Implementation and verification of an application with active path selection in IPv6 environment.

Makoto Otani
Graduate School of Science and Engineering, Saga University
1, Honjo, Saga 840-8502, Japan
otani@ai.is.saga-u.ac.jp

Kenzi Watanabe
Faculty of Science and Engineering, Saga University
1, Honjo, Saga 840-8502, Japan
watanabe@is.saga-u.ac.jp

Hiroki Kondo
Faculty of Science and Engineering, Saga University
1, Honjo, Saga 840-8502, Japan
kondo_h@ai.is.saga-u.ac.jp

Abstract: We propose a function of active path selection to the destination by users. We implemented the function into the FTP application by using the routing header defined in IPv6 specifications. We also developed the “Support system for quality selection'', which supports the selection by providing information on each path, experimentally. This paper describes details of the function and an implementation of the application. We also describe the “Support system for quality selection” briefly.

1. Introduction

People's purposes to using the Internet are spreading with the growth and the expansion of the Internet. The Internet is used every day as the Internet phone, the Video on Demand, Game, WWW, Mail, etc. As the use purpose of the Internet spreads, users begin to ask the Internet for various requirements, such as reservation of bandwidth, guarantee of security and so on.

We think that the next generation Internet needs to provide a function to select the communication quality according to the purpose of using. This communication quality means bandwidth, security, cost, reliability, etc.

On the other hand, Internet Service Provider (ISP) will provide many lines of various communication qualities in the near future. Therefore, users can select the communication quality that satisfies the purpose.

We propose a function of active path selection to the destination by users. This function offers a method to select lines that a user wants to use from lines of different communication quality. For example, it is the case that a big file is downloaded using a satellite (Fig. 1). In this way, when using other lines temporarily, it works effectively.

We implemented the function by using the routing header defined in IPv6[1)] specifications. It is specified that a packet pass the branch node (for example, the router of Fig. 1) of a lines by the routing header. In this way, a routing header implements as path selection. Since this header does not need to specify all nodes to the destination, path selection is easily implemented.

We implemented and verified the function of this quality selection to FTP application.

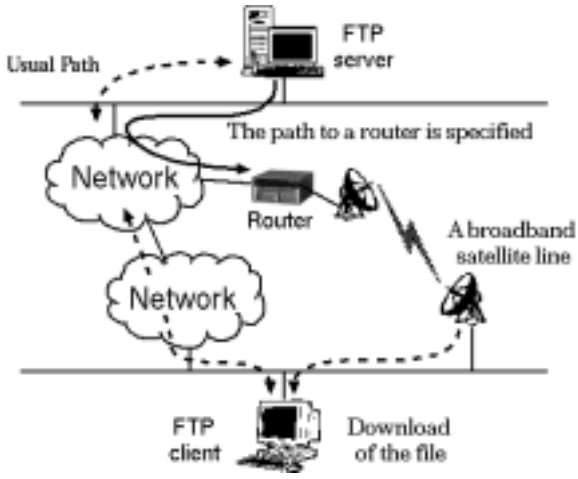

Fig. 1 Selection of the path according to the quality of data

2. Implementation of the Active Path Selection

2.1. Outline of the Active Path Selection

We have added the function in which a user can select the optimal path actively for application from two or more communication paths. This function has been implemented by using the routing header defined in IPv6 specifications.

The path is described as an address list of branch node into the routing header. A selection of paths is enabled by appending this header to an IPv6 packet. We implemented the function of the path selection to FTP application. It has actually implemented by extending the internal

command and transmission parameter code in FTP.

2.2. Selection of the path by the routing header

Active path selection is attained by implementing the function which appends a routing header to application.

As mentioned above, the routing header is defined as an extended header of IPv6. The interpretation and implementation of this header are required in IPv6. So, all applications can use this header. The format of a routing header is shown in Fig. 2.

Fig. 2 Specification of the routing header.

Selection of a path is implemented by describing an address list of the branch node on a path to this header. The first branch address is described to the standard header of IPv6. Other nodes and the address of the destination are described in order to "Address [1- n]" of a routing header (See Fig. 2).

2.3. Implementation of the Active Path Selection

We used the protocol stack developed by KAME Project[2] to implement path selection. Development environment is FreeBSD 3.5 and 4.5[3] [4].

FTP has two connections called a control connection and a data connection for one transaction. Moreover, there are two operations (get and put) for each two connections. Active path selection implements selection of among four connections.

2.3.1. Selection of a sending path

As an interface on the selection for file sending, "putroute" was added to the internal command of FTP client. This commands selects of the sending path of data and shows path information to a FTP server.

The specification of the "putroute" command is shown in Table 1.

