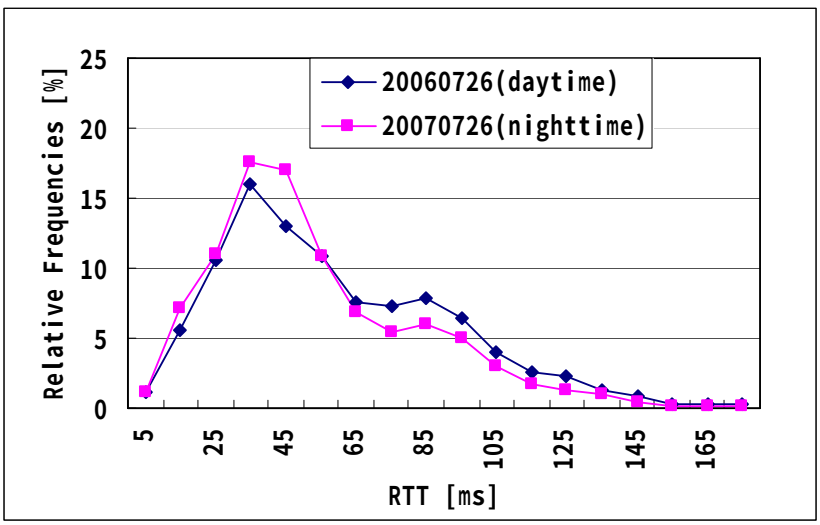

Figure 5-17 Hamburg, 200 networks, daytime vs. nighttime

Figure 5-18 compares round-trip-time distributions of various sizes of subsets. These four datasets were scanned on close dates. Except for 100 networks, the other three curves reach their peak at value 25 ms (delay bin 21-30 ms). The curve of 100 networks reaches at value 35 ms (delay bin 31-40 ms). As mentioned in the section of hop count distributions, number of 100 networks is not a representative quantity of Hamburg. The curve of 1000 networks gains most weights at lower round-trip-times.

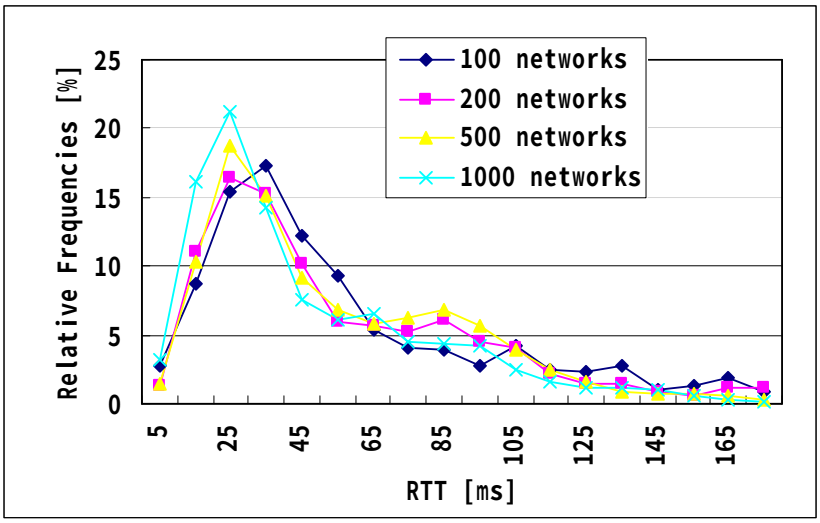

Figure 5-18 Hamburg, various sizes of networks

Figure 5-19 compares round-trip-time distributions on various dates. The size of all the subsets is 200. The most similar part of these curves is peak area. Other parts of the curves are diverse. If the first and the last date are considered, which have time difference of two months, the curve of the last date (20060926) gains much more weights on lower value of round-trip-times than that of the first date (20060726).

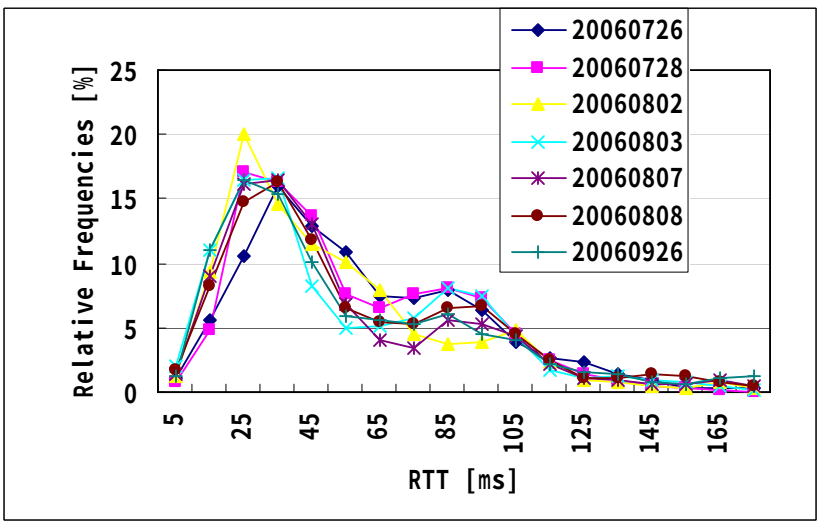

Figure 5-19 Hamburg, 200 networks, on various dates

Figure 5-20 compares as well round-trip-time distributions on various dates, but the size of subsets is 500. The similarity of these curves is much higher than that of 200 networks. One exception is the curve on date 20061002, whose peak value is much higher than others. Refer to hop count distribution which has the same comparison of various dates. The curve of date 20061002 has also a great variance of other curves, which is coincide with round-trip-time distribution.

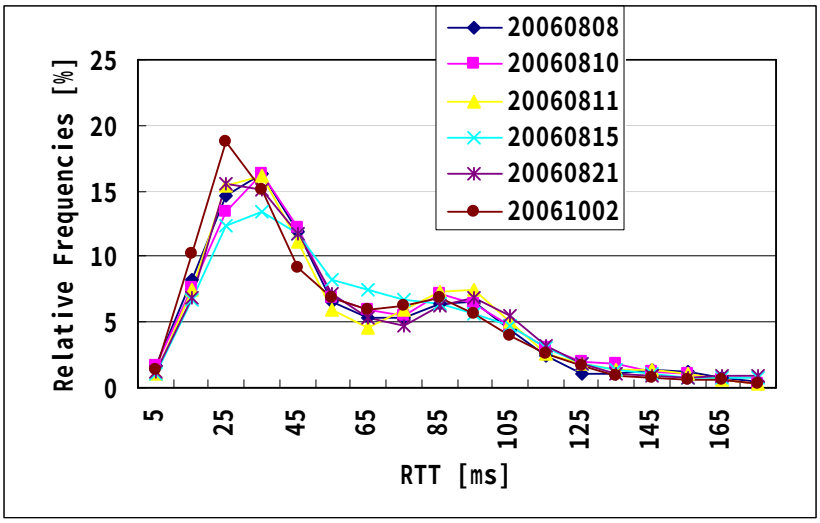

Figure 5-20 Hamburg, 500 networks, on various dates

- Berlin

The second to be analyzed is Berlin. Most of round-trip-time distributions reach their peak at value of 35 ms (delay bin 31-40 ms), which is identical to Hamburg. All of the curves have a long tail which might be caused by network congestion or jitter problem. Berlin has a much longer tail than Hamburg, which directly influences mean round-trip-time. Therefore, the mean round-trip-time of Berlin is about 23 ms larger than that of Hamburg.

Figure 5-21 compares round-trip-time distributions of daytime and nighttime. These two experiments were done based on the same subset of networks. The result is different from that of Hamburg. The curve of daytime gains much more weights on lower value of round-trip-time than that of nighttime. Although time during a day does not influence much on hop count distributions, it does have great impact on round-trip-time distributions.

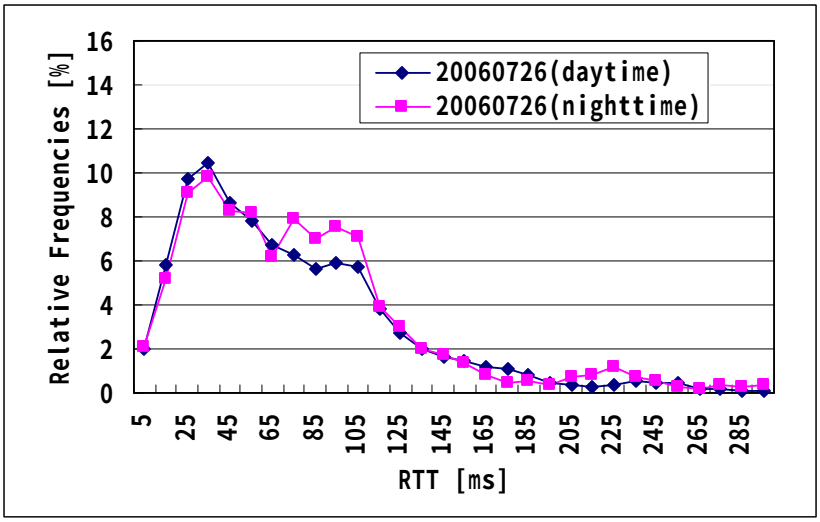

Figure 5-21 Berlin, 200 networks, daytime vs. nighttime

Figure 5-22 compares round-trip-time distributions of various sizes of subsets. These four datasets were scanned on close dates. All of the three curves reach their peak at value 35 ms (delay bin 31-40 ms), which is one more delay bin than that of Hamburg. These three curves present great diversity, especially for curve of 100 networks and 500 networks.

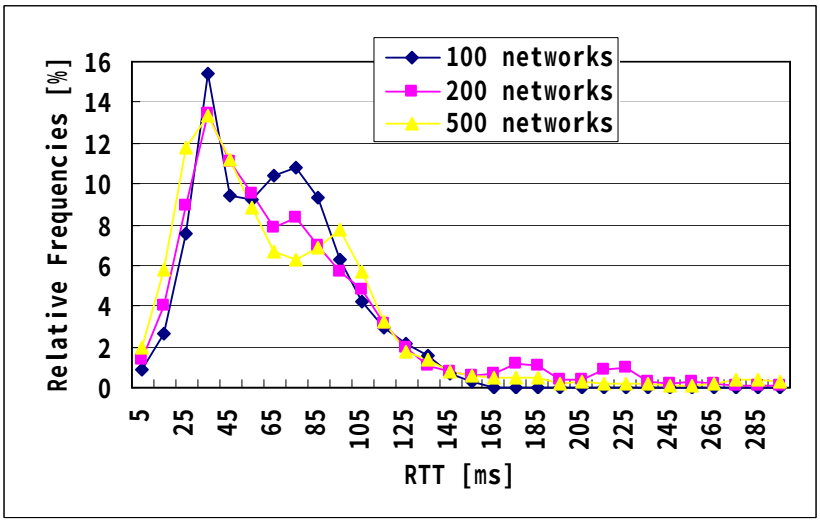

Figure 5-22 Berlin, various sizes of networks

Figure 5-23 compares round-trip-time distributions on various dates. The size of all the subsets is 200. The round-trip-time distribution shows great variance on various dates. To consider the yellow curve and brown curve, which have a time difference of two months, they are very dissimilar, no matter the peak value or shape of the curve.

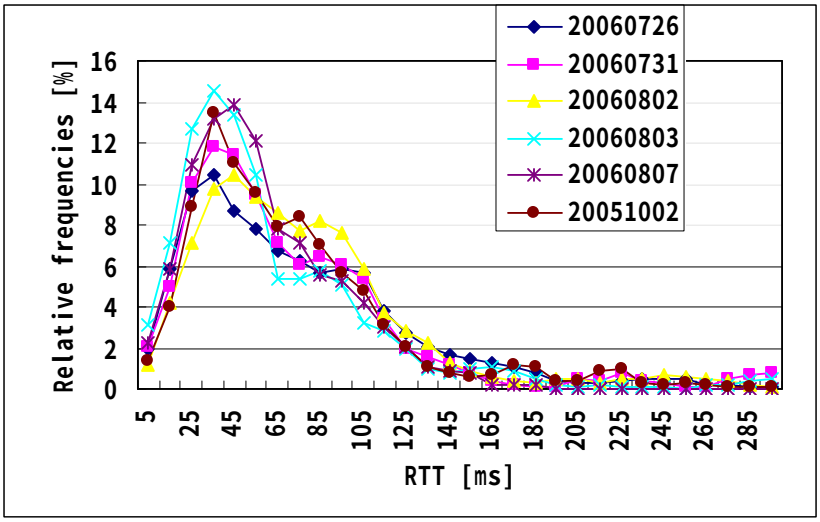

Figure 5-23 Berlin, 200 networks, on various dates

Figure 5-24 compares as well round-trip-time distributions on various dates, but the size of subsets is 500. The great variance is at both peak areas. Compared with 200 networks, the variance of different dates is much less.

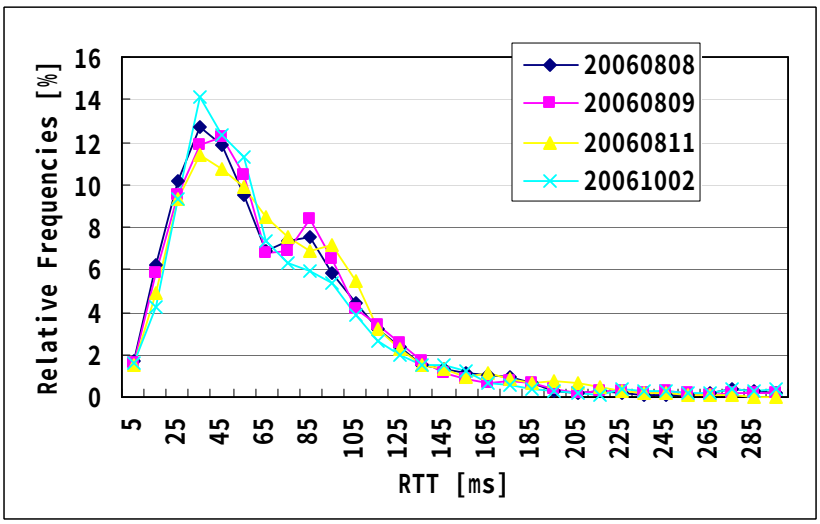

Figure 5-24 Berlin, 500 networks, on various dates

- San Francisco

The third city to be analyzed is San Francisco, which locates in another continent. The round-trip-time distributions spread much wider than that of Hamburg and Berlin. Most curves show three peaks: 1) at RTT 15 ms (delay bin 11-20 ms); 2) at RTT 185 ms (delay bin 181-190 ms); 3) at RTT 355 ms (delay bin 351-360 ms). RTT value of 15 means the transit node locates near to both destinations, which is probably in San Francisco or somewhere near it. However, the value of this peak is rather small compared with the largest peak value,

which indicates most transit nodes of destinations in San Francisco locate far away from San Francisco or even still in Germany. The peak areas are much narrower than that of Hamburg and Berlin. Refer to the results of hop count distributions of the same comparison, round-trip-time distributions show less variance.

Figure 5-25 compares round-trip-time distributions on various dates. The size of all the subsets is 200. Three peaks are quite obvious of these curves although two of them are quite small compared with the large one.

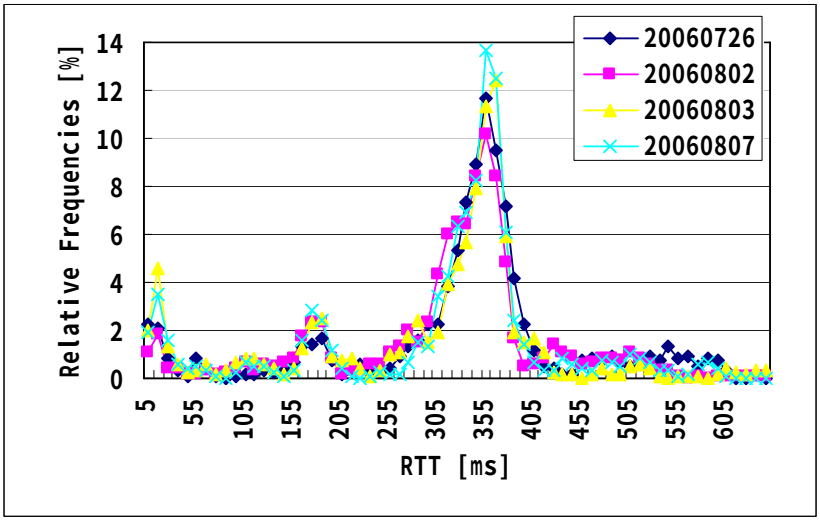

Figure 5-25 San Francisco, 200 networks, on various dates

Figure 5-26 compares as well round-trip-time distributions on various dates, but the size of subsets is 500. These two round-trip-time distributions differ mostly at peak areas. The other parts are quite similar except that the pink curve has a much longer tail.

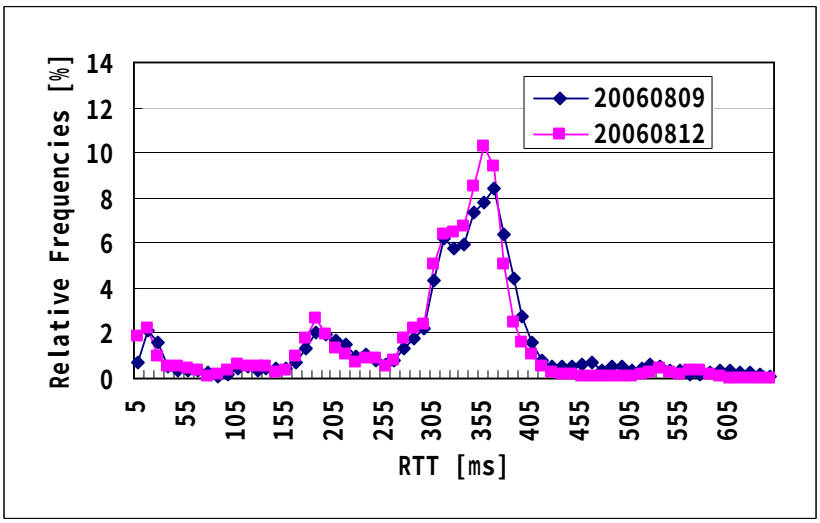

Figure 5-26 San Francisco, 500 networks, on various dates

- Shanghai

The last city to be analyzed is Shanghai. The round-trip-time distributions of Shanghai show great dissimilarity. There is always a peak at RTT 5 ms (delay bin 0-10 ms). This probably means the transit nodes for these pairs are located in Shanghai or somewhere near Shanghai. There are as well two obvious peaks of all the curves: one at RTT 395 ms (delay bin 391-400 ms) and the other at RTT 835 ms (delay bin 831-840 ms). These values means the transit node is located far away from Shanghai or even still in Germany. Therefore, round-trip-time distributions of Shanghai do not represent regional delay distributions because most of transit nodes are located in another country.

Figure 5-27 compares round-trip-time distributions on various dates. The size of all the subsets is 200. These curves are quite dissimilar. The curves spread much wider and the peak is not so obvious except for some extreme cases. The round-trip-time distribution on date 20060930 shows great variance of other dates, that all the peaks are not obvious.

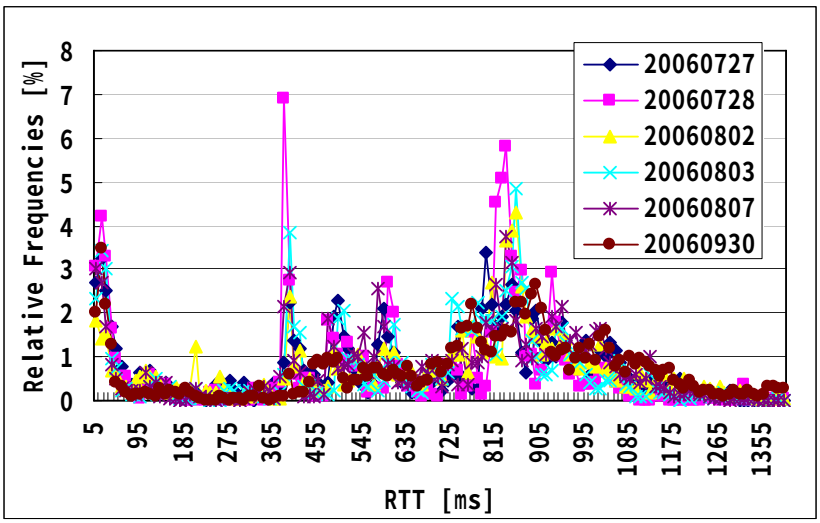

Figure 5-27 Shanghai, 200 networks, on various dates

Figure 5-28 compares as well round-trip-time distributions on various dates, but the size of subsets is 500. These two round-trip-time distributions show less variance than that of 200 networks. Because two round-trip-time distributions were obtained from two close dates, it is also hard to judge whether they will show great variance if more dataset are obtained.

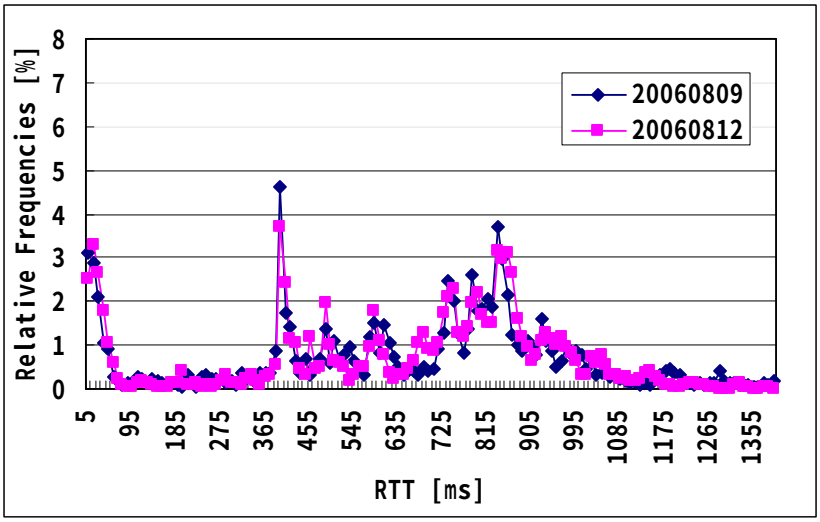

Figure 5-28 Shanghai, 500 networks, on various dates

- Comparison of four cities

Figure 5-29 compares round-trip-time distributions of these four cities. The size of all the subsets is 200. Except for Shanghai, other cites have an obvious peak. The curve of Shanghai in this diagram shows a small peak at RTT 835 ms (delay bin 831-840 ms). The curves of Berlin and Hamburg gain great weights at lower value of RTTs. The curve of San Francisco looks like "right-shifted" of Hamburg and Berlin.

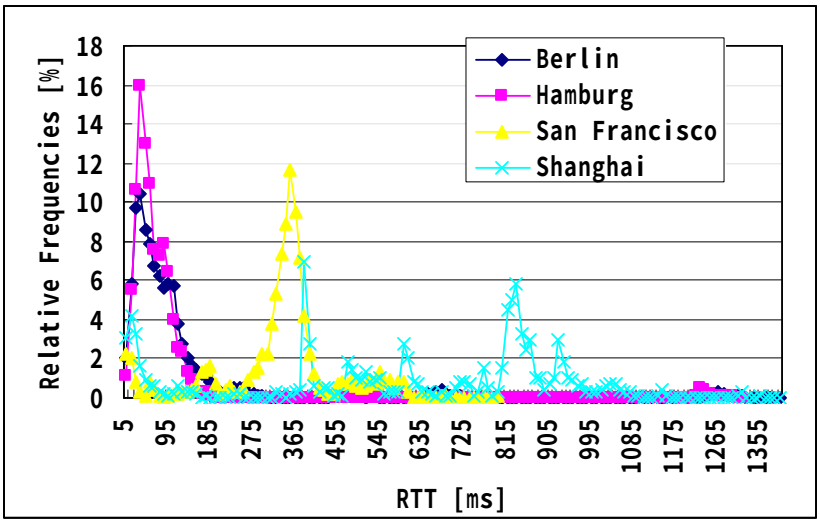

Figure 5-29 four cities, 200 networks

Figure 5-30 compares as well round-trip-time distributions of these four cities, but the size of all the subsets is 500. The situation is the same as 200 networks. The curves of Berlin and Hamburg are quite similar except that the peak value of Hamburg is much larger than that of Berlin. The curve of Shanghai looks much like white noise, which is uniformly distributed.

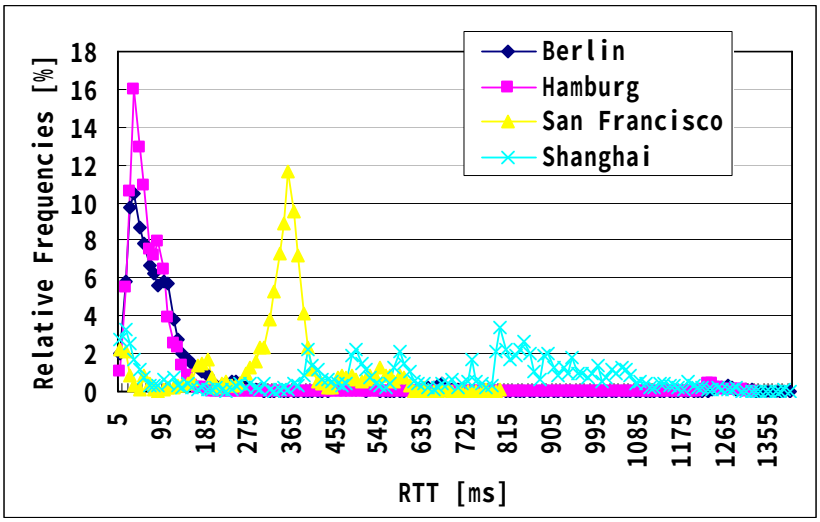

Figure 5-30 four cities, 500 networks

- Random destinations

Figure 5-31 shows round-trip-time distribution of a random-destination dataset of size 200. A random-destination dataset means the 200 IP ranges are chosen randomly from MaxMind GeoIP database. The locations of these IP ranges are unknown. The mean round-trip-time is 278.809 ms, which is much higher than that of Hamburg and Berlin and close to that of San Francisco and much lower than that of Shanghai. The peak value of random-destination dataset is close to that of Shanghai which is about 4%

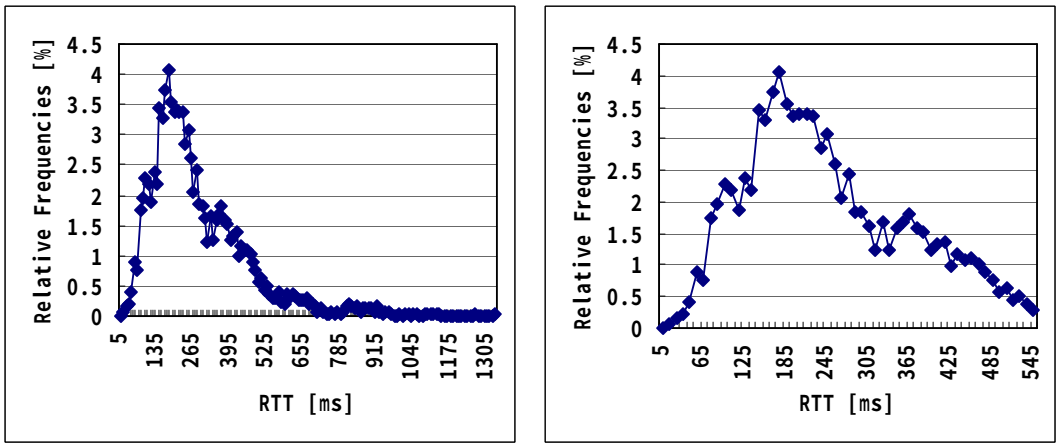

Figure 5-31 random, 200 networks

Figure 5-32 shows the comparison of random-destination dataset to that of four cities. The round-trip-time distribution of random-destination dataset is much widely distributed than that of Berlin, Hamburg and San Francisco. The value of relative frequency at peak is much lower than other cities. The peak is RTT 175 ms (delay bin 171-180 ms) which is

smaller than of San Francisco.

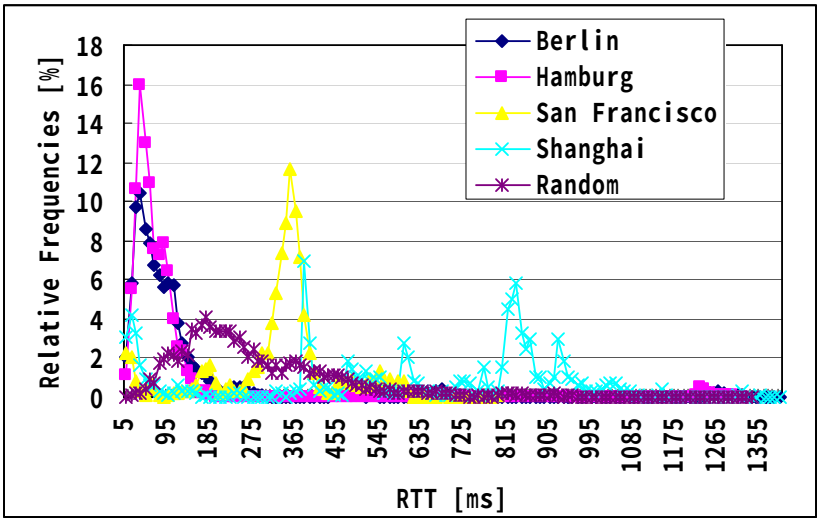

Figure 5-32 random vs. four cities, 200 networks

- Summary

In this section, we compare round-trip-time distributions of four cities. From the results of Hamburg and Berlin, we find time during a day does have impact on round-trip-time distributions, which is different from hop count distributions. The other two parameters, number of networks and various dates, have influences on round-trip-time distributions as well, which is identical to the results obtained from hop count distributions.

Four cities show great variances of round-trip-times distributions except Hamburg and Berlin are similar to each other. Hamburg and Berlin shows great similarity in round-trip-time distributions because they are located in the same country and no global transit providers are involved. The round-trip-time distributions of Shanghai are meaningless in order to analyze regional delay distributions, because most of the transit nodes are not located in Shanghai and the locations of the transit nodes show great diversity. Most likely if both end-hosts are located in Shanghai, the location of transit node will not be outside China. The scan results of a remote destination are quite unpredicted, since there are so many factors which will influence routing paths, e.g. peering point of tier-1 or national ISPs.

All the diagrams of hop count distributions of four cities are shown in Appendix D. The diagrams on the left part represent round-trip-times versus relative frequencies. Most of the distributions have a long tail, which is probably caused by network congestion or some other problems. Therefore, the diagrams of round-trip-time distributions are zoomed in at the critical part of curve which ranges from 0 ms to about twice times of mean round-trip-time of that city.

5.4 Analysis of data from multi-origins

As mentioned in the previous section, the scan results of remote destinations are quite meaningless. A way to obtain better estimation of remote destinations of edge distance is to perform *traceroute* from other locations, which are close to destinations. Other locations are chosen from [24] and are close to destination city, e.g. in the same country. Only the recently scanned data are validated because routing paths might be changed. Hence, validating data scanned long ago has no great meaning.

- Hamburg

Three datasets of Hamburg are validated. They are validated through other two locations[④]. The minimum hop distance of each pair is chosen from the minimum value of these three results. On average, minimum hop distances is 0.1 to 0.2 hop count less than what have been obtained from single-origin. The upper bounds of hop distances are improved, but the difference is not so much (less than one hop count). Since only a bit more than 10%of the destinations which are reachable from other locations, the minimum hop distances are not optimum.

	Mean hop distance of single-origin [hops]	Mean hop distance of minimum value of multi-origins [hops]	Difference [hops]	Accessible ratio of other locations [%]
20060922	12.124	11.892	0.232	11.304
20060926	10.996	10.787	0.209	14.827
20061002	10.004	9.891	0.113	12.809

Table 5-6 comparison of mean hop count of single-origin and multi-origins of Hamburg

Figure 5-33 compares hop count distributions on 20060922 of single-origin and minimum values from multi-origins. Between hop count 9 and hop count 13, the curve of minimum value gains more weights than that of single-origin. Other parts of the curves are quite similar.

[④] These two locations are http://bandit.probe-networks.de/cgi-bin/trace and http://www.traceroutegateway.de/ .

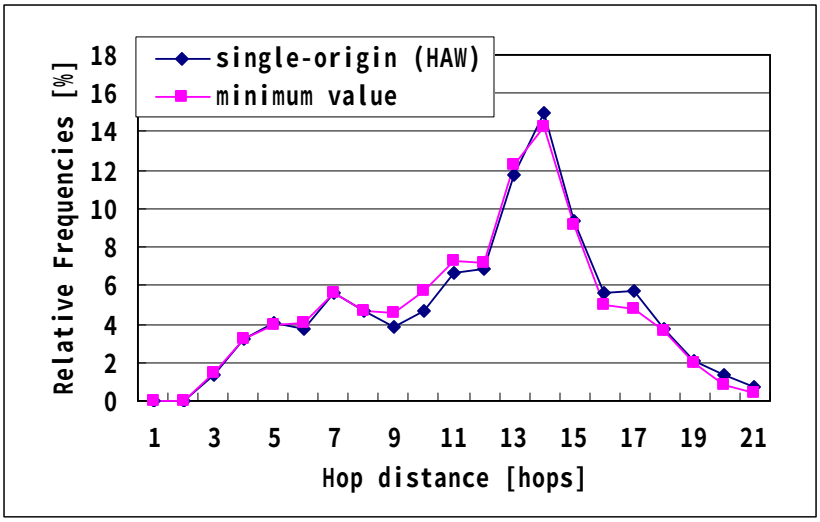

Figure 5-33 single vs. multi-origins, Hamburg, 20060922, 100 networks

Figure 5-34 compares hop count distributions on 20060926 of single-origin and minimum values from multi-origins. Between hop count 6 and 11, the relative percentage of minimum values are higher than that of single-origin.

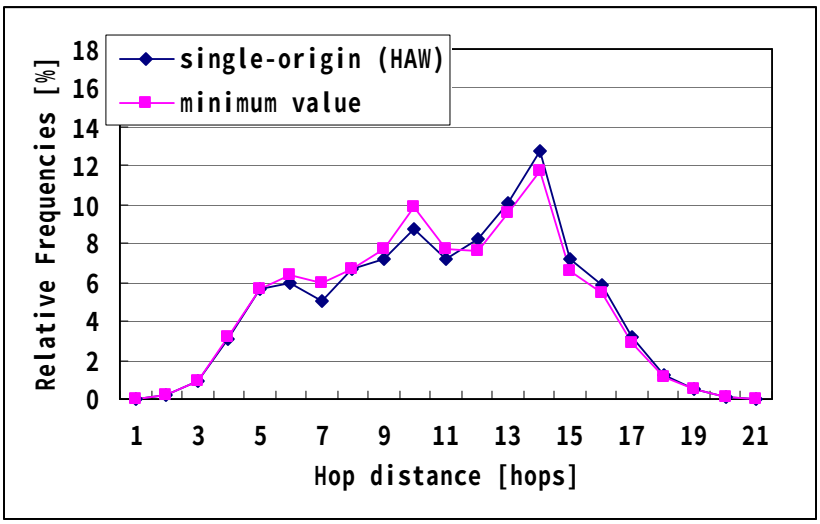

Figure 5-34 single vs. multi-origins, Hamburg, 20060926, 200 networks

Figure 5-35 compares hop count distributions on 20061002 of single-origin and minimum values from multi-origins. The variance of these two distributions is least since the difference of mean hop distance between them is only 0.1 hop count.

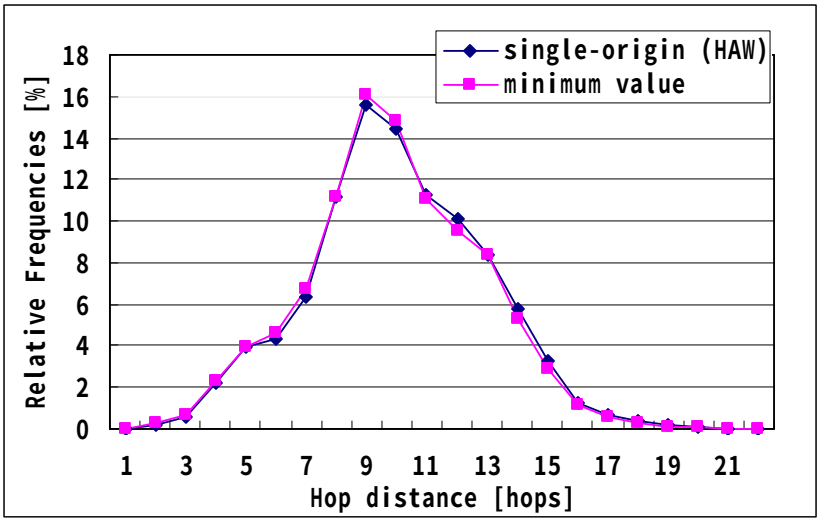

Figure 5-35 single vs. multi-origins, Hamburg, 20061002, 500 networks

- Berlin

One dataset of Berlin is validated. It is validated with the same locations as Hamburg. The difference between mean hop distance of single-origin and that of minimum hop distance from multi-origins is 0.25 hop count. This value is greater than that of Hamburg. The improvement of upper bound hop count distributions is larger than that of Hamburg, although only 7% of the destinations are reachable from other locations. More datasets need to be validated to clarify this result.

	Mean hop distance of single-origin [hops]	Mean hop distance of minimum value of multi-origins [hops]	Difference [hops]	Accessible ratio of other locations [%]
20061004	12.694	12.445	0.250	7.168

Table 5-7 comparison of mean hop count of single-origin and multi-origins of Berlin

Figure 5-36 compares hop count distributions of single-origin and multi-origins of Berlin. The variance of these two hop count distributions is greater than that of Hamburg. Because the accessible ratio of other locations is quite low and most of the values are still taken from test location, the shape of the curve retains.

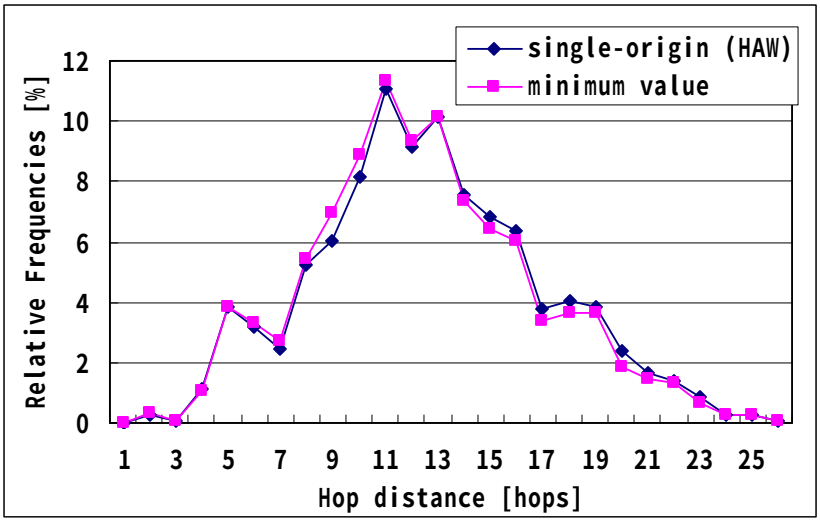

Figure 5-36 single vs. multi-origins, Berlin, 20061004, 100 networks

- San Francisco

One of the dataset of San Francisco is validated through anther origin rather than through websites. Another origin locates in San Diego, which is close to San Francisco. The scanning in both origins was performed on the same subset of San Francisco. The experimental setup in San Diego is the same as that in Hamburg. Table 5-8 lists the measurements from both origins. 90 accessible networks are found from Hamburg and 93 accessible networks are found from San Diego. The ratios of network accessibility do not vary a lot from both locations, which are both below 50%. The mean hop distance of Hamburg is about five hops more than that of San Diego, while the mean RTTs differ quite much of both origins. The mean RTT of origin San Diego is about 73 ms, which almost equals to mean RTT of Hamburg (origin Hamburg). The great dissimilarity of two metrics verifies what was stated by CAIDA people: “We conclude that hop count is not a representative metric for expressing Internet connectivity geographically.”[4]

	Mean hop distance [hops]	Mean RTT [ms]	Accessible networks [#]
Origin: Hamburg	22.219	320.207	90
Origin: San Diego	17.559	73.186	93
Difference	4.660	247.021	3

Table 5-8 comparison of origin in Hamburg and in San Diego

Figure 5-37 compares hop count distributions of different origins. These two curves are totally dissimilar. The hop count distribution of origin San Diego does not have a long tail as origin Hamburg do. The peak area of origin San Diego is much more obvious than that of origin Hamburg.

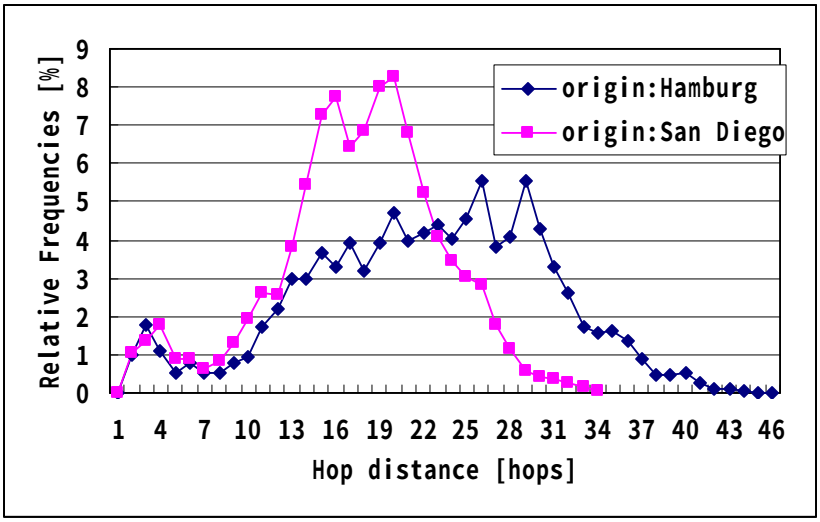

Figure 5-37 comparison of hop count distributions of origin in Hamburg and origin in San Diego, destination: San Francisco, 20061106, 200 networks

Figure 5-38 compares round-trip-time distributions of different origins. The peak value of origin San Diego is at RTT 45 ms (delay bin 41-50 ms), which is rather small compared with that of origin Hamburg. The round-trip-time distribution of origin San Diego is similar to that of Hamburg (both origin and destination are in Hamburg), which indicates more precise data can be obtained if origin and destination are located closely.

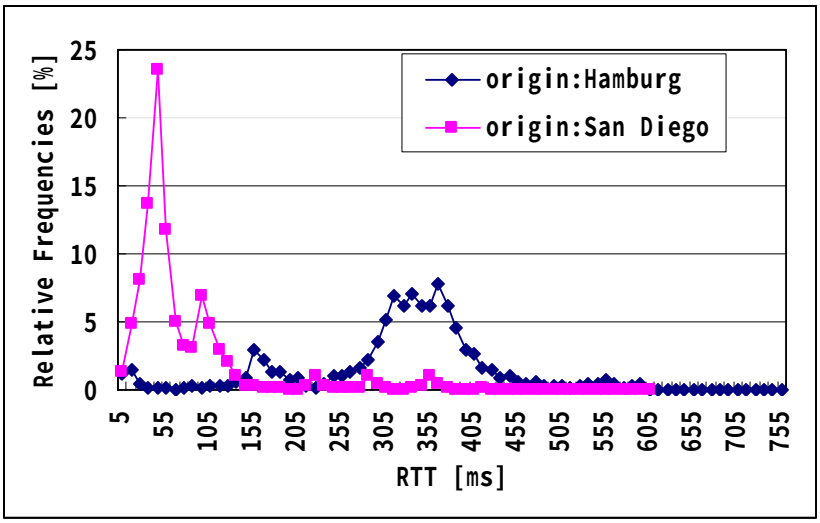

Figure 5-38 comparison of round-trip-time distributions of origin in Hamburg and origin in San Diego, destination: San Francisco, 20061106, 200 networks

- Shanghai

One dataset of Shanghai is validated. It is validated through a location in Hong Kong[⑤] which is much closer to Shanghai than Hamburg, since there is no suitable location in

[⑤] The location in Hong Kong is http://traceroute.hgc.com.hk/cgi-bin/nph-traceroute .

Shanghai has been found. The difference between mean hop distance of single-origin and that of minimum hop distance from multi-origins is 0.549 hop count, which is expected to be larger. Since more than 90% of the networks are not accessible from the location in Hong Kong and more than 90% of minimum values are still taken from scanned results of Hamburg, the minimum values do not offer much better estimation of upper bound edge distance. This value is greater than that of Hamburg and Berlin. It seems improvement for locations faraway is better than the nearby locations.

	Mean hop distance of single-origin [hops]	Mean hop distance of minimum value of multi-origins [hops]	Difference [hops]	Accessible ratio of other locations [%]
20060930	25.396	24.847	0.549	6.867

Table 5-9 comparison of mean hop count of single-origin and multi-origins of Shanghai

Figure 5-39 compares hop count distributions of single-origin and multi-origins of Shanghai. Since the difference of mean hop distance is large, the variance of these two hop count distributions is also large. Between hop count 16 and hop count 22, the curve of minimum value gains much more weights than that of single-origin. At the peak area, the percentage of single-origin is much higher than that of minimum value.

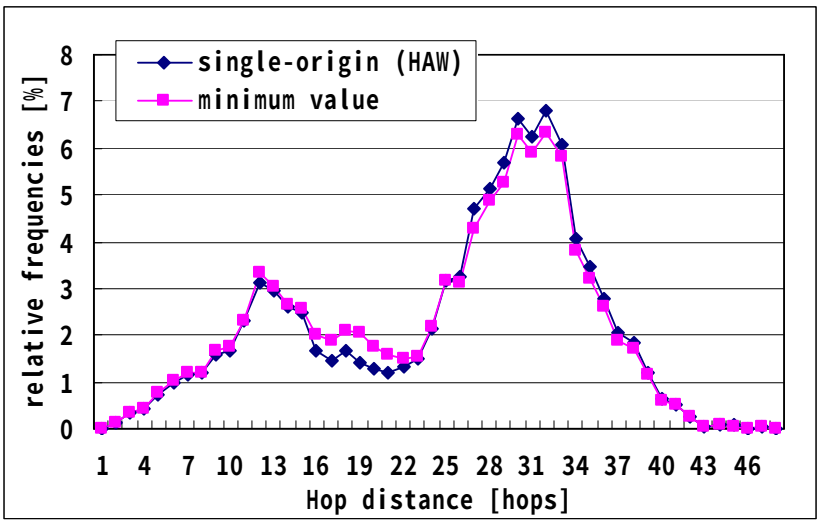

Figure 5-39 single vs. multi-origins, Shanghai, 20060930, 500 networks

Summary

In this section, we compare data from single-origin with minimum values of multi-origins. Although the accessible ratio is not high enough, the upper bound estimations of edge distance have been improved, especially for remote destination such as Shanghai. In order to obtain better estimations, more source-origins should be chosen.

6. Conclusions

This work is motivated by handover issue in a mobile network. In Mobile IPv6, performance of handover is strongly topology dependent due to Binding Update procedures. The network distance of two neighboring access routers influences handover latency and loss of packets. Usually, the previous access router and the next access router are in the geographic vicinity. Therefore, we measure network distance of Internet edges which are located in the same city and study the delay distributions of different cities.

IP clustering is done by a commercial geolocation database. Four cities are chosen as destination dataset and they are Hamburg, Berlin, San Francisco and Shanghai.

We use two basic metrics to measure network distance and they are hop count and round-trip-time. We use the *traceroute* program to infer routing paths and round-trip-time of each intermediate router. For each pair of routing path, the last coinciding router is determined as the transit node for these two end-hosts. An upper bound of edge distance can be estimated via this transit node.

After studying delay distributions of four cities, we found out that time during a day does not influence on hop count distributions while has impact on round-trip-time distributions. Dates affect hop count distributions as well round-trip-time distributions, which infers a change of network topology or routing paths. The size of subset of destinations is also relevant to the change of hop count distributions and round-trip-time distributions. For cities near test location, better estimations can be obtained. The results of faraway destinations are worse than expected and a bit irregular than nearby destinations.

Varying source of *traceroute* probes is one way to validate scanned data from single-source.

We have only selected four cites as our destination datasets, which is far and away not enough to study regional delay distributions. A possible improvement of this work is to use more source-origins and more destination datasets. Another improvement is to use IP source routing if applicable.

7. Acknowledgements

I would like to thank my guiding professor, Prof. Dr. Schmidt, for his patient guidance. I have learned a lot of network knowledge from him. I also thank Mr. Wählisch for his help. Prof. Schmidt and Mr. Wählisch have given me a lot of comments and advice in modifying programs and monitoring network status.

I am grateful to Multimedia Lab of HAW Hamburg for offering experimental infrastructures. I thank Prof. Dr. rer. nat. Renz and Mr. Ißleib for their kind help and aids.

Bibliography

[1] W. Richard Stevens. *TCP/IP Illustrated volume 1: The Protocols*. Addiso-Wesley, 1994.

[2] L. Subramanian, "On Inferring the Geographic Properties of the Internet", Master's thesis, UC Berkeley, 2002.