Table 1 Specification of the "putroute" command.

Command			Action and Status		
command	option 1	option 2	action	control	data
putroute	**-**	**-**	view	yes	yes
putroute	**(path)**	**-**	config	yes	yes
putroute	**ctl**	**-**	view	yes	-
putroute	**data**	**-**	view	-	yes
putroute	**off**	**-**	config	yes	yes
putroute	**on**	**-**	config	yes	yes
putroute	**ctl**	**(path)**	config	yes	-
putroute	**ctl**	**off**	config	yes	-
putroute	**ctl**	**on**	config	yes	-
putroute	**data**	**(path)**	config	-	yes
putroute	**data**	**off**	config	-	yes
putroute	**data**	**on**	config	-	yes
help	**putroute**	**-**	command help		

The "ctl" and the "data" are the arguments of "putroute". These arguments specify action about the sending of control and data connection, respectively. When those arguments do not exist, it means that both connections are specified.

In the "path" argument, the name of a branched node is separated using @, and described in order. For example, when passing "v6router1", "v6router2", and "v6router4" in order, it describes like "@v6router1@v6router2@v6router4".

The "on" is an argument which enables path selection. And "off" is an opposite meaning of "on". When those arguments do not exist, the path information is shown.

A routing header comprises of path information specified by the user. Moreover, this header is added and sent to a packet.

2.3.2. Selection of a receiving path

As an interface on the selection for file receiving, "getroute" was added to the internal command of FTP client. This command implements the selection of the receiving path of data and the reference of path information to a FTP server. It is the same as that of "putroute".

The specification of the "getroute" command is shown in Table 2.

This command enables a setup and reference of a receiving path.

However, in receiving action, the packet is sent from a FTP server. It is necessary to specify the path of receiving in the FTP server. Therefore, it is necessary to inform the path receiving path specified by the "getroute" command to a FTP server.

We also extended the transmission parameter code of FTP. The code is "CROUTE" and

"DROUTE". The specification of these parameter codes is shown in Table 3.

Table 2 Specification of the "getroute" command.

Command			action and status		
command	option 1	option 2	action	control	data
getroute	**-**	**-**	view	yes	yes
getroute	**(path)**	**-**	config	yes	yes
getroute	**ctl**	**-**	view	yes	-
getroute	**data**	**-**	view	-	yes
getroute	**off**	**-**	config	yes	yes
getroute	**on**	**-**	config	yes	yes
getroute	**ctl**	**(path)**	config	yes	-
getroute	**ctl**	**off**	config	yes	-
getroute	**ctl**	**on**	config	yes	-
getroute	**data**	**(path)**	config	-	yes
getroute	**data**	**off**	config	-	yes
getroute	**data**	**on**	config	-	yes
Help	**getroute**	**-**	command help		

Table 3 Specification of the "CROUTE" and the "DROUTE" code.

Code	Option	Action
CROUTE	**SHOW**	Reference of status (control)
CROUTE	**ON**	Path selection is enabled. (control)
CROUTE	**OFF**	Path selection is disabled. (control)
CROUTE	**(path)**	setup of a path. (control)
DROUTE	**SHOW**	Reference of status. (data)
DROUTE	**ON**	Path selection is enabled. (data)
DROUTE	**OFF**	Path selection is disabled. (data)
DROUTE	**(path)**	Setup of a path. (data)
HELP	**CROUTE**	Help information. (CROOUTE)
HELP	**DROUTE**	Help information. (DROUTE)

The "CROUTE" is a code about a control connection's receiving path, and the "DROUTE" is a code about a data connection's receiving path. The "SHOW" is an argument that shows to the path information. The "ON" is an argument that enables path selection. The "OFF" is an opposite meaning of the "ON". "path" is described using "@".

A FTP server sets the path information according to the sent transmission code. The server answers a result to the client.

The response code of a FTP protocol was extended for this response. When the setup of a path is successful, the server answers "200" and the path information on a response code. Moreover, when the setup fails, the server answers "501" and the path information on a response code.

Operation of a path setup in reception of data is shown in Fig. 3.

If a user sets up a receiving path by the "getroute" command, a FTP client will send the path information to a FTP server using a transmission parameter code. Finally, a FTP server answers a result of the setup using a response code.

Fig. 4 is a log when setting the receiving path in a data connection using the extended command.