[3] T. Ronda, N. Bila, "Appono: A Geolocator for the Internet", University of Toronto, 2005. [Online]. Available: http://www.cs.toronto.edu/~ronda/courses/internet_systems/project/appono_report.pdf

[4] B. Huffaker, M. Fomekov, D. Moore, E. Nemeth, and k claffy, "Measurements of the Internet topology in the Asia-Pacific Region", in *INET 2000 Proceedings*. Yokohama, Japan: Internet Society, June 2000. [Online]. Available: Http://www.isoc.org/inet2000/cdproceedings/8e/8e_3.htm

[5] A. Lakhina, J. W. Byers, M. Crovella, and I. Matta, "On the Geographic Location of Internet Resources", IEEE Journal on Selected Areas in Communications, vol. 21, no. 6, pp. 934-948, August 2003

[6] B. Cheswick, H. Burch and S. Branigan, "Mapping and Visualizing the Internet", Proceedings of the 2000 USENIX Annual Technical Conference, June 18-23, 2000, San Diego, California, USA. [Online]. Available: http://db.usenix.org/events/usenix2000/general/full_papers/cheswick/cheswick_html/mapping.html

[7] R. Govindan, H. Tanmunarunkit, "Heuristics for Internet Map Discovery", USC/Information Sciences Institute, CA, USA. [Online]. Available: www.isi.edu/div7/publication_files/heuristics.pdf

[8] T.C. Schmidt and M. Wählisch, "Predictive versus Reactive-Analysis of Handover Performance and Its Implications on IPv6 and Multicast Mobility", *Telecommunication Systems*, 30:1/2/3, 123-142, November 2005.

[9] T.C. Schmidt and M. Wählisch, "Morphing Distribution Trees – On the Evolution of Multicast States under Mobility and an Adaptive Routing Scheme for Mobile SSM Sources", *Telecommunication Systems*, November, 2006.

[10] M. Janic and P.V. Mieghem, "On properties of multicast routing trees", International Journal of Communication Systems. Int, J. Commum. Syst 2006; 19:95-114 Published online 14 November 2005 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/dac.750

[11] N. Spring, R. Mahajan, D. Wetherall and T. Anderson, “Measuring ISP Topologies With Rocketfuel”, IEEE/ACM *Trans. Netw.*, vol. 12, no. 1, pp. 2-16, 2004.

[12] D.B. Johnson, C. Perkins and J. Arkko. “Mobility Support in IPv6”. RFC 3775, IETF, June 2004

[13] R. Koodli, “Fast handovers for mobile IPv6”. RFC 4068, IETF, July 2005.

[14] V. Paxson, “End-to-End Routing Behavior in the Internet”, IEEE/ACM *Trans. Netw.*, vol. 7, no. 3, pp. 277-292, 1999.

[15] The CAIDA web site
http://www.caida.org/home/

[16] GeoIP demo from MaxMind
http://www.maxmind.com/app/locate_ip

[17] Geographic database NetGeo from CAIDA
http://netgeo.caida.org/perl/netgeo.cgi

[18] IP to country database
http://www.ip2country.net/

[19] IP Address Map lookup service from Geobytes, Inc
http://www.geobytes.com/iplocator.htm

[20] WHOIS databases
Europe: http://www.ripe.net/whois/
Asia and Pacific regions: http://www.apnic.net/apnic-bin/whois.pl
America: http://www.arin.net/whois/

[21] IP2Location database from Hexasoft Development Sdn. Bhd
http://www.ip2location.com/

[22] Nmap — Network exploration tool and security / port scanner
http://insecure.org/nmap/

[23] *Traceroute* program
ftp://ftp.ee.lbl.gov/traceroute.tar.gz

[24] *traceroute* from somewhere else
http://www.traceroute.org/

Appendix A. Comparisons of IP2Geo Databases①

IP address		Actual location	www.ip2location.com	www.ip2country.net	www.maxmind.com	www.geobytes.com/iplocator.htm	http://netgeo.caida.org/perl/netgeo.cgi	www.hostip.info
62.206.21.194	country	Germany	GERMANY	Germany	Germany	Germany	NL②	GERMANY
	region	Hamburg			Hamburg	Hamburg	NORTH HOLLAND	
	city	Hamburg			Hamburg	Hamburg	AMSTERDAM	Lichtenstein
	ISP		STUDENTENWERK-HAMBURG-NET	Mediascape communications GmbH	Broadnet AG			
	orginization	studentenwerk			Studentenwerk Hamburg			
134.102.119.36	country	Germany	GERMANY	Germany	Germany	Unable to locate	DE	GERMANY
	region	Bremen			Bremen		BREMEN	
	city	Bremen			Bremen		BREMEN	Rottweil
	ISP	Uni Bremen	UNIVERSITAET BREMEN	Universitaet Bremen	Universitaet Bremen			
	orginization				Universitaet Bremen			

① This comparison is based on free demos of each IP2Geo Database and no official judgments are made to decide which database is good and which database is not so good.
Last update: 2006-05-02
② Red indicates erroneous geographic location

85.178.187.28	country	Germany	GERMANY	Germany	Germany	Germany	US	GERMANY
	region		BERLIN		Berlin	Berlin	CALIFORNIA	
	city	Berlin	BERLIN		Berlin	Berlin	MARINA DEL REY	Siegen
	ISP		HANSENET-A DSL	ALICE DSL	HanseNet Telekommunikation GmbH			
	orginization				ALICE DSL			
87.123.108.232	country	Germany	GERMANY	Germany	Germany	Unable to locate	US	?
	region	Berlin	BERLIN		Berlin		CALIFORNIA	
	city	Berlin	BERLIN		Wilhelmsruh		MARINA DEL REY	?
	ISP		VERSATEL DEUTSCHLAND DYNAMIC POOL	Versatel Nord-Deutschland GmbH	Versatel Nord-Deutschland GmbH			
	orginization				Versatel Deutschland Dynamic Pool			
134.30.15.24	country	Germany	GERMANY	Germany	Germany	Iran	DE	GERMANY
	region	Berlin			Berlin	Tehran	BERLIN	
	city	Berlin			Berlin	Tehran	BERLIN	Bingen
	ISP		HAHN-MEITNER-INSTITUT BERLIN GMBH	Hahn-Meitner-Institut Berlin GmbH	Hahn-Meitner-Institut Berlin GmbH			

	orginization				Hahn-Meitner-Institut Berlin GmbH			
84.176.124.146	country	Germany	GERMANY	Germany	Germany	Germany	US	GERMANY
	region	Hessen	HESSEN		Hessen	Hessen	CALIFORNIA	
	city	Frankfrank	FRANKFURT		Neulsenburg	Frankfurt	MARINA DEL REY	Nuremberg
	ISP	Deutsche Telekom	DEUTSCHE TELEKOM AG	Deutsche Telekom AG	Deutsche Telekom AG			
	orginization				Deutsche Telekom AG			
85.176.9.161	country	Germany	GERMANY	Germany	Germany	Germany	US	GERMANY
	region	Hamburg	HAMBURG		Hamburg	Hamburg	CALIFORNIA	
	city	Hamburg	HAMBURG		Hamburg	Hamburg	MARINA DEL REY	Oberhausen
	ISP		HANSENET-A DSL	ALICE DSL	HanseNet Telekommunikation GmbH			
	orginization				ALICE DSL			
85.182.74.128	country	Germany	GERMANY	Germany	Germany	Germany	US	GERMANY
	region	Hamburg	HAMBURG		Hamburg	Hamburg	CALIFORNIA	
	city	Hamburg	HAMBURG		Hamburg	Hamburg	MARINA DEL REY	Köln
	ISP		HANSENET-A DSL	ALICE DSL	HanseNet Telekommunikation GmbH			
	orginizatio				ALICE DSL			

	n							
83.110.235.217	country	United Arab Emirates	UNITED ARAB EMIRATES	United Arab Emirates	United Arab Emirates	United Arab Emirates	US	UNITED ARAB EMIRATES
	region		DUBAI		Dubai	Dubayy	CALIFORNIA	
	city	Dubai	DUBAI		Dubai	Dubai	MARINA DEL REY	Al ¯Ayn
	ISP		EMIRATES TELECOMMUNICATIONS CORPORATION	PROVIDER Local Internet Registry	Emirates Telecommunications Corporation			
	orginization				Emirates Telecommunications Corporation			
220.238.136.30	country	Australia	AUSTRALIA	Australia	Australia	Unable to locate	AU	AUSTRALIA
	region				Victoria		NEW SOUTH WALES	
	city	Melbourne			Melbourne		MILTON	Hornsby
	ISP		OPTUS INTERNET - RETAIL	OPTUS INTERNET - RETAIL	OPTUS INTERNET - RETAIL			
	orginization				OPTUS INTERNET - RETAIL			
203.88.245.82	country	Australia	AUSTRALIA	Australia	Australia	Australia	AU	AUSTRALI
	region		NEW SOUTH		New South	New South		

			WALES		Wales	Wales		
	city	Sydeney	SYDNEY		Concord	Sydney	MILTON	Sydney
	ISP		TEL.PACIFIC PTY LTD	Rivers Network P/L	Tel.Pacific Pty Ltd			
	orginization				Tel.Pacific Pty Ltd			
218.80.174.144	country	China	CHINA	China	China	China	AU	CHINA
	region		SHANGHAI	Shanghai	Shanghai	Shanghai	NEW SOUTH WALES	
	city	Shanghai	SHANGHAI		Shanghai	Shanghai	MILTON	Hong Kong
	ISP		CHINANET SHANGHAI PROVINCE NETWORK	CHINANET Shanghai province network	Data Communication Division			
	orginization				CHINANET Shanghai province network			
61.177.29.241	country	China	CHINA	China	China	China	AU	CHINA
	region	Jiangsu	JIANGSU	jiangsu	Jiangsu	Jiangsu	NEW SOUTH WALES	
	city	Suzhou	SUZHOU		Suzhou	Suzhou	MILTON	Chongqing
	ISP		CHINANET JIANGSU PROVINCE SUZHOU CITY NETWORK	CHINANET jiangsu province suzhou city network	Data Communication Division			
	orginization				CHINANET jiangsu			

					province suzhou city network			
220.173.137.140	country	China	CHINA	China	**China**	China	AU	**CHINA**
	region	Guangxi	GUANGXI	guangxi	**Guangxi**	Guangxi	NEW SOUTH WALES	
	city	Guilin	**GUANGXI**[③]		**Guilin**	Guilin	MILTON	**Huizhou**
	ISP		CHINANET GUANGXI PROVINCE NETWORK	CHINANET guangxi province network	**Data Communication Division**			
	orginization				**CHINANET Guangxi province network**			
218.17.221.78	country	China	CHINA	China	**China**	China	AU	**CHINA**
	region	Guangdong	GUANGDONG		**Guangdong**	Guangdong	NEW SOUTH WALES	
	city	Shenzhen	SHENZHEN		**Shenzhen**	Shenzhen	MILTON	**Suzhou**
	ISP		SHENZHEN JINYUELONG COMPUTER CO. LTD	ChinaNet Guangdong province network	**CHINANET Guangdong province network**			
	orginization							
222.44.64.65	country	China	CHINA	China	**China**	Unable to locate	US	**CHINA**

[③] Blue indicates inaccurate geographic location

	region				**Beijing**		CALIFORNIA	
	city	Shanghai			**Beijing**		MARINA DEL REY	**Jinghong**
	ISP		CHINA RAILWAY TELECOMMUNICATIONS CENTER	CHINA RAILWAY TELECOMMUNICATIONS CENTER	**CHINA RAILWAY TELECOMMUNICATIONS CENTER**			
	orginization				**CHINA RAILWAY TELECOMMUNICATIONS CENTER**			
82.46.173.186	country	UK	UNITED KINGDOM	Great Britain	**United Kingdom**	Unable to locate	US	**UNITED KINGDOM**
	region		ENGLAND		**Birmingham**		CALIFORNIA	
	city	Birmingham	BIRMINGHAM		**Birmingham**		MARINA DEL REY	**Halesowen**
	ISP		TELEWEST-HSD_2-SMALL_HEATH	PROVIDER Local Registry	**Telewest Broadband**			
	orginization				**Telewest Broadband**			
18.38.5.242	country	USA	UNITED STATES	United States	United States	United States	US	**UNITED STATES**
	region		MASSACHUSETTS	MASSACHUSETTS	Massachusetts	Massachusetts	MASSACHUSETTS	
	city		CAMBRIDGE	Cambridge	Cambridge	Cambridge	CAMBRIDGE	**Cambridge, MA**
	ISP	MIT	MASSACHUS	Massachusetts	Massachusetts		MIT	

			ETTS INSTITUTE OF TECHNOLOGY	Institute of Technology	Institute of Technology			
	orginization				Massachusetts Institute of Technology			
202.120.224.18	country	China	CHINA	China	China	China	CN	CHINA
	region	Shanghai	SHANGHAI		Shanghai	Shanghai	SHANGHAI	
	city	Shanghai	SHANGHAI		Shanghai	Shanghai	SHANGHAI	Shanghai,
	ISP	Fudan Uni	FUDAN UNIVERSITY	China Education and Research Network	China Education and Research Network		FDU-CN	
	orginization				Fudan University			
141.48.144.197	country	Germany	GERMANY	Germany	Germany	Spain	DE	EUROPEAN UNION
	region				Sachsen-Anhalt	Canarias	SAXONY-ANHALT	
	city	Halle			Halle	Santa Cruz de Tenerife	HALLE	Vienna
	ISP	Uni Halle	MARTIN-LUTHER-UNIVERSITAET HALLE-WITTENBERG	Martin-Luther-Universitaet Halle-Wittenberg	Martin-Luther-Universitaet Halle-Wittenberg		MLU-LAN	
	orginization				Martin-Luther-Universitaet Halle-Wittenb			

					erg			
218.78.226.182	country	China	**CHINA**	China	**China**	Unable to locate	AU	**CHINA**
	region	Shanghai		Shanghai	**Shanghai**			
	city	Shanghai			**Shanghai**		MILTON	**Ching-ch'uan**
	ISP		**EAST-CHINA TEACHERS COLLEGE**	CHINANET Shanghai province network	**Data Communication Division**			
	orginization							

Appendix B. Data format

B.1 Individual host-to-host pair

```
Filename:
/home/master01/Scanned_Data/Hamburg/20060922/183107_12_100
Date of scan: 2006/09/22
################################################

Host1: 62.206.100.5
Host2: 217.72.131.66
################################################

traceroute information of 1st host 62.206.100.5:
Hop 1 141.22.64.1 RTT 0.266 ms
Hop 2 141.22.4.121   RTT 0.3 ms
Hop 3 188.1.47.57 RTT 5.008 ms
Hop 4 188.1.144.158  RTT 20.36 ms
Hop 5 188.1.18.109   RTT 21.329 ms
Hop 6  RTT  ms
Hop 7 188.1.144.54   RTT 20.845 ms
Hop 8 80.81.192.208  RTT 20.467 ms
Hop 9 62.206.100.5   RTT 43.829 ms
################################################

traceroute information of 2nd host 217.72.131.66:
Hop 1 141.22.64.1 RTT 0.285 ms
Hop 2 141.22.4.121   RTT 0.649 ms
Hop 3 188.1.47.57 RTT 3.579 ms
Hop 4 188.1.144.158  RTT 3.95 ms
Hop 5 213.248.103.97 RTT 4.027 ms
Hop 6 80.239.144.66  RTT 3.942 ms
Hop 7 217.72.128.3   RTT 4.275 ms
Hop 8  RTT  ms
Hop 9 217.72.131.66  RTT 4.689 ms
################################################

last intersection for two host is : 188.1.144.158

################################################
```

```
hop distance from source to host1: 9
hop distance from source to host2: 9

hop distance from last intersect to host 1: 5
hop distance from last intersect to host 2: 5
hop distance between two hosts: 10
################################################

rtt from source to host1: 43.829 ms
rtt from source to host2: 4.689 ms

rtt from last intersect to host1: 23.469 ms
rtt from last intersect to host2: 0.739 ms
rtt difference between two hosts: 24.208 ms
```

B.2 Overall matrix

	network1	network2	network3
network1	N/A	hop distance between nw1 & nw2	hop distance between nw1 & nw3
network2	RTT diff between nw1 & nw2	N/A	hop distance between nw2 & nw3
network3	RTT diff between nw1 & nw3	RTT diff between nw2 & nw3	N/A

B.3 Comparison database

nw1	nw2	min	HAW	traceroutegateway
1	2	12	12	N/A
1	3	16	17	16
1	4	14	14	N/A
2	3	15	15	18
2	4	17	17	18

Appendix C. Hop count distributions

Hamburg

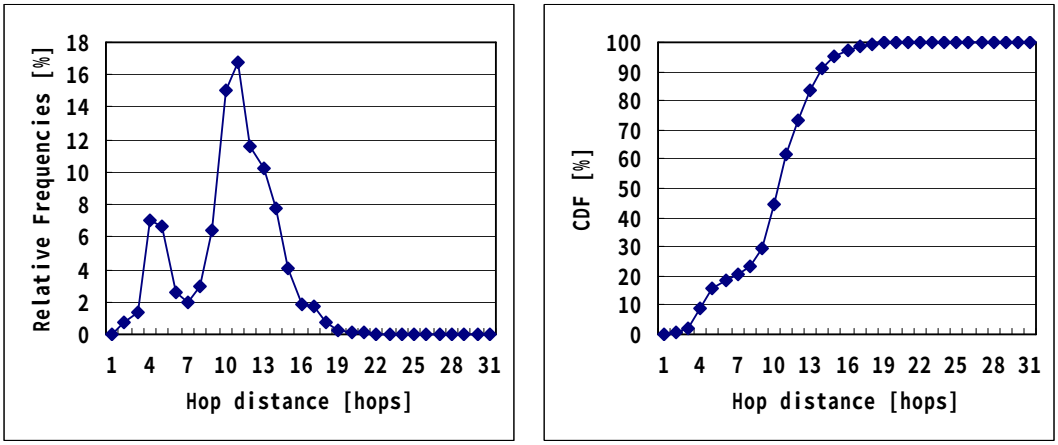

Fig. 1 Hamburg, 20060726_114436, 200 networks

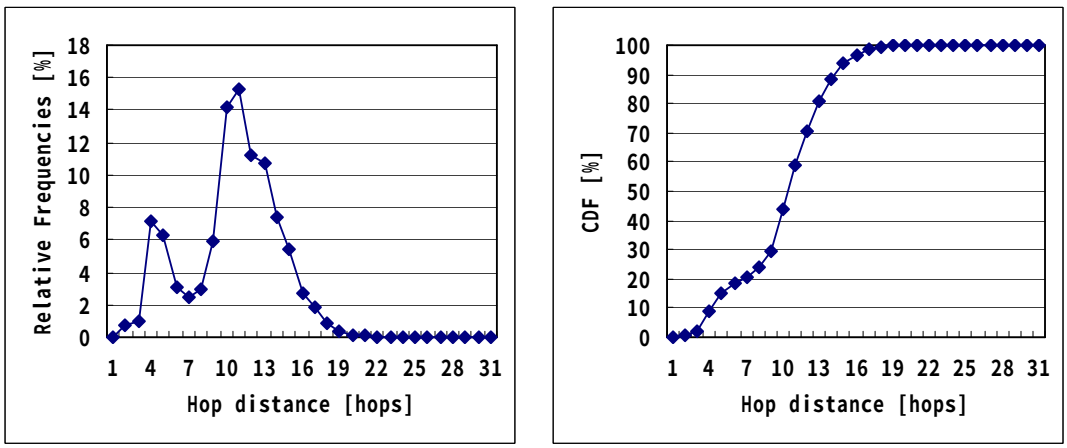

Fig. 2 Hamburg, 20060726_205735, 200 networks

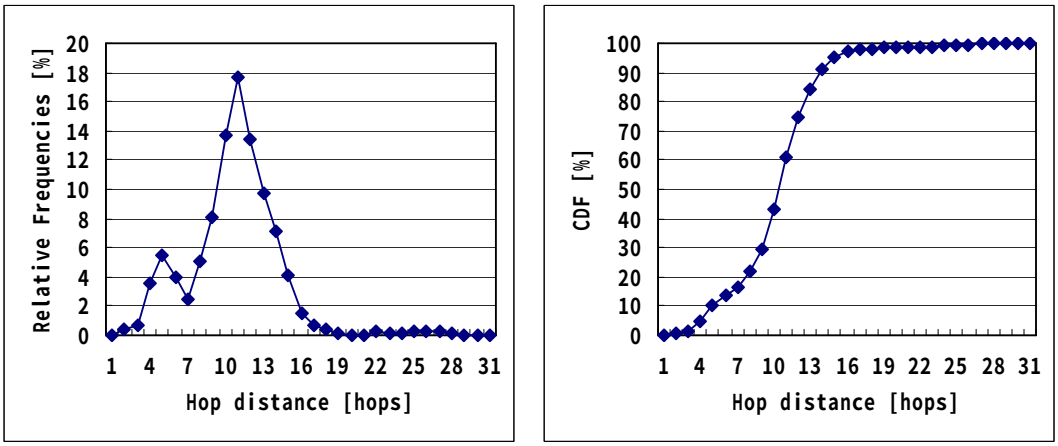

Fig. 3 Hamburg, 20060728_202345, 200 networks

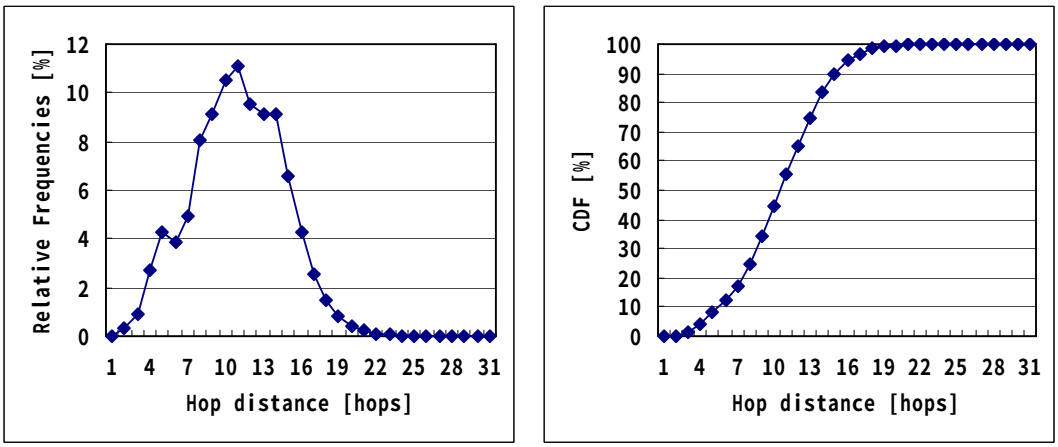

Fig. 4 Hamburg, 20060802_142704, 200 networks

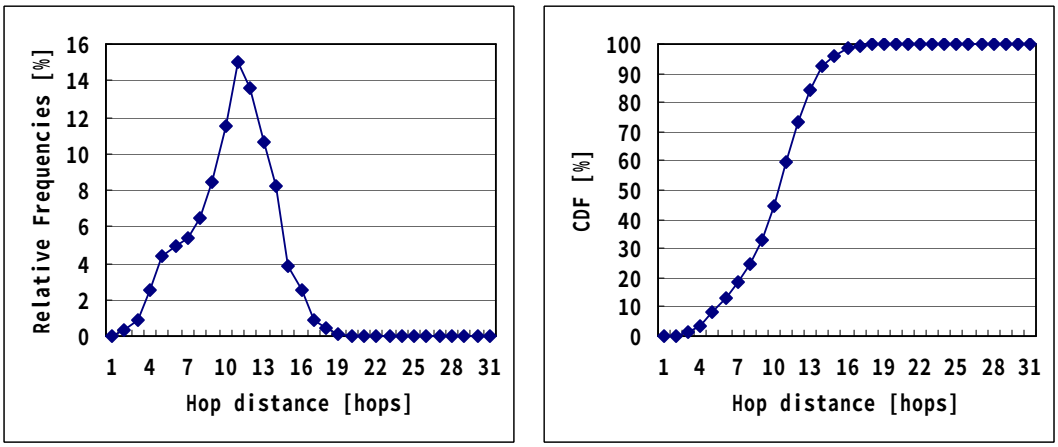

Fig. 5 Hamburg, 20060803_155825, 200 networks

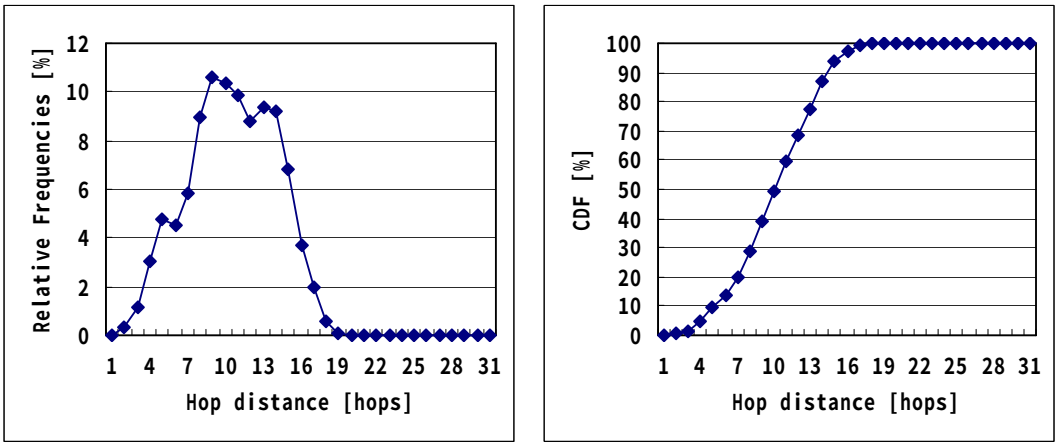

Fig. 6 Hamburg, 20060807_144843, 200 networks

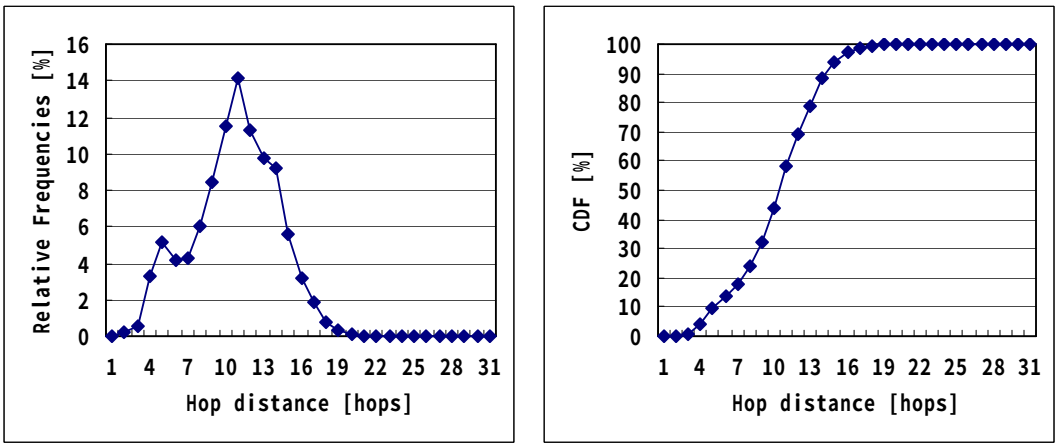

Fig. 7 Hamburg, 20060808_202012, 500 networks

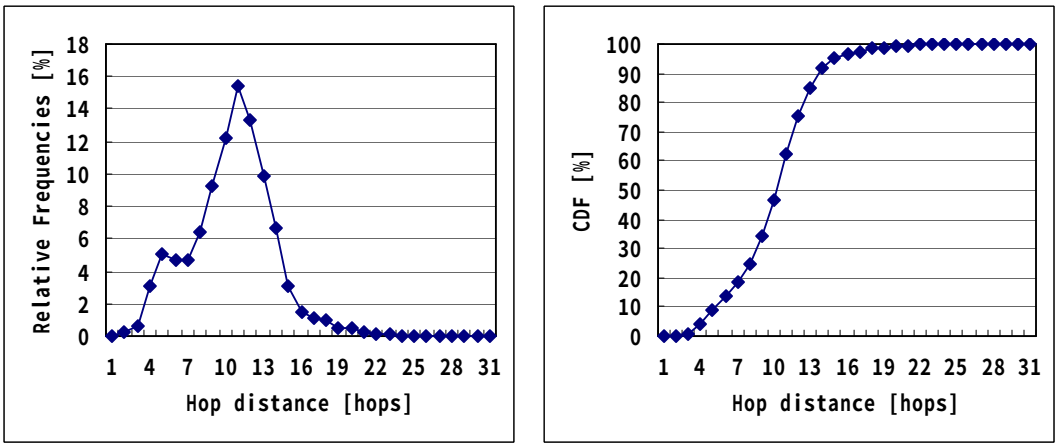

Fig. 8 Hamburg, 20060810_205630, 500 networks

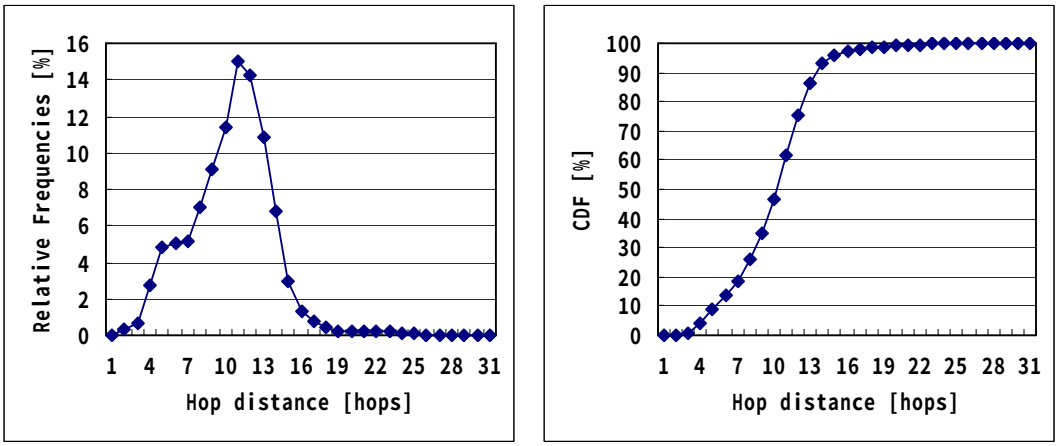

Fig. 9 Hamburg, 20060811_200506, 500 networks

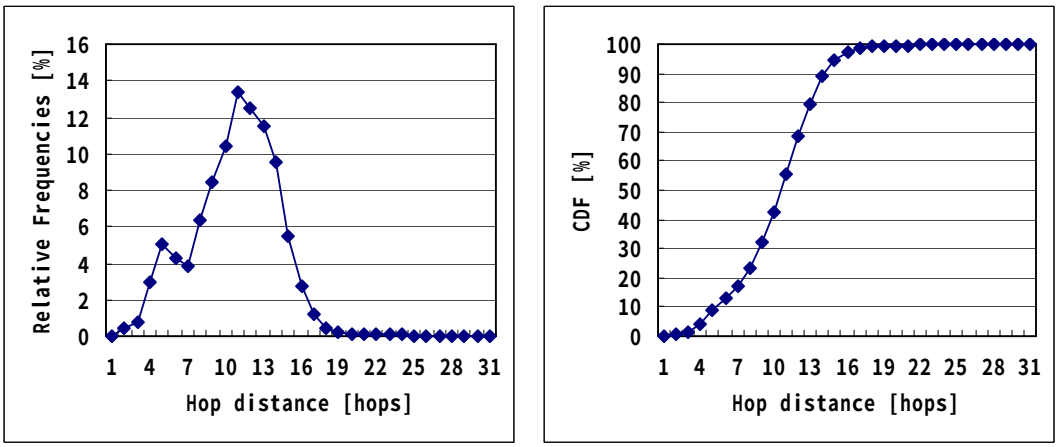

Fig. 10 Hamburg, 20060815_192410, 500 networks

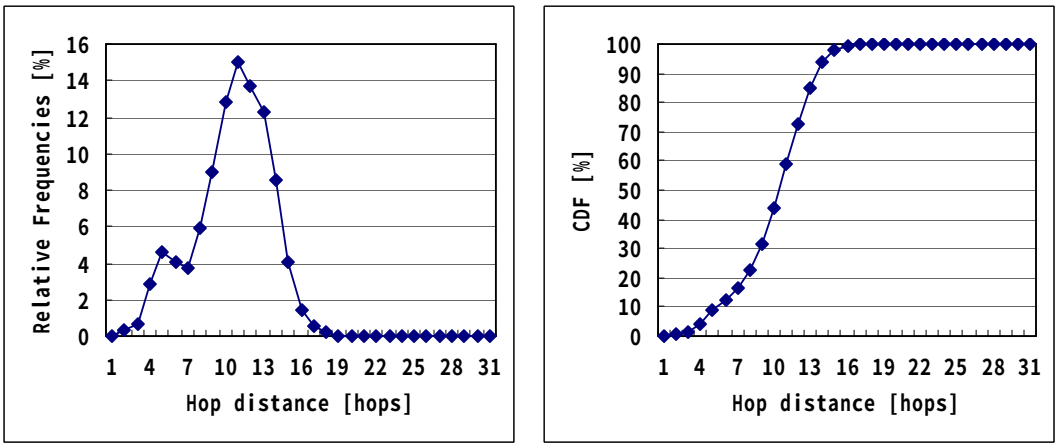

Fig. 11 Hamburg, 20060821_192300, 500 networks

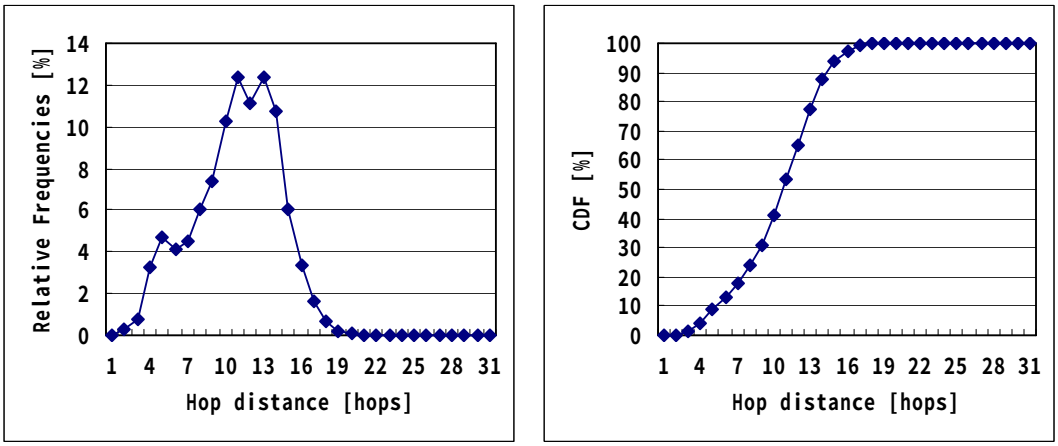

Fig. 12 Hamburg, 20060901_201950, 1000 networks

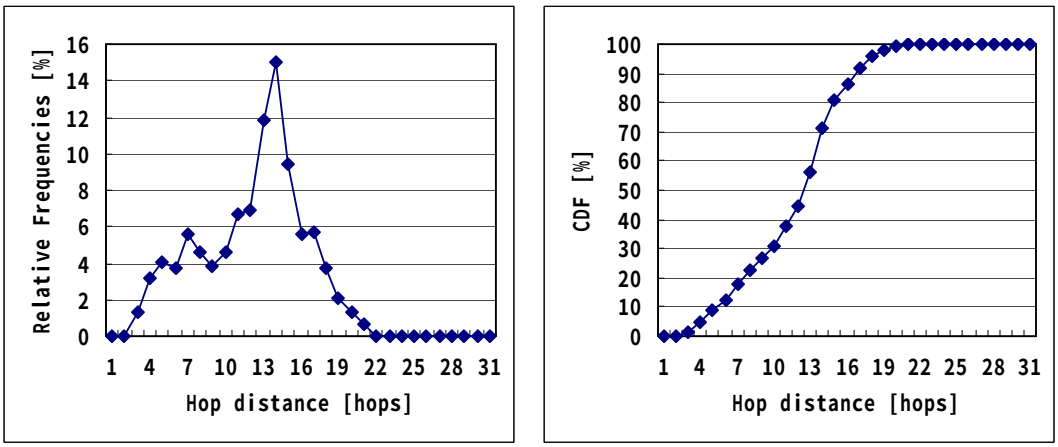

Fig. 13 Hamburg, 20060922_183115, 100 networks

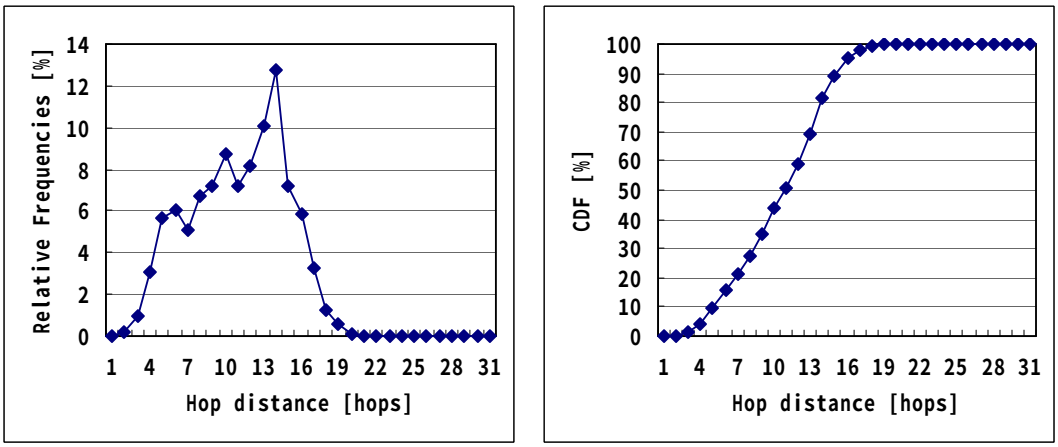

Fig. 14 Hamburg, 20060926_230110, 200 networks

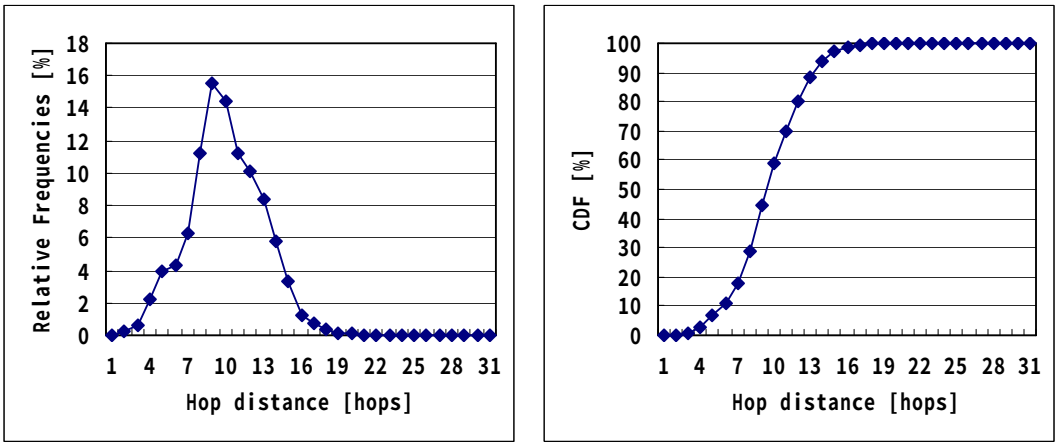

Fig. 15 Hamburg, 20061002_183049, 500 networks

Berlin

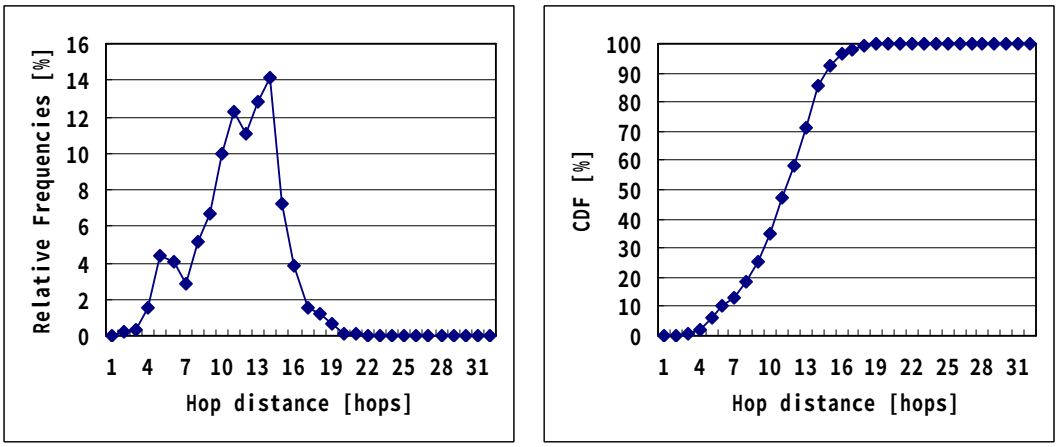

Fig. 16 Berlin, 20060726_024053, 200 networks

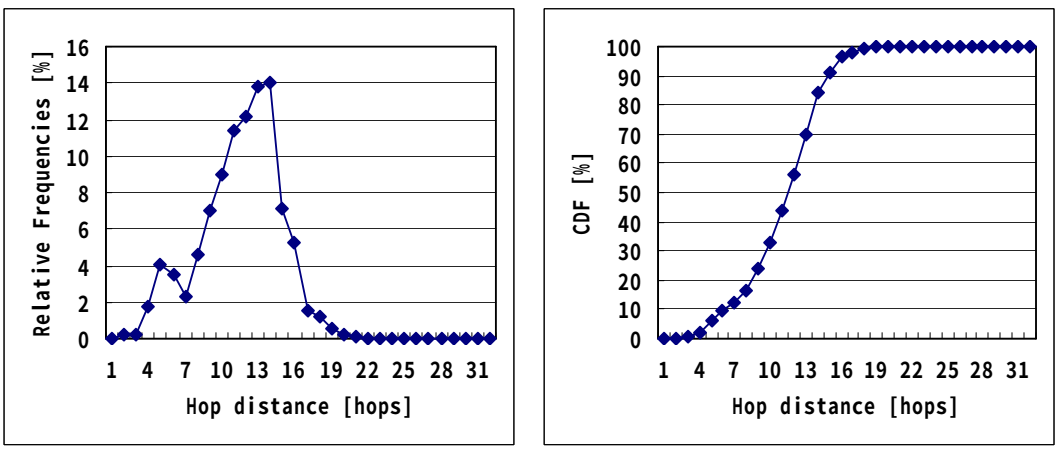

Fig. 17 Berlin, 20060726_171319, 200 networks

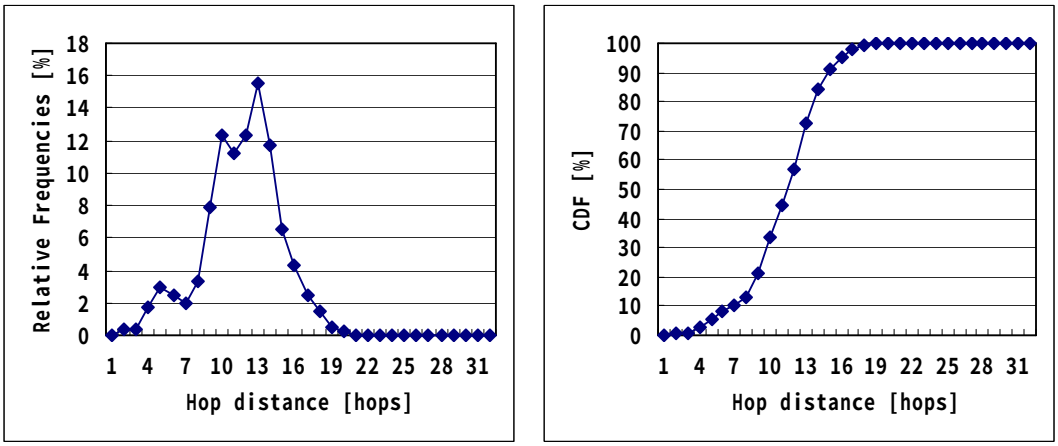

Fig. 18 Berlin, 20060731_042056, 200 networks

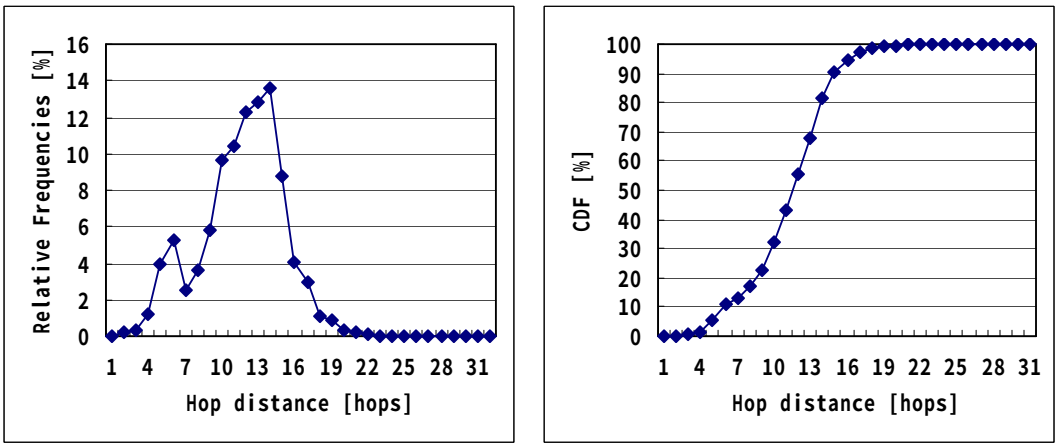

Fig. 19 Berlin, 20060802_150317, 200 networks

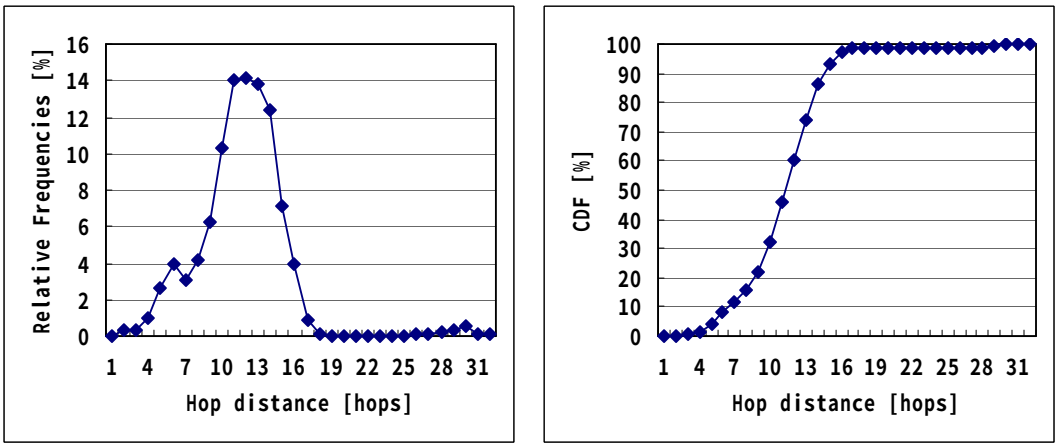

Fig. 20 Berlin, 20060803_152848, 200 networks

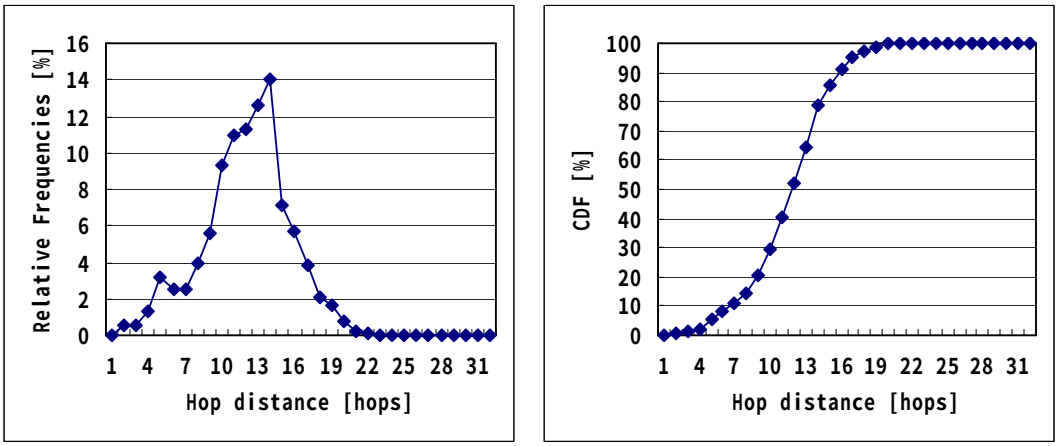

Fig. 21 Berlin, 20060807_142738, 200 networks

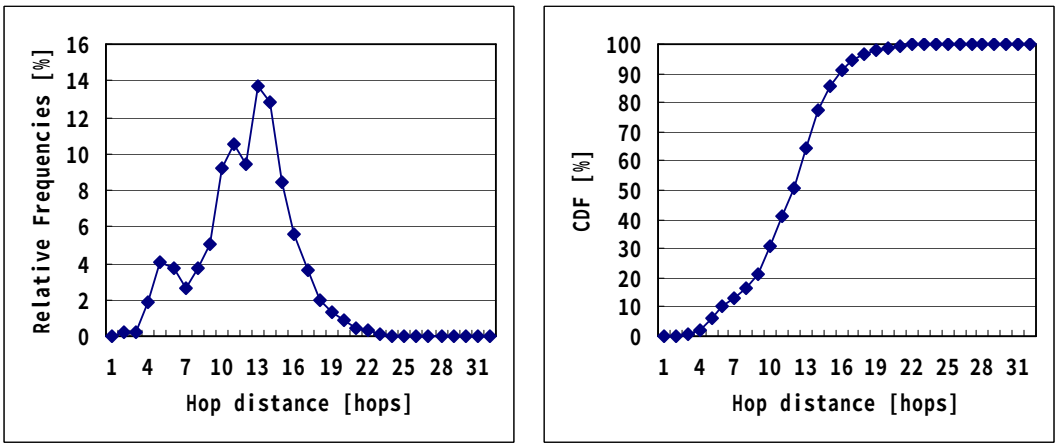

Fig. 22 Berlin, 20060808_065115, 500 networks

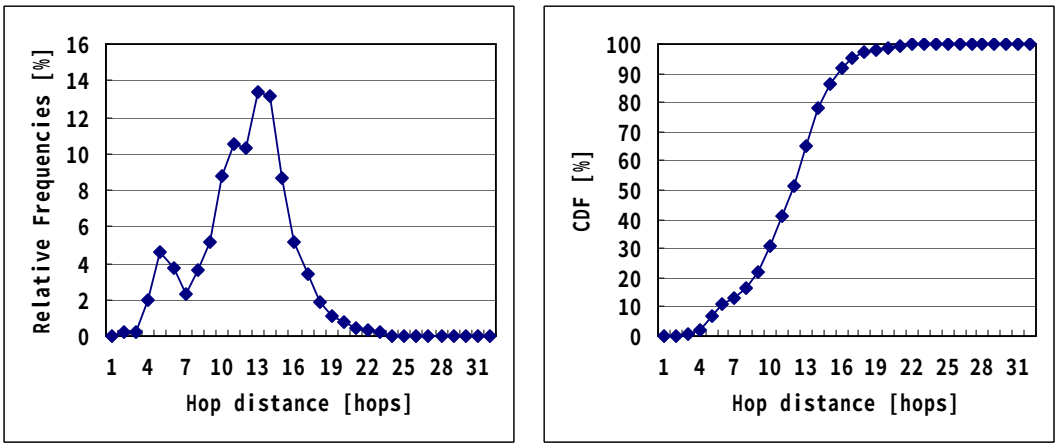

Fig. 23 Berlin, 20060809_041654, 500 networks

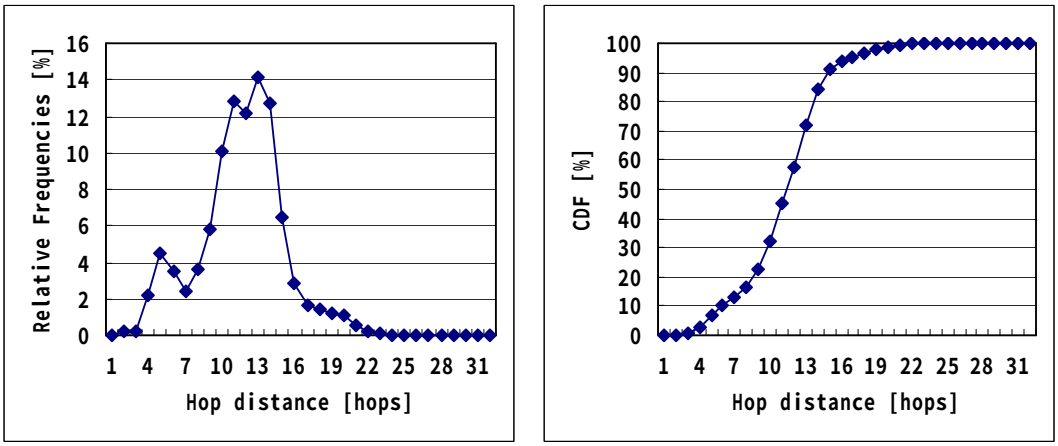

Fig. 24 Berlin, 20060811_042452, 500 networks

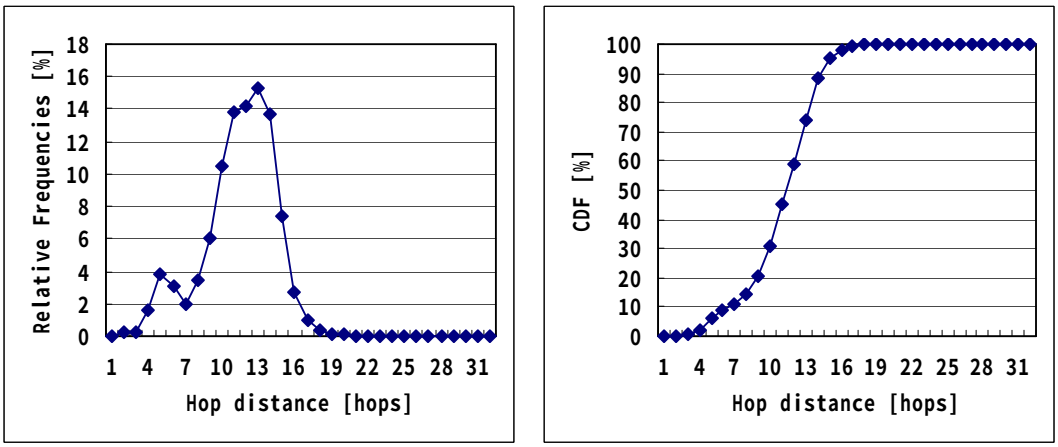

Fig. 25 Berlin, 20060811_222407, 500 networks

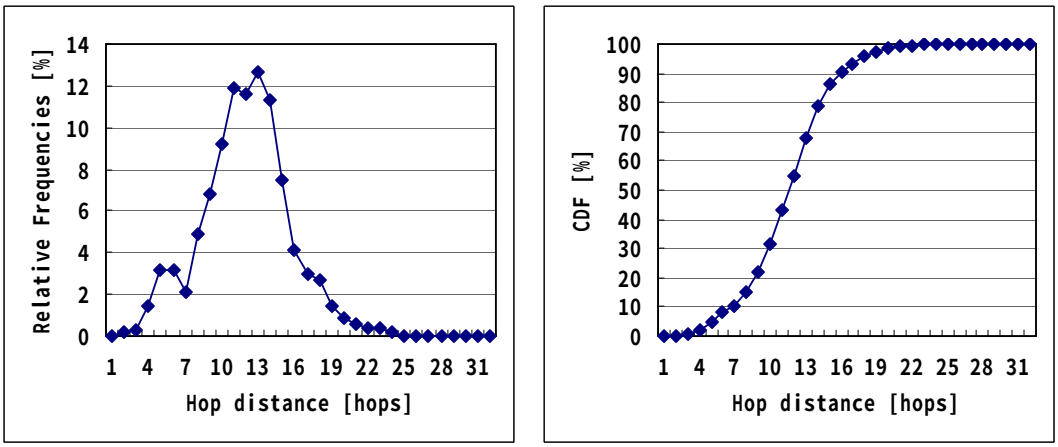

Fig. 26 Berlin, 20061002_163723, 200 networks

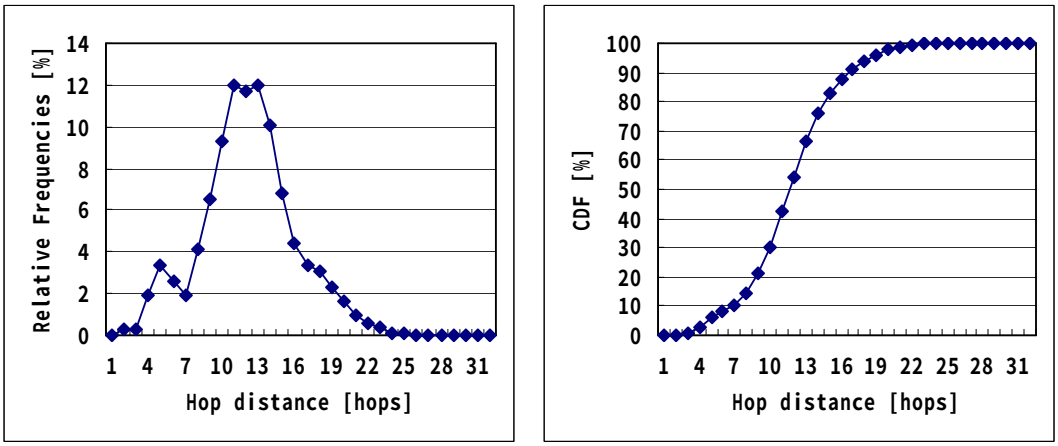

Fig. 27 Berlin, 20061002_232639, 500 networks

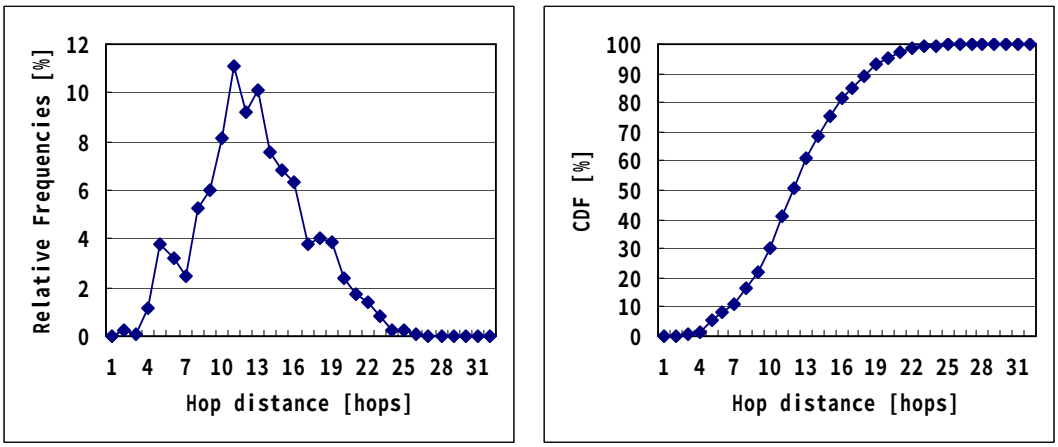

Fig. 28 Berlin, 20061004_161750, 100 networks

San Francisco

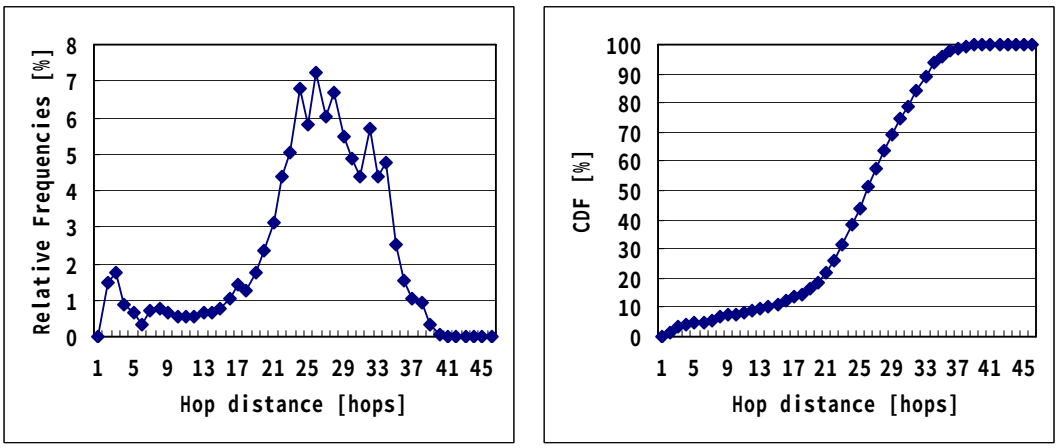

Fig. 29 San Francisco, 20060726_125729, 200 networks

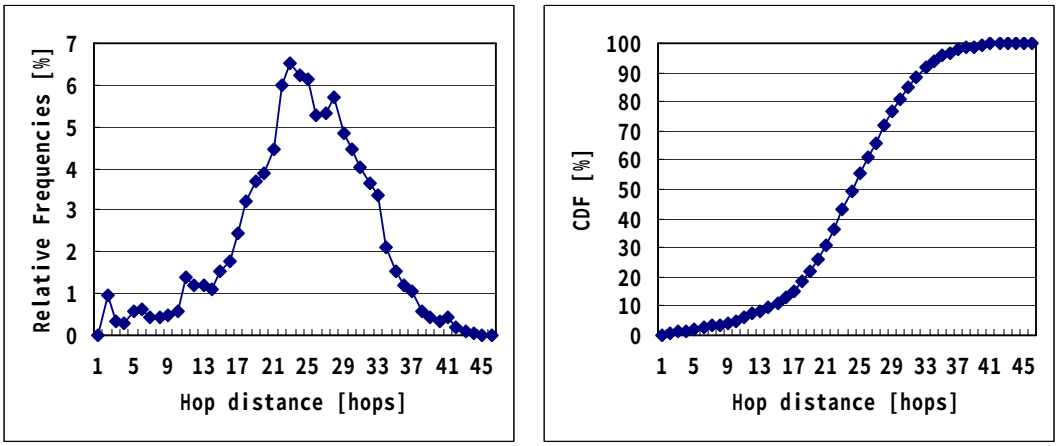

Fig. 30 San Francisco, 20060802_180923, 200 networks

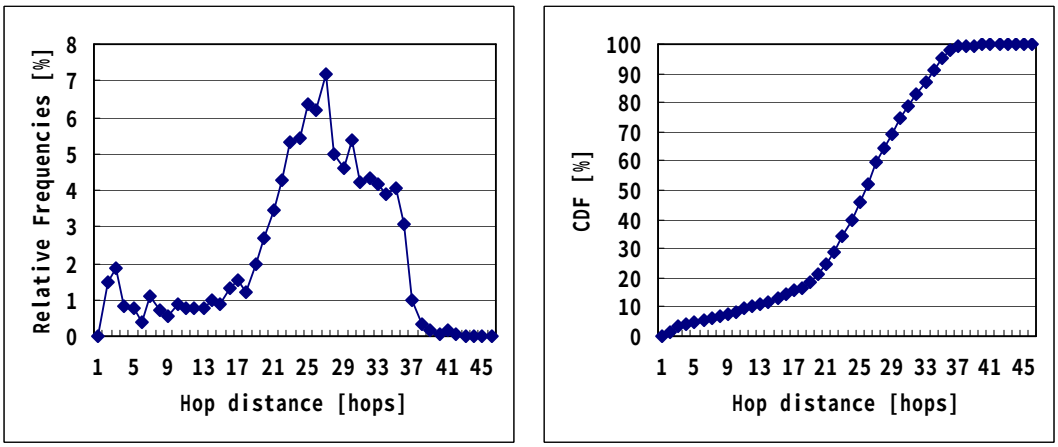

Fig. 31 San Francisco, 20060803_195448, 200 networks

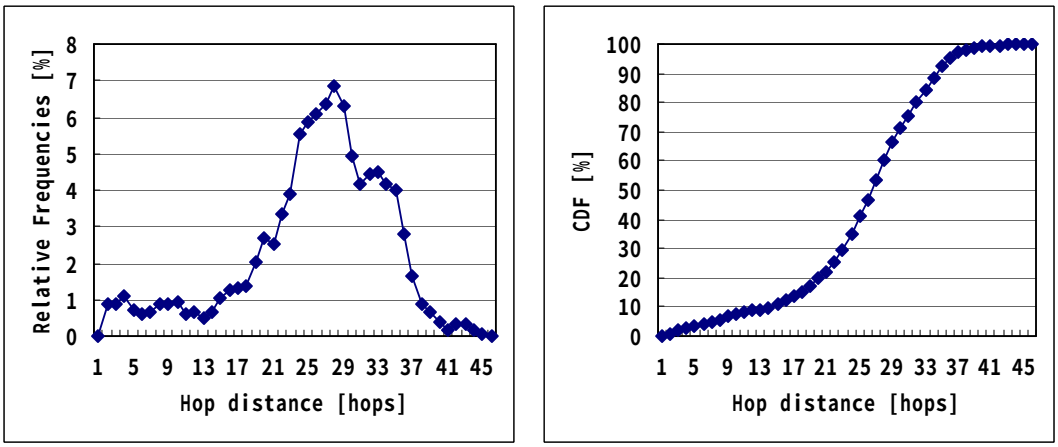

Fig. 32 San Francisco, 20060807_161808, 200 networks

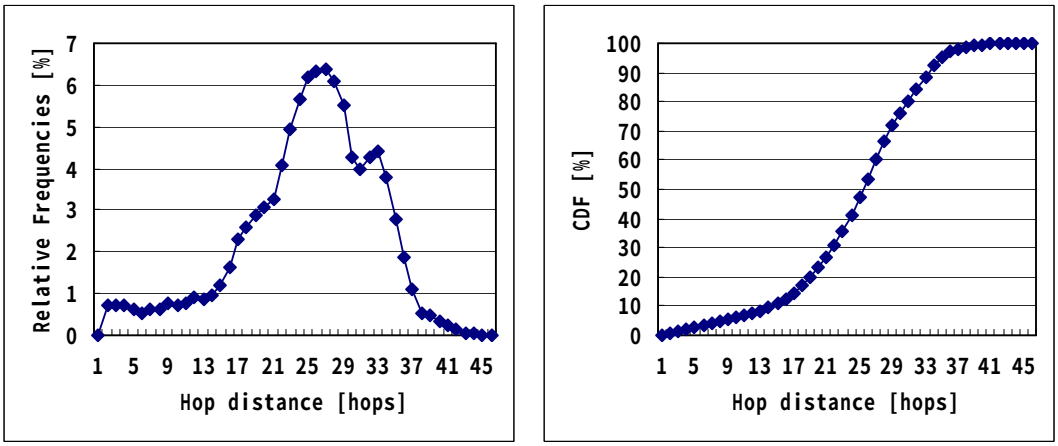

Fig. 33 San Francisco, 20060809_030747, 500 networks

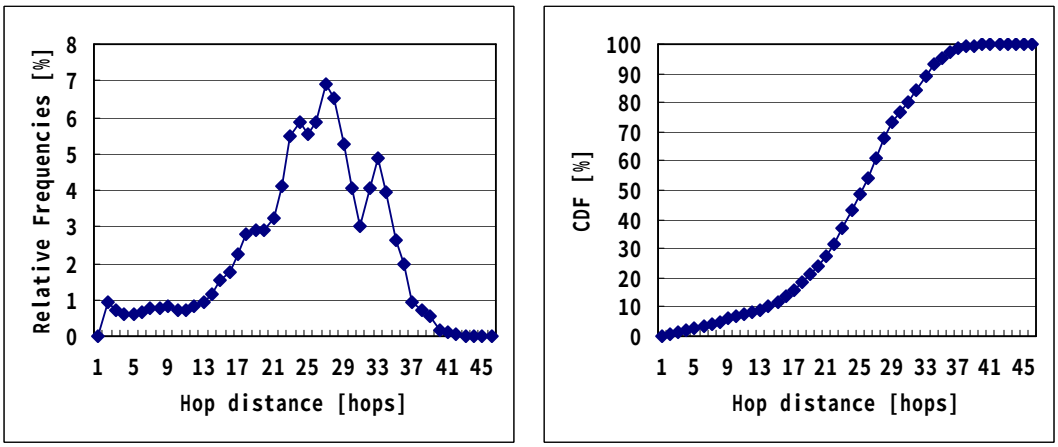

Fig. 34 San Francisco, 20060812_003223, 500 networks

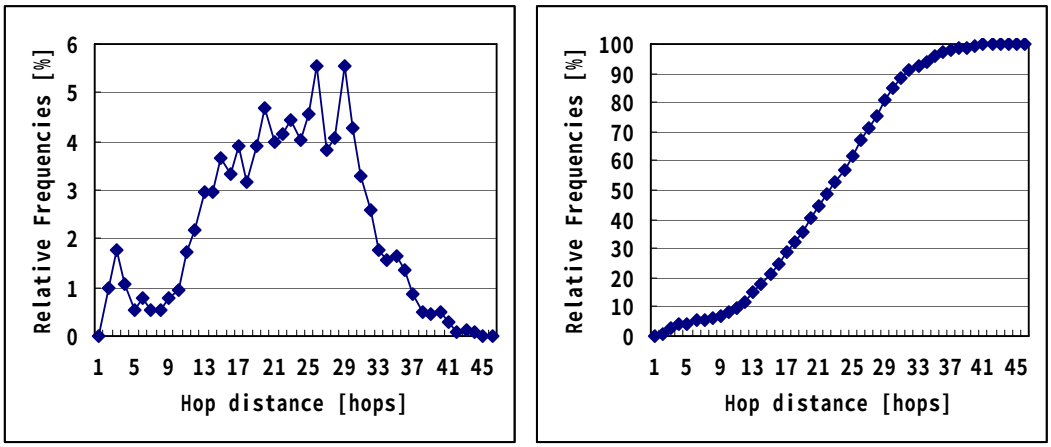

Fig. 35 San Francisco, 20061107_202944, 200 networks

Shanghai

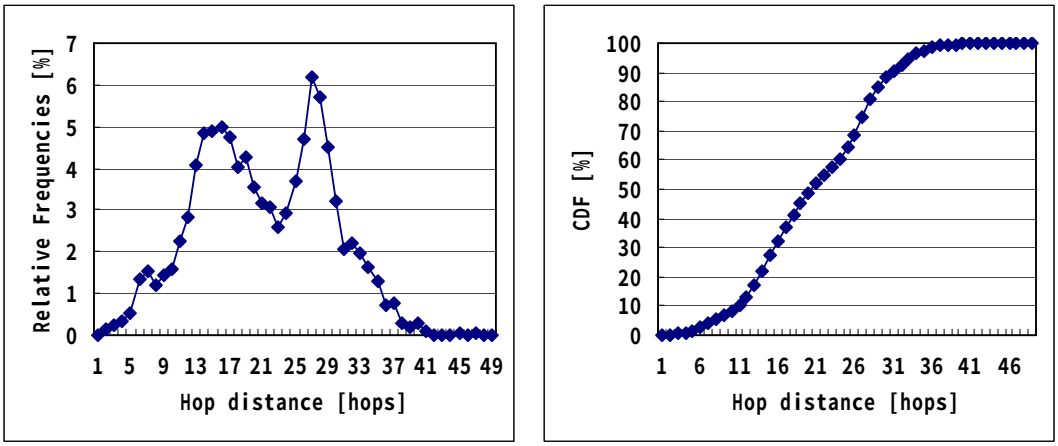

Fig. 36 Shanghai, 20060727_191127, 200 networks

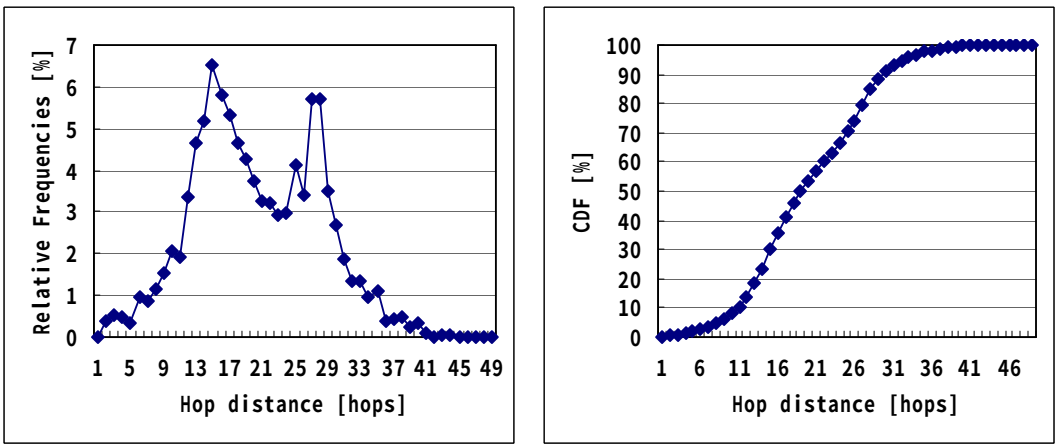

Fig. 37 Shanghai, 20060728_232308, 200 networks

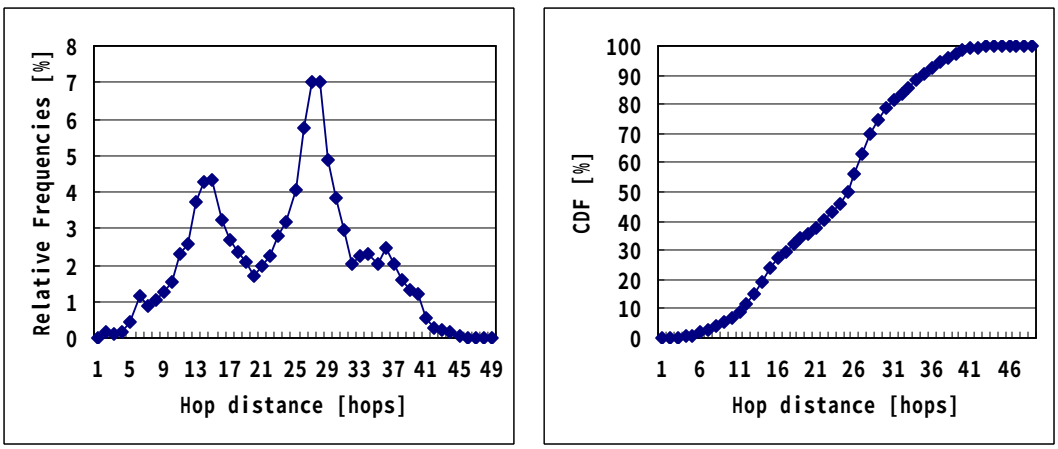

Fig. 38 Shanghai, 20060802_174350, 200 networks

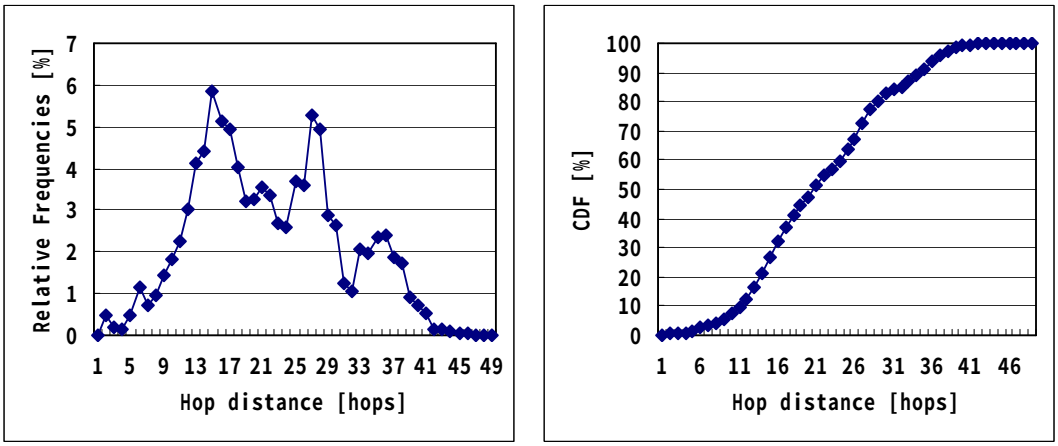

Fig. 39 Shanghai, 20060803_210942, 200 networks

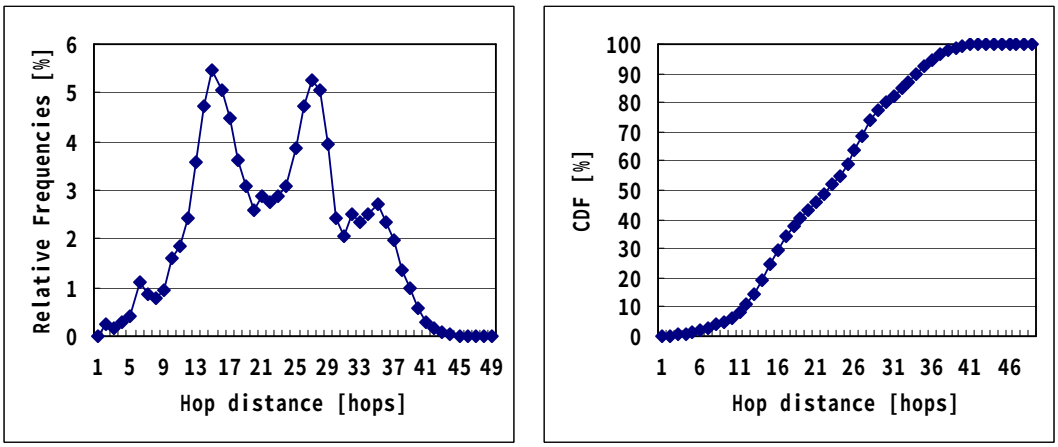

Fig. 40 Shanghai, 20060807_211101, 200 networks

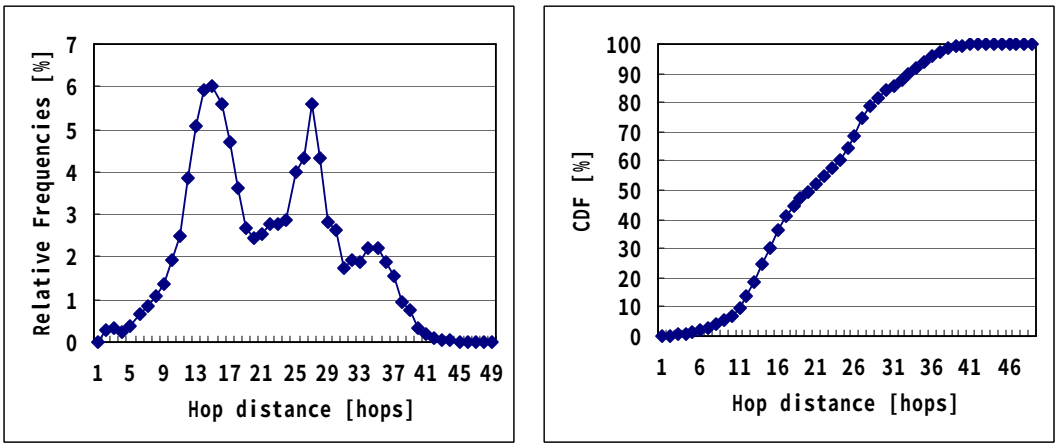

Fig. 41 Shanghai, 20060809_053425, 500 networks

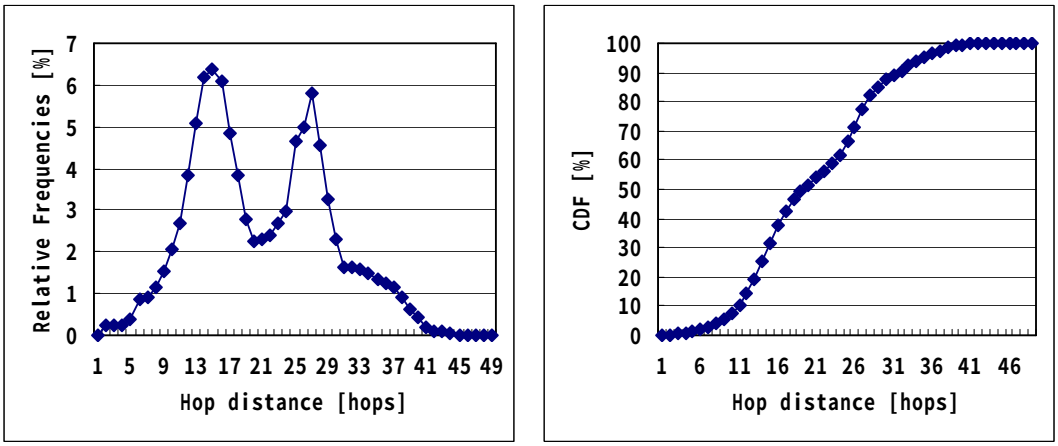

Fig. 42 Shanghai, 20060812_084532, 500 networks

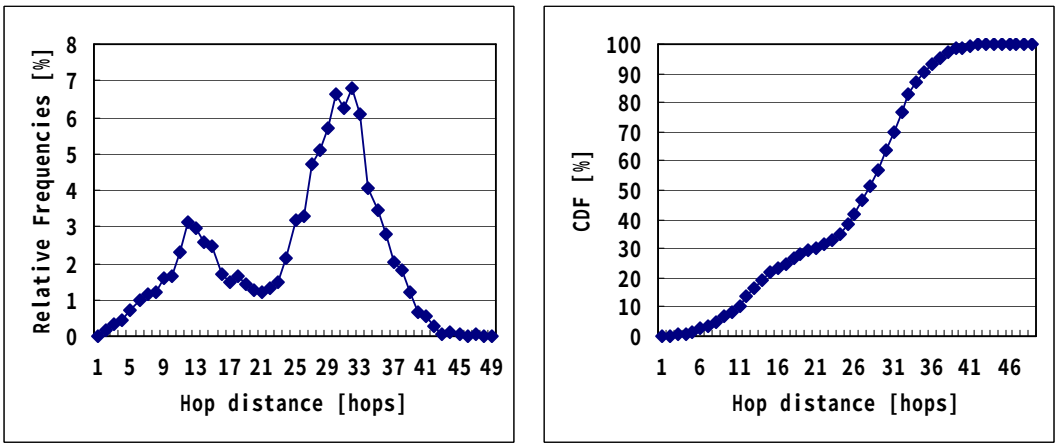

Fig. 43 Shanghai, 20060930_011959, 200 networks

Appendix D. Round-trip-time distributions

Hamburg

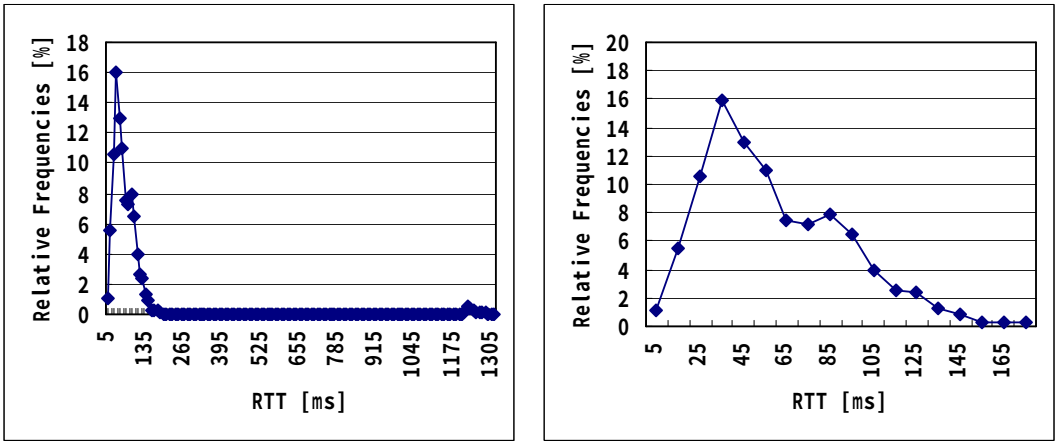

Fig. 44 Hamburg, 20060726_114436, 200 networks

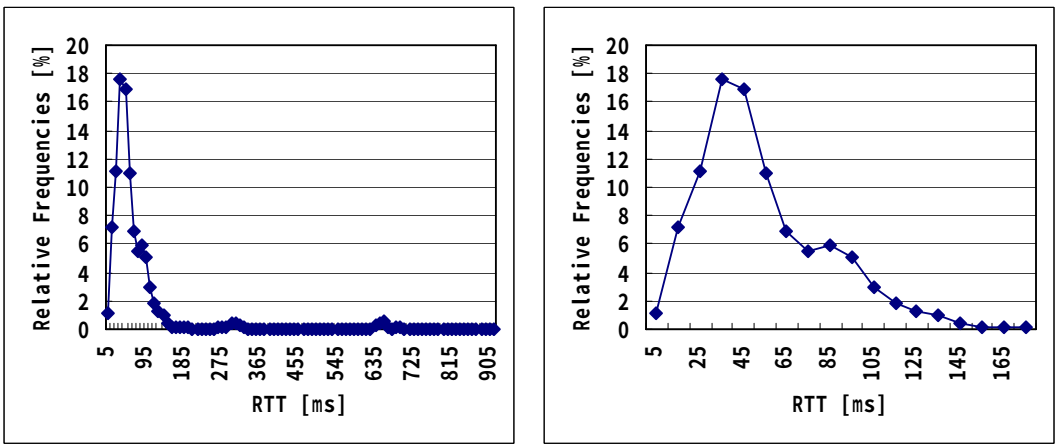

Fig. 45 Hamburg, 20060726_205735, 200 networks

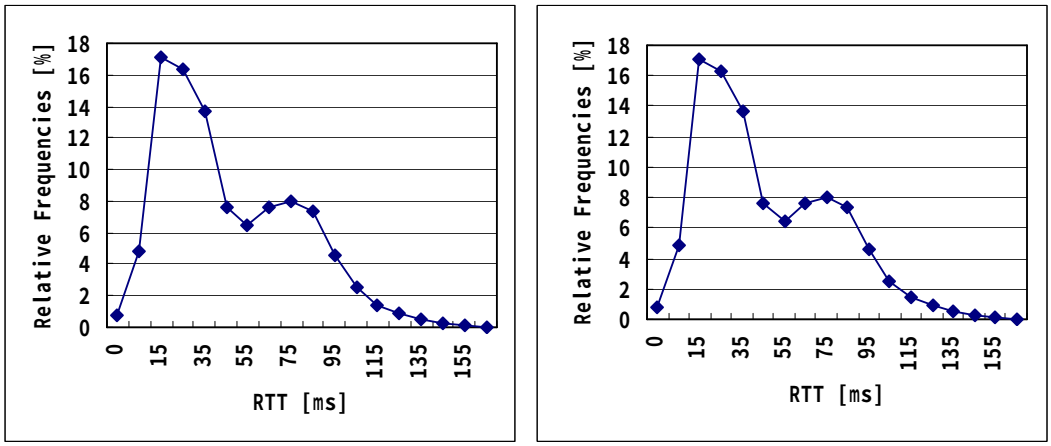

Fig. 46 Hamburg, 20060728_202345, 200 networks

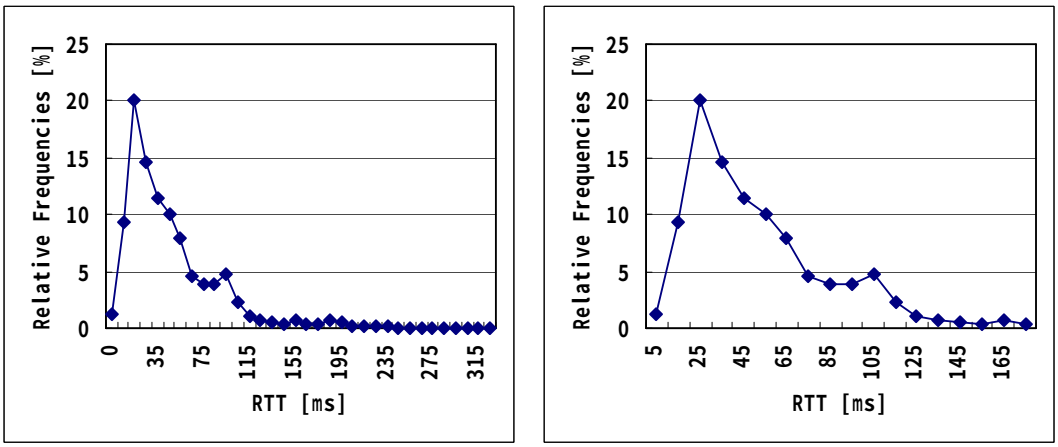

Fig. 47 Hamburg, 20060802_142704, 200 networks

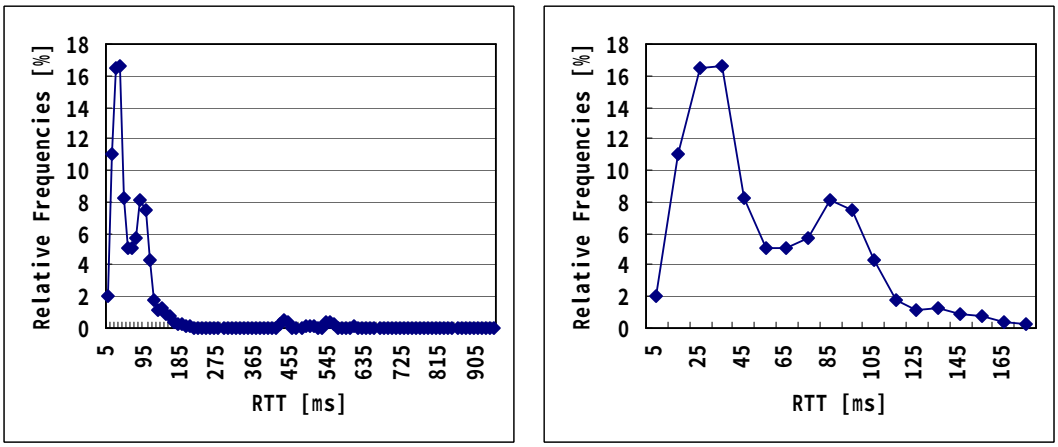

Fig. 48 Hamburg, 20060803_155825, 200 networks

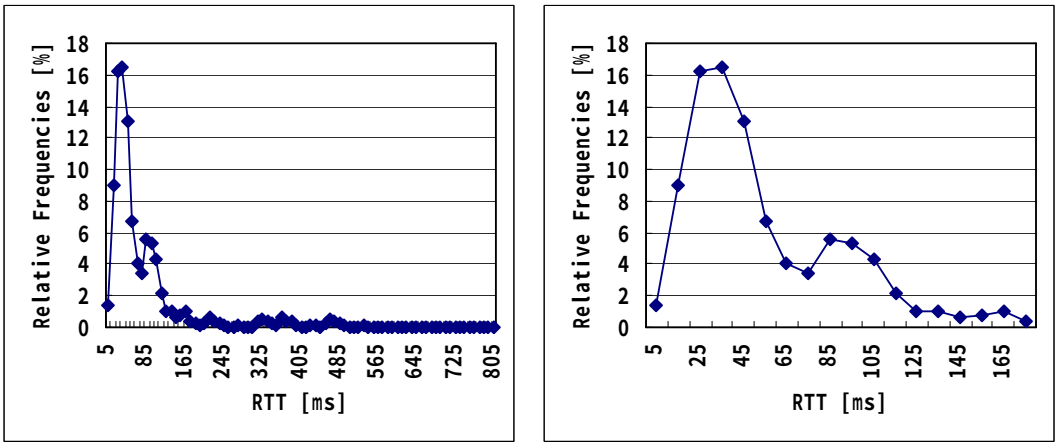

Fig. 49 Hamburg, 20060807_144843, 200 networks

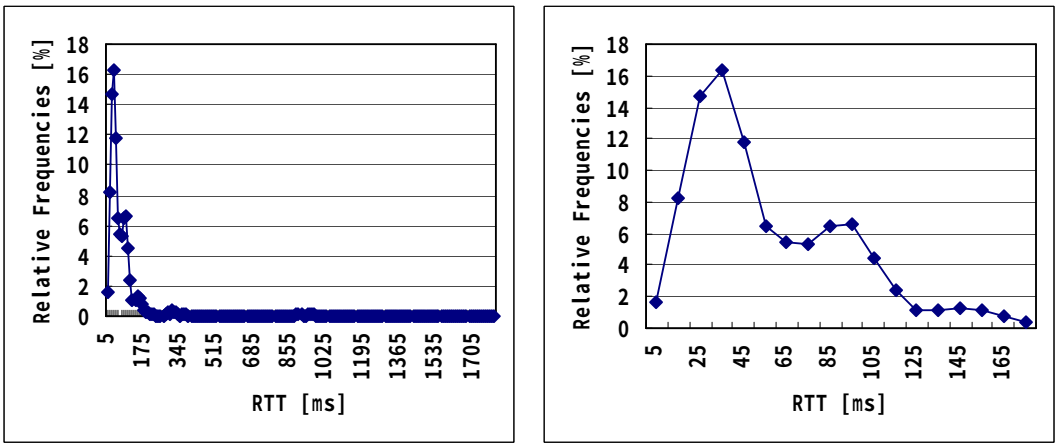

Fig. 50 Hamburg, 20060808_202012, 500 networks

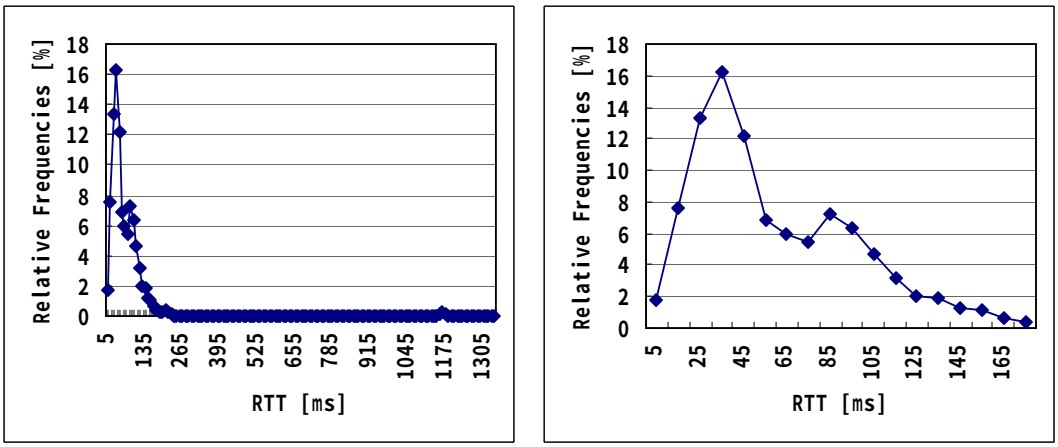

Fig. 51 Hamburg, 20060810_205630, 500 networks

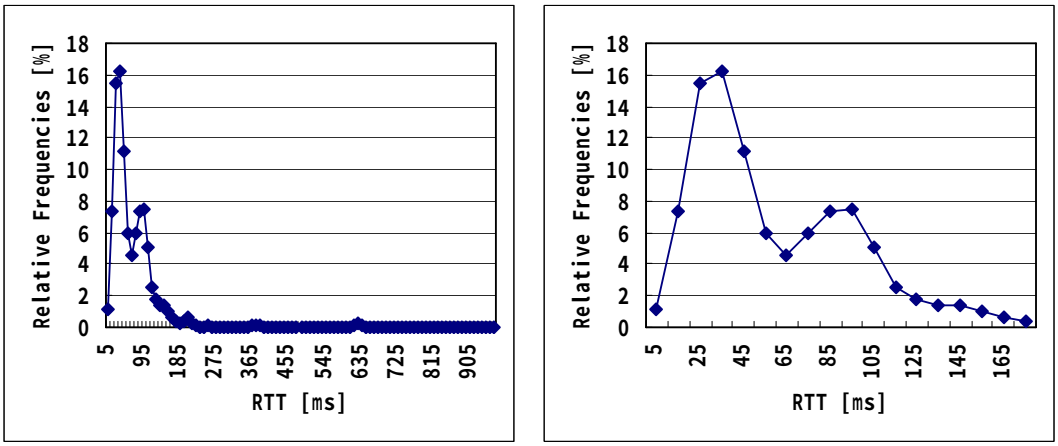

Fig. 52 Hamburg, 20060811_200506, 500 networks

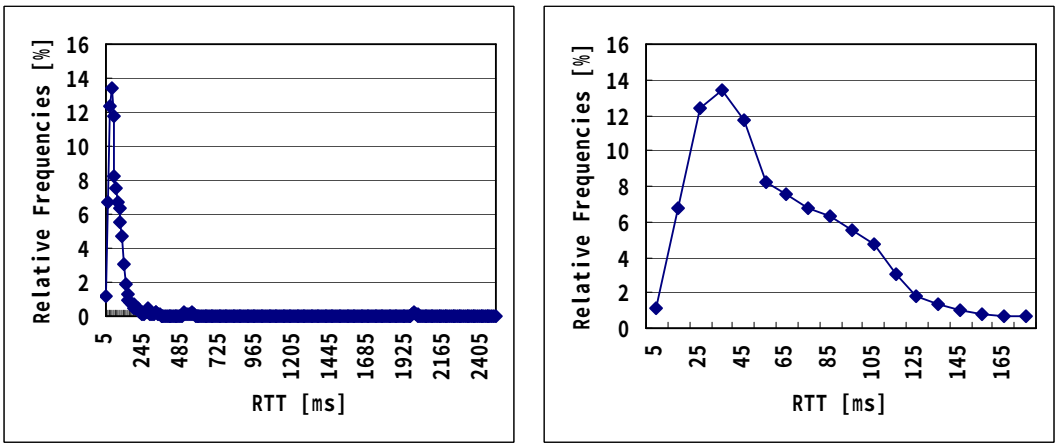

Fig. 53 Hamburg, 20060815_192410, 500 networks

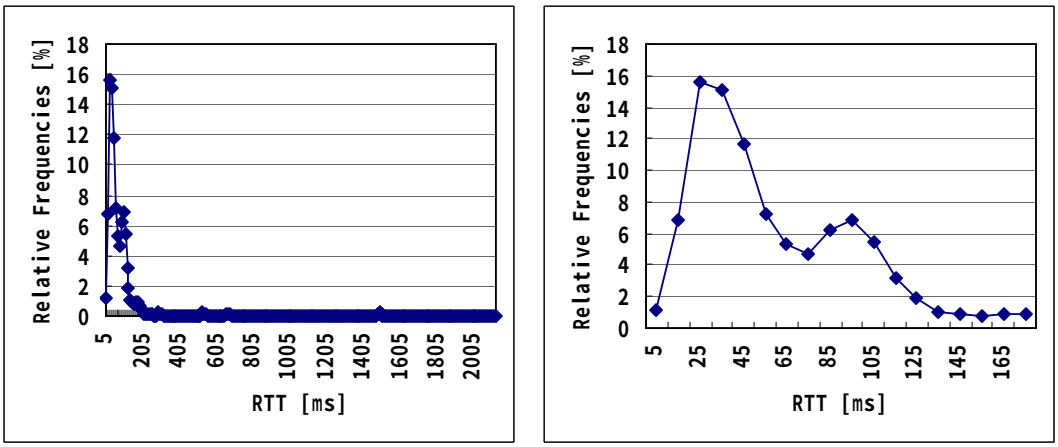

Fig. 54 Hamburg, 20060821_192300, 500 networks

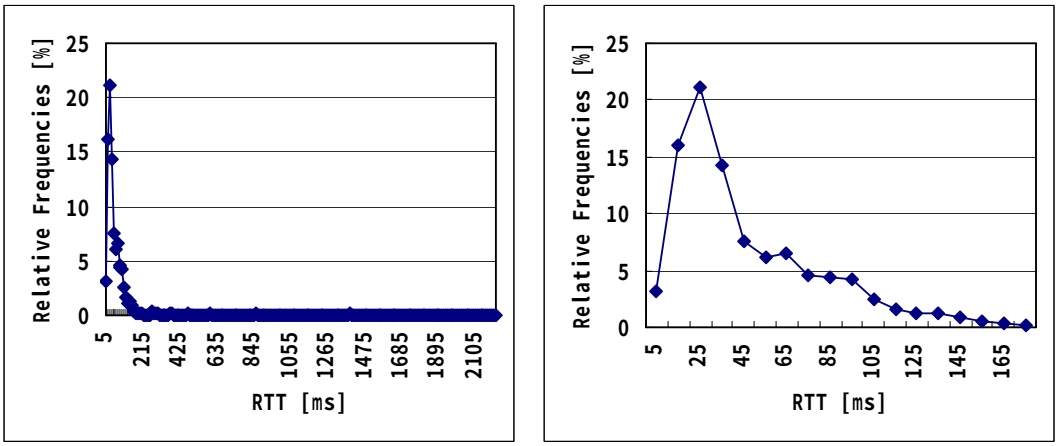

Fig. 55 Hamburg, 20060901_201950, 1000 networks

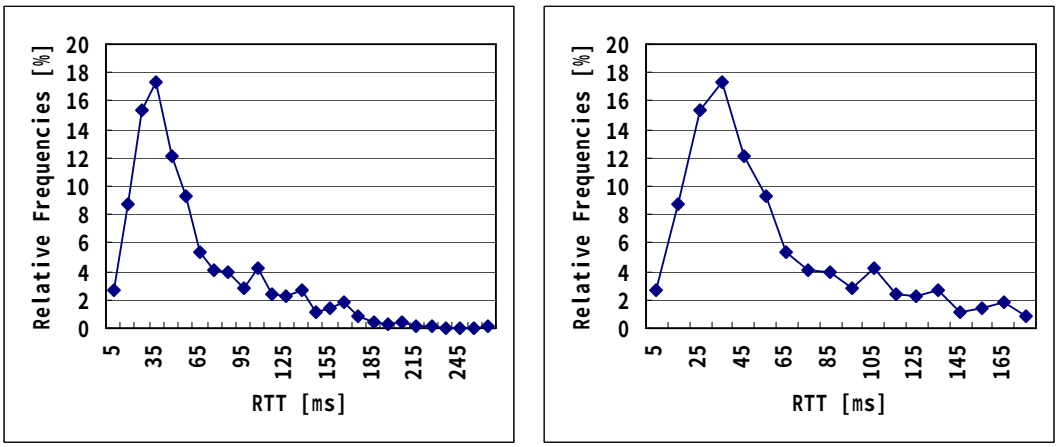

Fig. 56 Hamburg, 20060922_183115, 100 networks

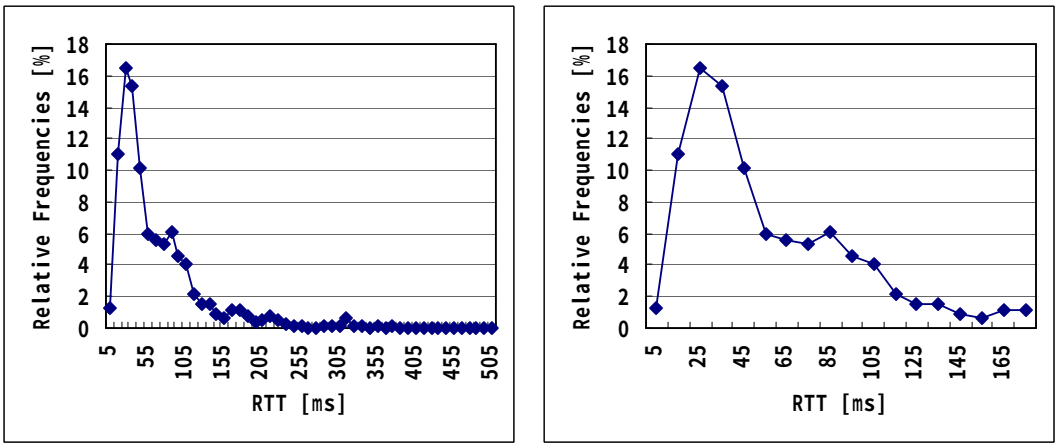

Fig. 57 Hamburg, 20060926_230110, 200 networks

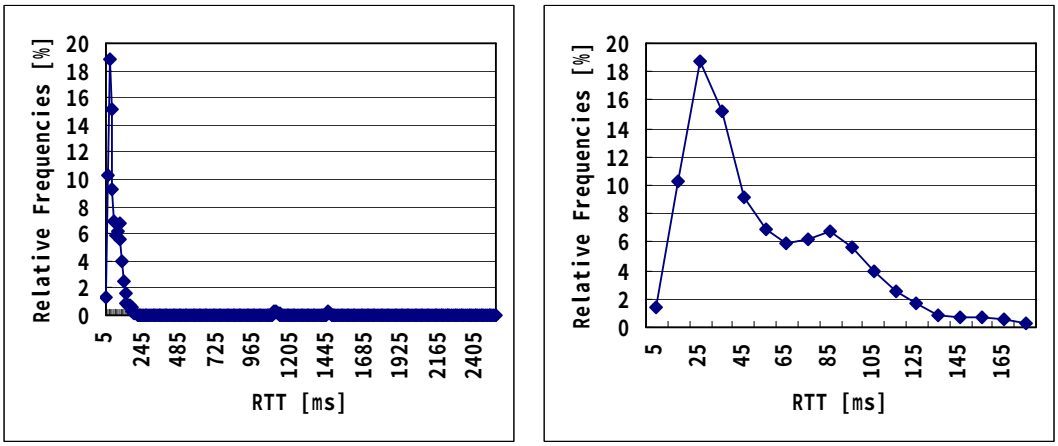

Fig. 58 Hamburg, 20061002_183049, 500 networks

Berlin

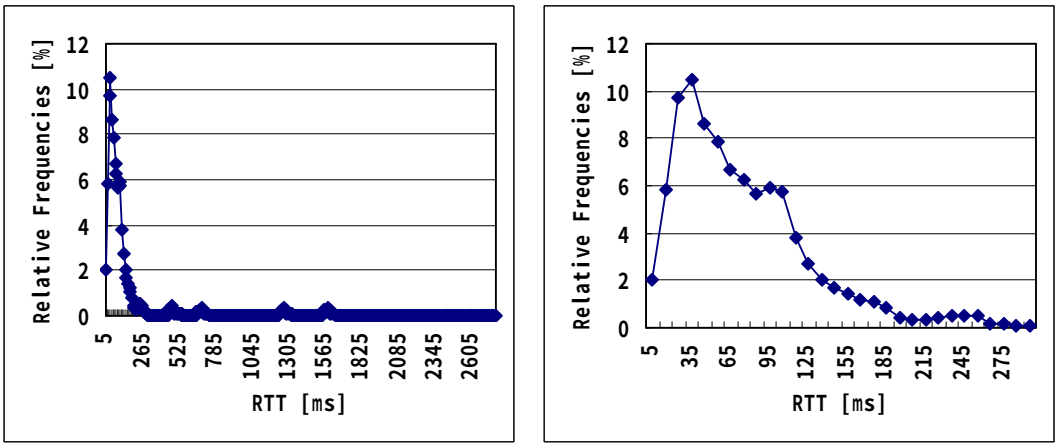

Fig. 59 Berlin, 20060726_024053, 200 networks

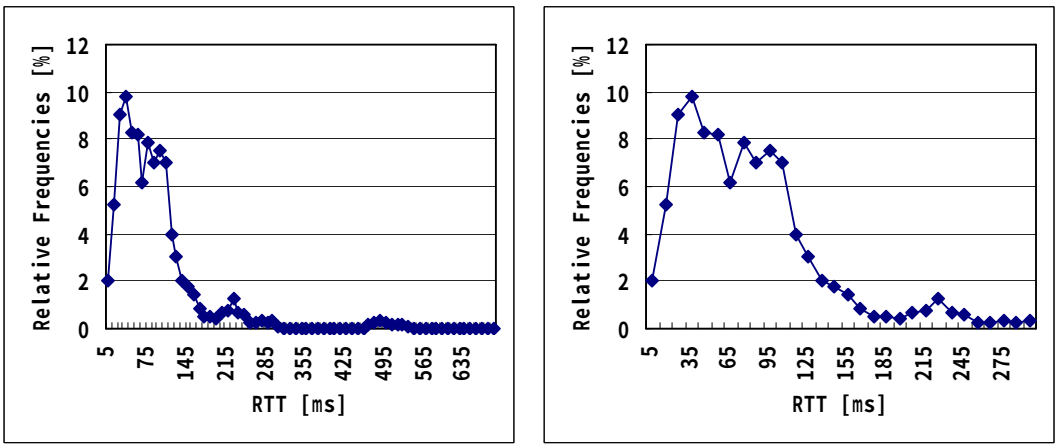

Fig. 60 Berlin, 20060726_171319, 200 networks

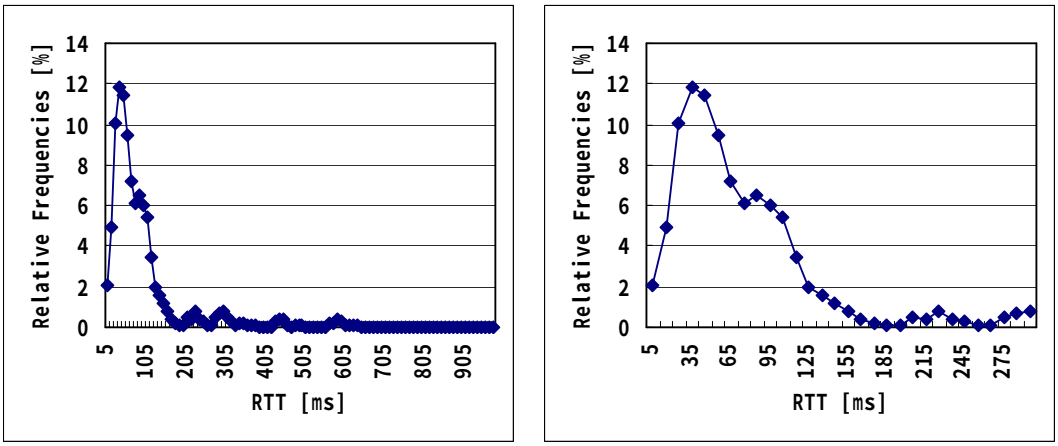

Fig. 61 Berlin, 20060731_042056, 200 networks

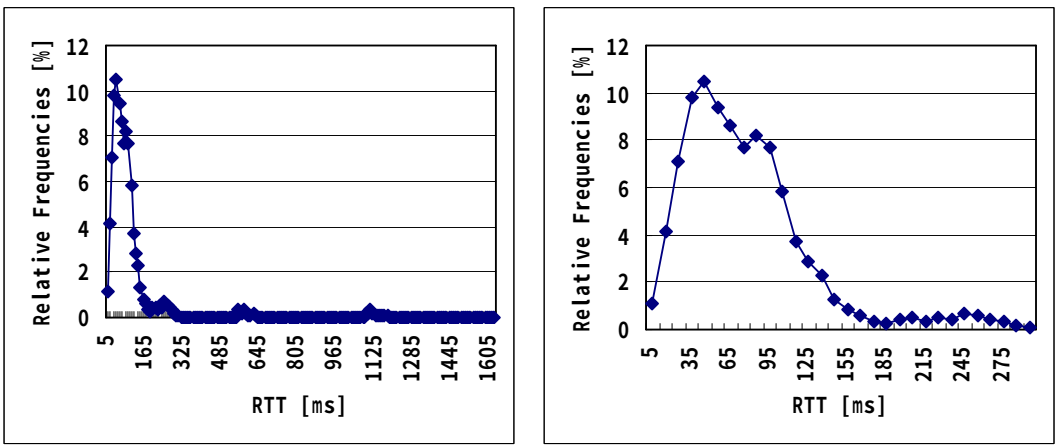

Fig. 62 Berlin, 20060802_150317, 200 networks

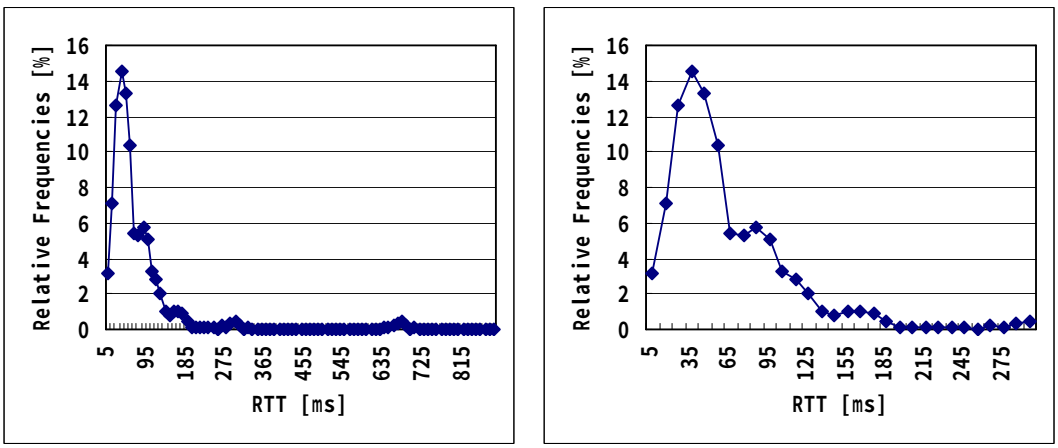

Fig. 63 Berlin, 20060803_152848, 200 networks

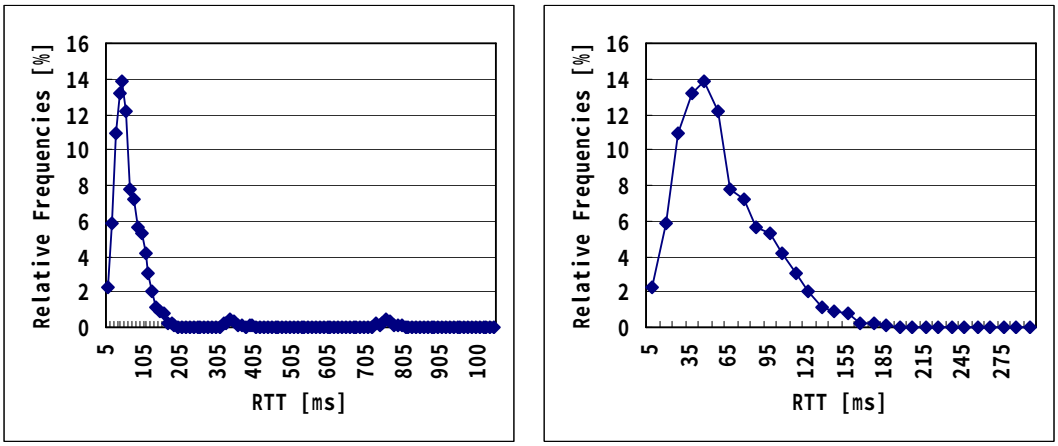

Fig. 64 Berlin, 20060807_142738, 200 networks

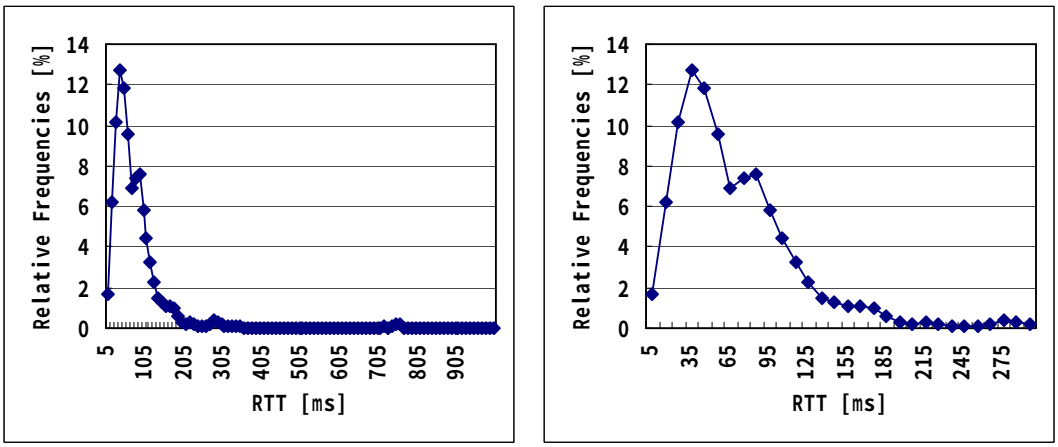

Fig. 65 Berlin, 20060808_065115, 500 networks

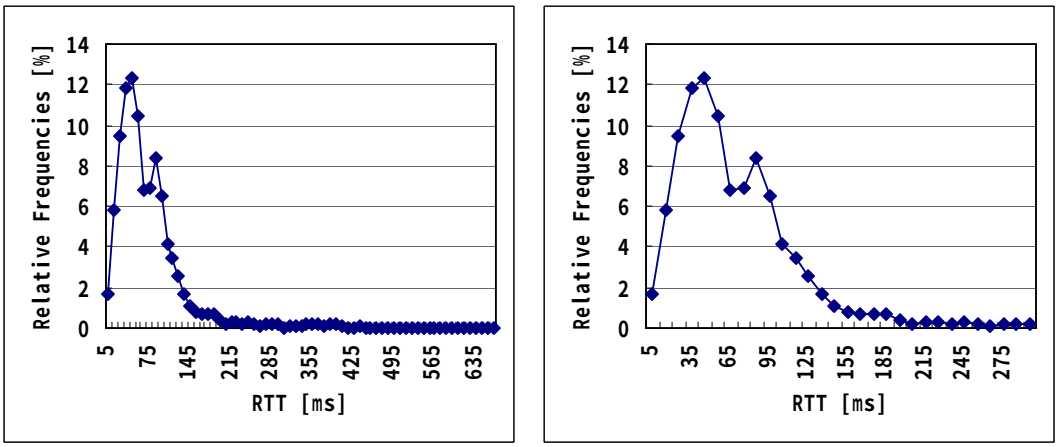

Fig. 66 Berlin, 20060809_041654, 500 networks

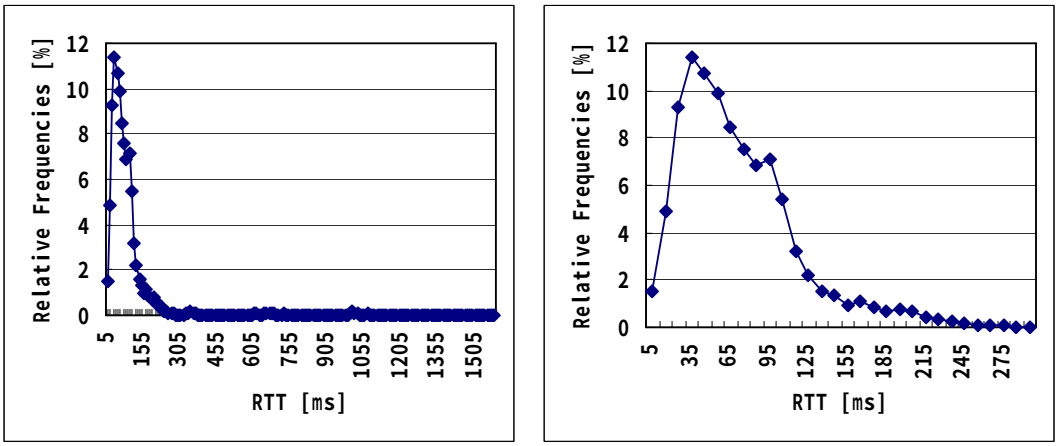

Fig. 67 Berlin, 20060811_042452, 500 networks

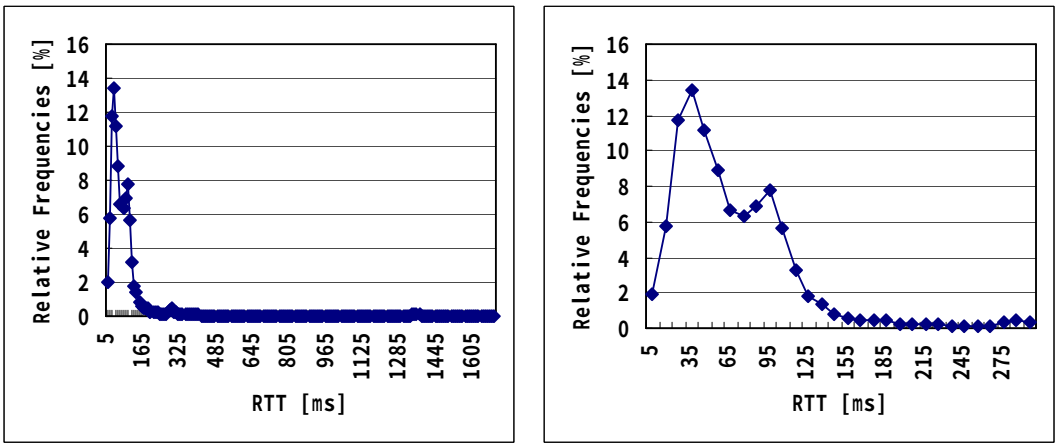

Fig. 68 Berlin, 20060811_222407, 500 networks

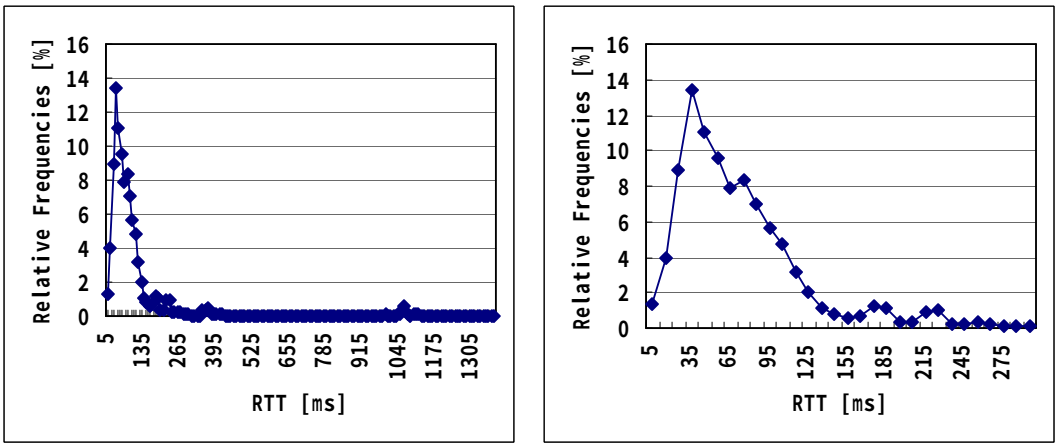

Fig. 69 Berlin, 20061002_163723, 200 networks

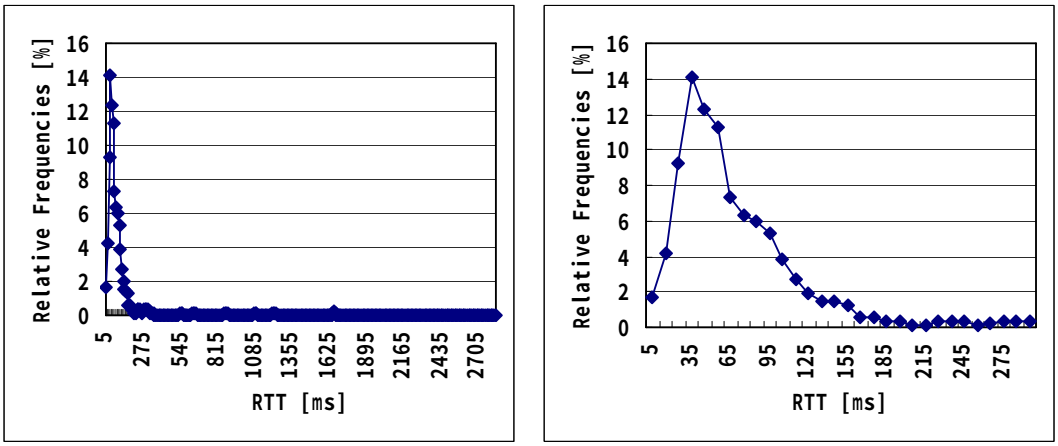

Fig. 70 Berlin, 20061002_232639, 500 networks

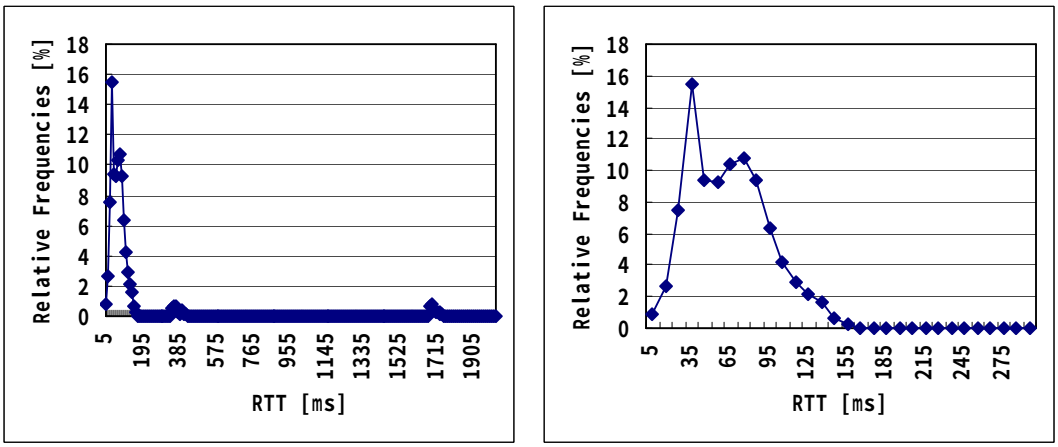

Fig. 71 Berlin, 20061004_161750, 100 networks

San Francisco

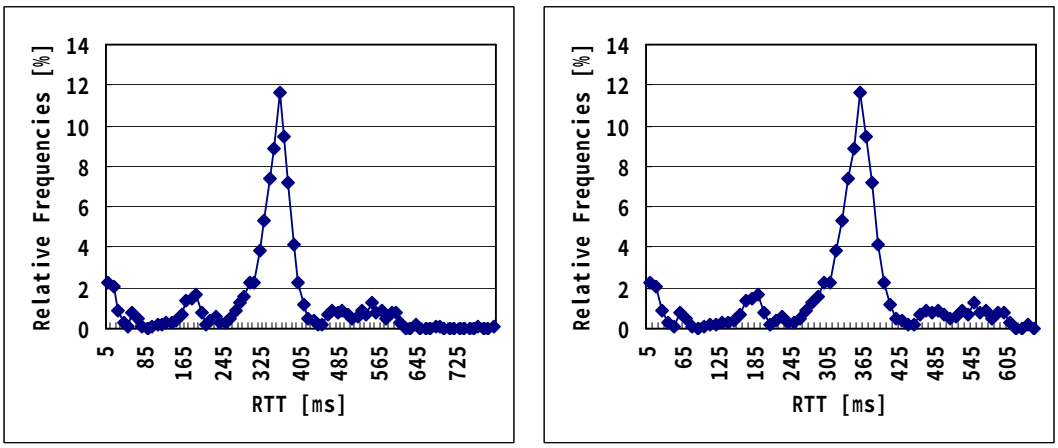

Fig. 72 San Francisco, 20060726_125729, 200 networks

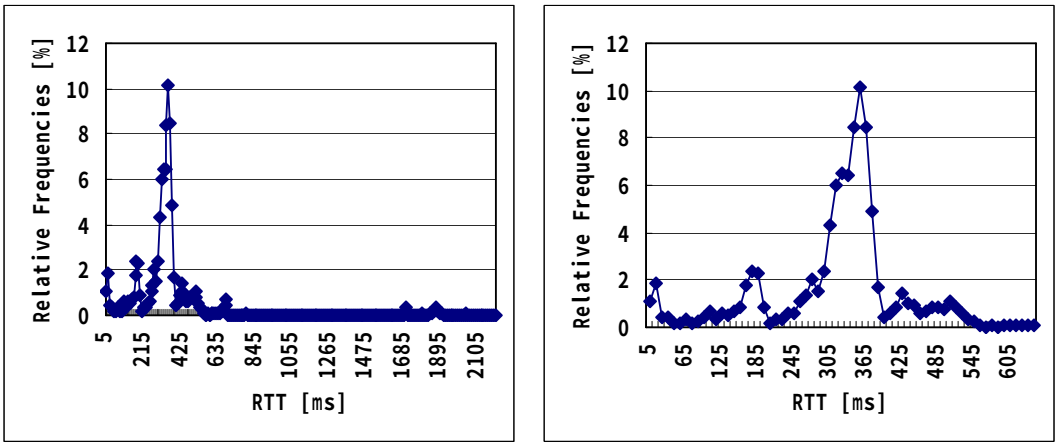

Fig. 73 San Francisco, 20060802_180923, 200 networks

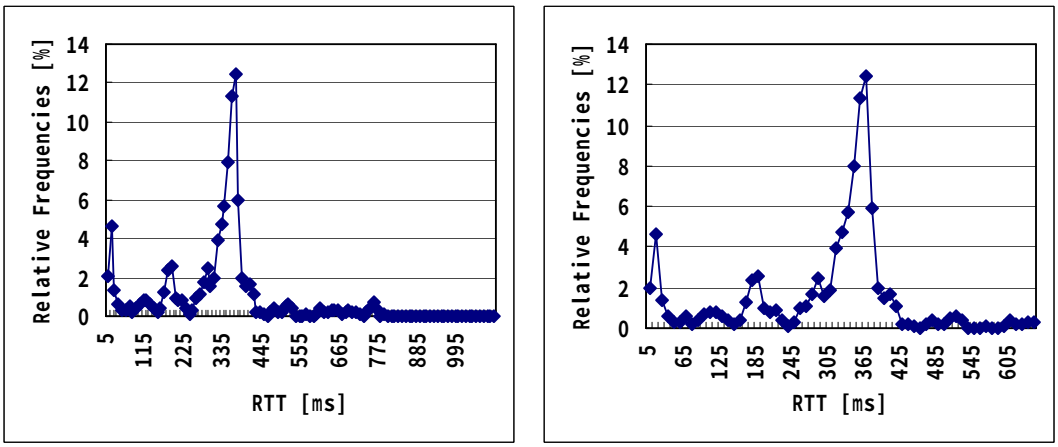

Fig. 74 San Francisco, 20060803_195448, 200 networks

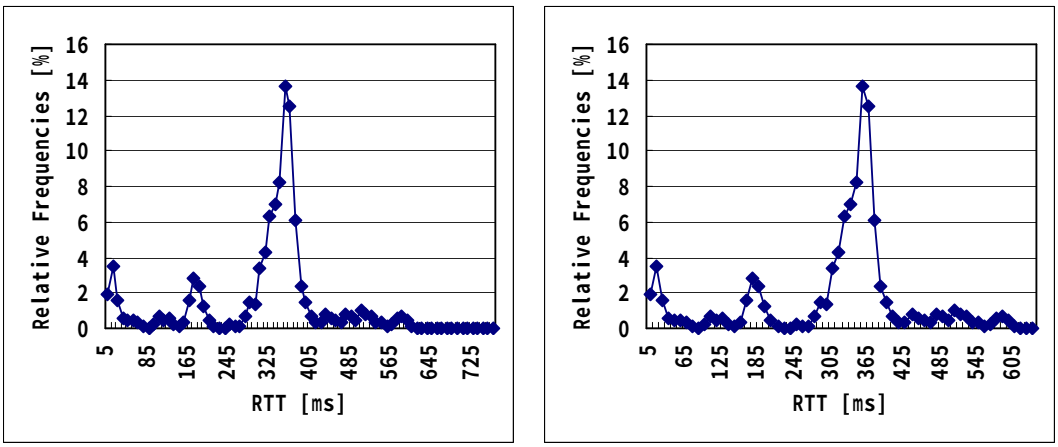

Fig. 75 San Francisco, 20060807_161808, 200 networks

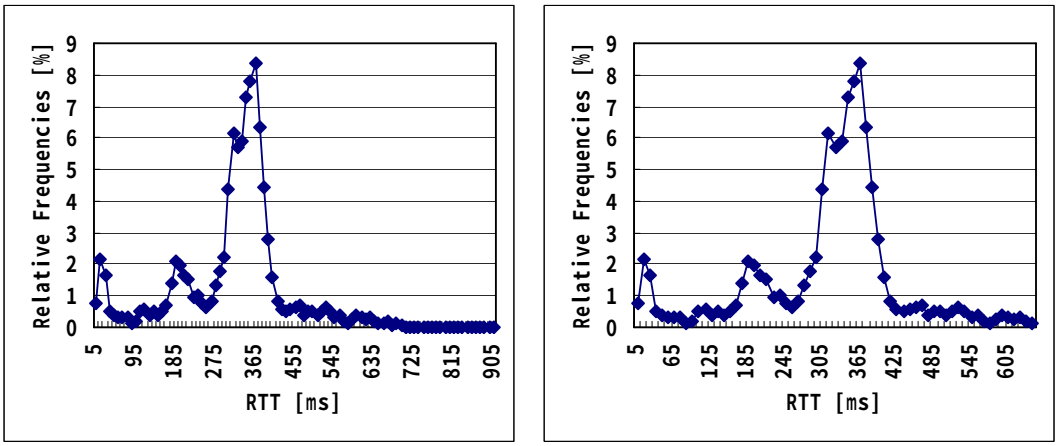

Fig. 76 San Francisco, 20060809_030747, 500 networks

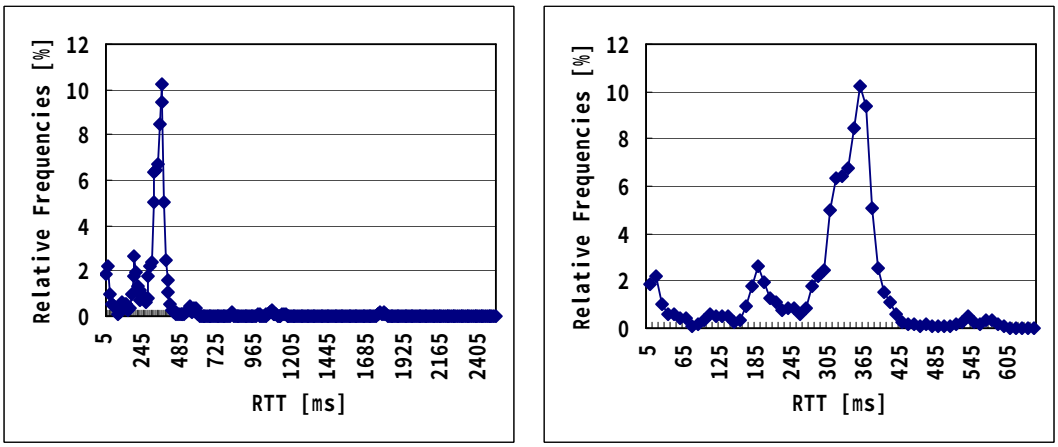

Fig. 77 San Francisco, 20060812_003223, 500 networks

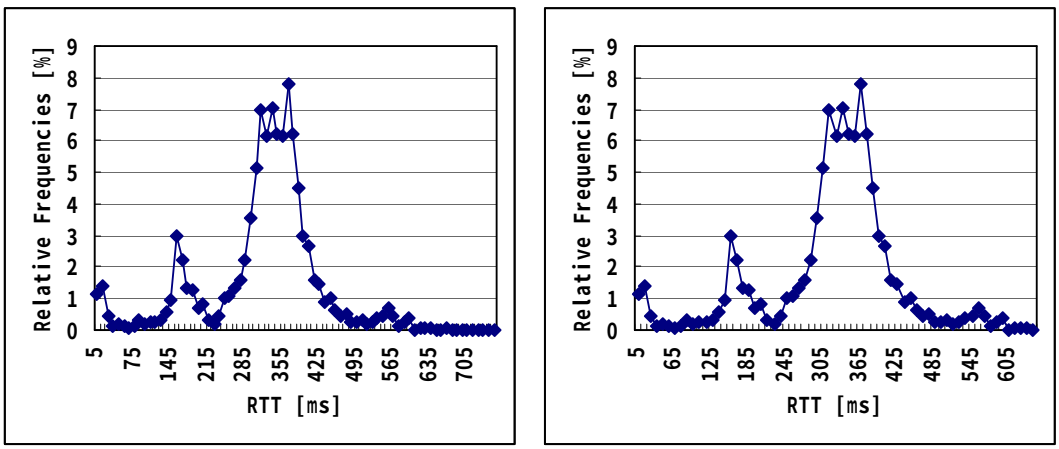

Fig. 78 Fig. 35 San Francisco, 20061107_202944, 200 networks

Shanghai

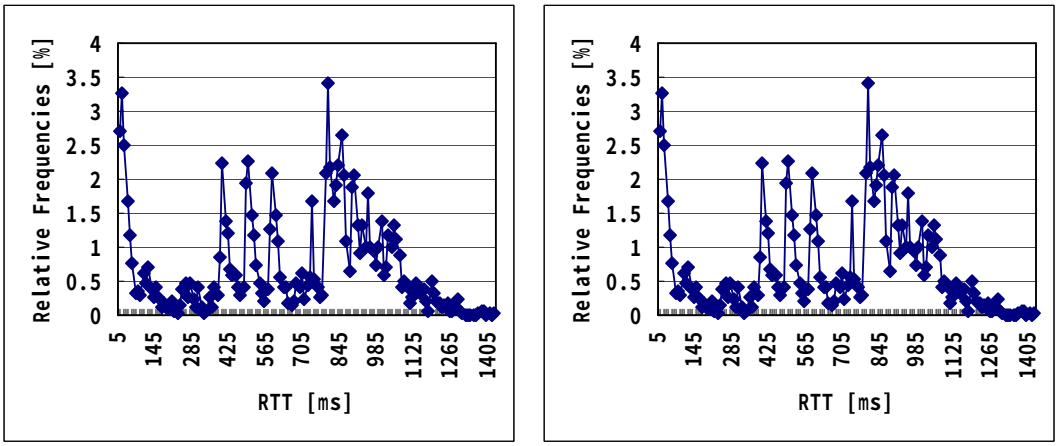

Fig. 79 Shanghai, 20060727_191127, 200 networks

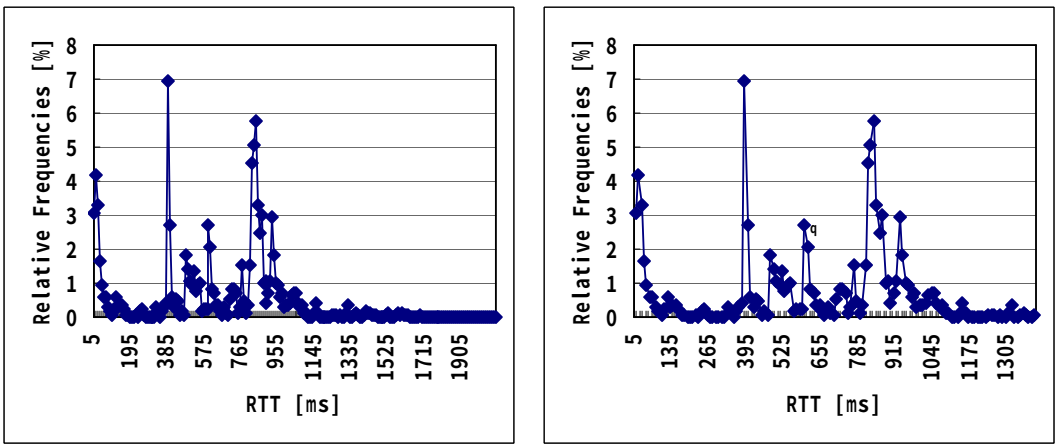

Fig. 80 Shanghai, 20060728_232308, 200 networks

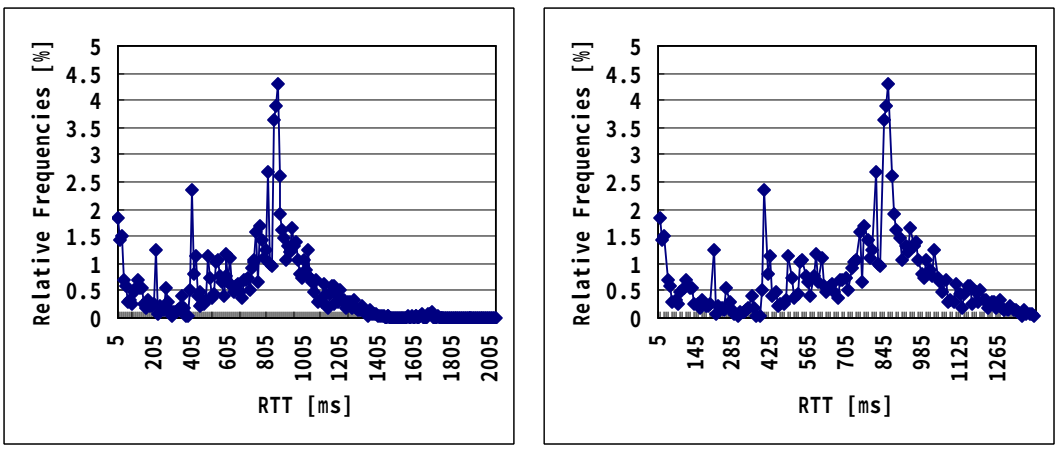

Fig. 81 Shanghai, 20060802_174350, 200 networks

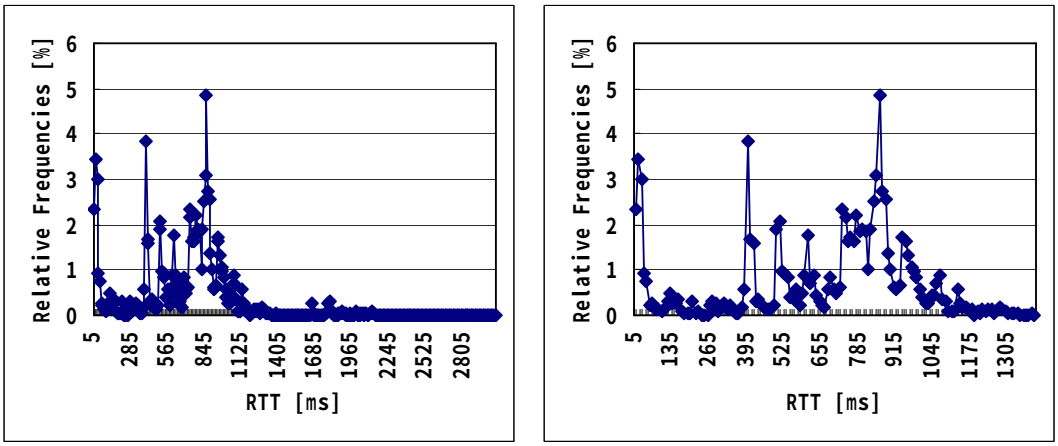

Fig. 82 Shanghai, 20060803_210942, 200 networks

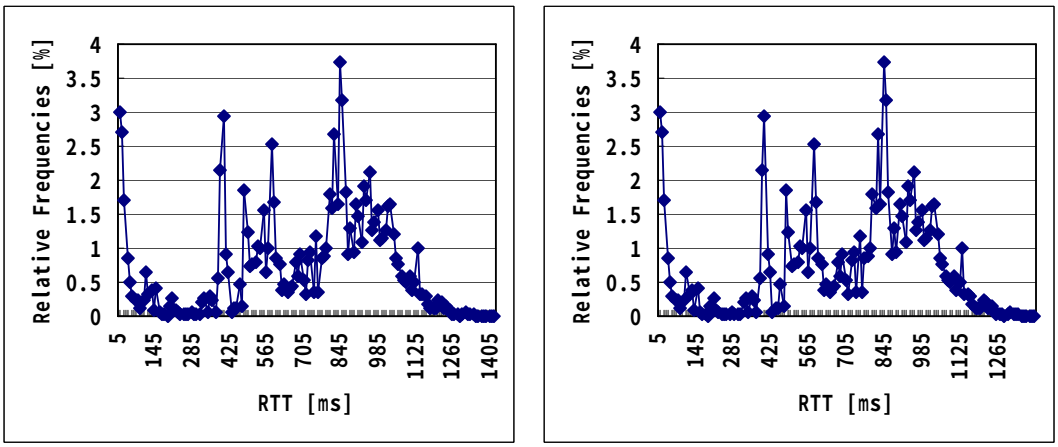

Fig. 83 Shanghai, 20060807_211101, 200 networks

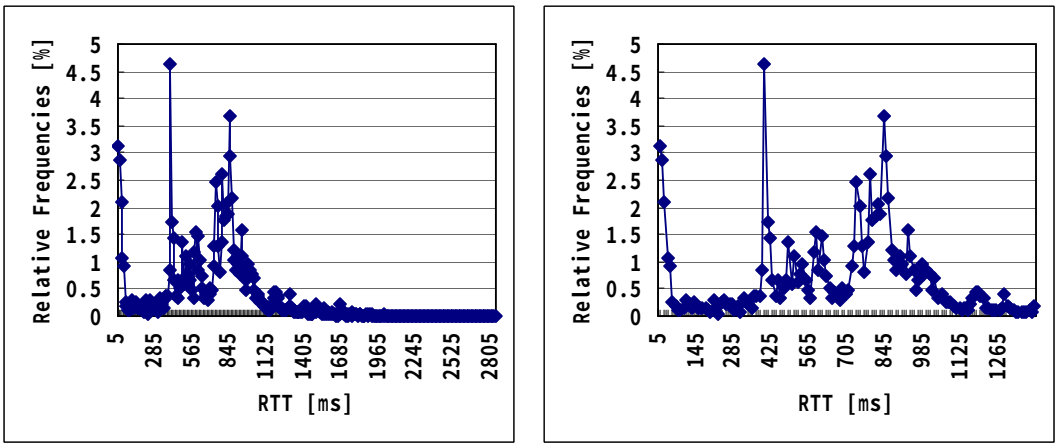

Fig. 84 Shanghai, 20060809_053425, 500 networks

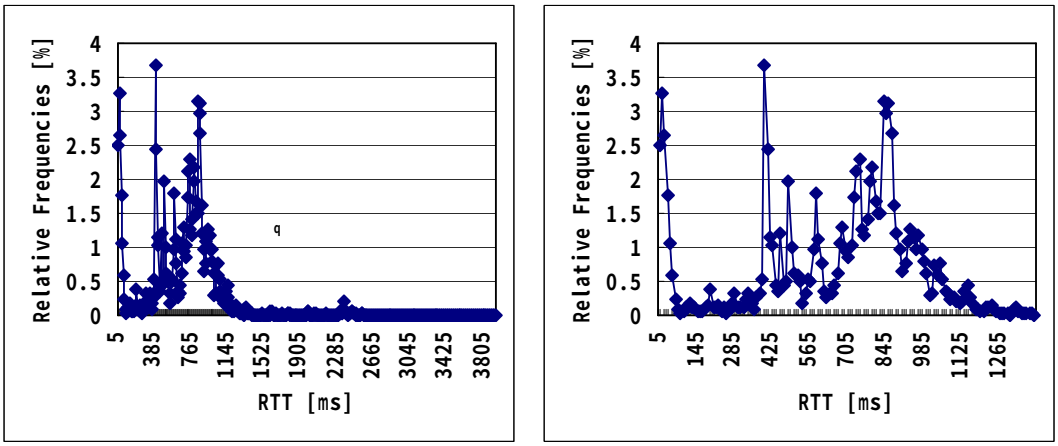

Fig. 85 Shanghai, 20060812_084532, 500 networks

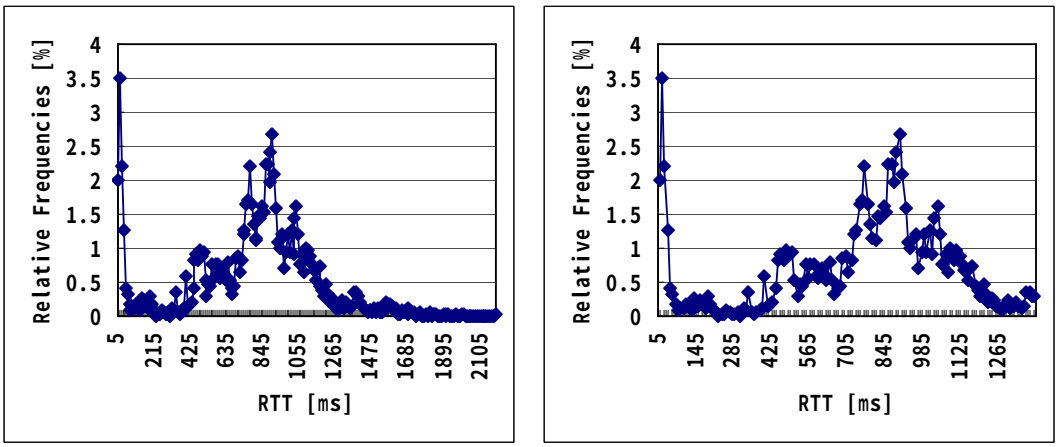

Fig. 86 Shanghai, 20060930_011959, 200 networks

Declaration

I declare within the meaning of section 25(4) of the Examination and Study Regulations of the International Degree Course Information Engineering that: this Master report has been completed by myself independently without outside help and only the defined sources and study aids were used. Sections that reflect the thoughts or works of others are made known through the definition of sources.

City, Date **Signature**