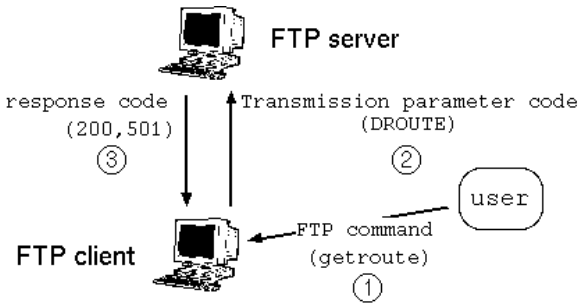

Fig. 3 Operation to set up a receiving path.

```
% ftp v6host1
Connected to v6host1
220 v6host1 FTP server (Version 6.00) ready.
Name (otani):
331 Password required for user
Password:
230 User otani logged in.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> getroute data @v6router1@v6router2
200 DROUTE set data route ->
        '@v6router1@v6router2' command ok.
ftp>
```

Fig. 4 An example of the operation to set up the path.

3. Prototyping of "Support system for quality selection"

As for a user, the active path selection is achieved by the function that we implemented. However, this function needs to specify nodes explicitly, when selecting a path. Therefore, it is necessary to provide quality information. The user who does not understand the quality of a communication path. Therefore, we developed the "Support system for quality selection", experimentally. This system supports the path selection by providing information on each path. This system provides path information that the network administrator knows to the users. Therefore, when a user uses this system, users can select paths without knowing details of network quality. This section describes the function of "Support system for quality selection", and the outline of the implementation.

3.1. The function of "Support system for quality selection"

"Support system for quality selection" is the system which supports the user who does not know line quality. The support is accomplished by providing quality information.

Table 4 An example of a configure table for the selection.

source	destination	quality	path	application
2001:200:160:1::/64	2001:200:160:2::/64	Broadband	@router1@router2	FTP
2001:200:160:1::/64	2001:200:160:2::/64	Secure	@router3	SMTP

In the present version, the administrator registers static information that the network administrator knows. The contents to set up are the IPv6 address of each node, communication quality, a path, the property of application, etc. Using the information provided from this system, a user can select the quality appropriate for the purpose, and can choose communication paths.

3.2. Implementation of "Support system for quality selection"

We developed this system using FreeBSD. Following is an operation of this system.

1. Path selection application sends the property of the address and application, in order to get quality information.

2. "Support system for quality selection" sends quality information to the application.

3. Selecting the quality which a user wants to use, Path selection application sends the information to Support system.

4. The support system realizes the request, then the support system answers the path to the application.

In for example, the case of a network such as Fig. 5, the administrator sets up quality information as shown in Table 4.

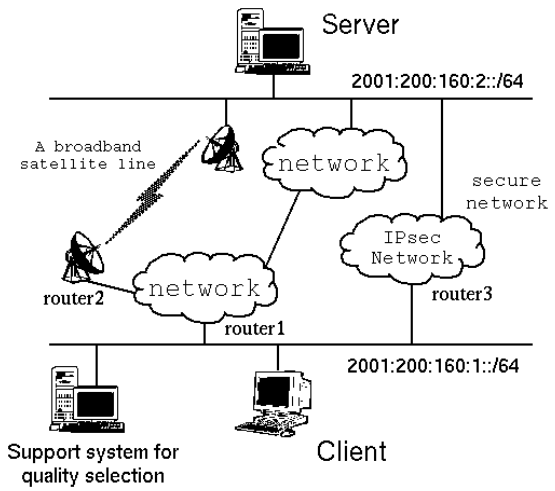

Fig. 5 An example network.

Section 3.3 describes operation of the system when a user selects a path. Section 3.4 describes extension of path selection application.

Table 5 Extension of commands for the support system for quality selection.

Command				Action and Status
command	option 1	option 2	option 3	
putroute	**server**	**-**	**-**	config
getroute	**Server**	**-**	**-**	config
putroute	**Server**	**(quality)**	**-**	config
getroute	**Server**	**(quality)**	**-**	config
putroute	**ctl**	**server**	**(quality)**	config
getroute	**ctl**	**server**	**(quality)**	config
putroute	**data**	**server**	**(quality)**	config
getroute	**data**	**server**	**(quality)**	config
qput	**(quality)**	**(filename)**	**[(filename)]**	config and sending
qget	**(quality)**	**(filename)**	**[(filename)]**	config and receiving

3.3. Operation of "Support system for quality selection"

The example which communicated by using support system is shown in Fig. 6. In this example, it communicates from "host1" to "host3". In this case, it passes through high-speed network for sending. And passes through broadband network for receiving. Following is the operation.

1. FTP application sends the property of the address and application, in order to get quality information.

2. Support system sends quality information to path selection application.

3. The FTP client communicates using the received information.

3.4. Extension of the FTP client

In order to use Support system, it is necessary to implement a mechanism for a FTP client to get information from this system.
Therefore, we extended "putroute" and "getroute" further (Table 5).

If "server" is specified as an option from these two commands, this system will judge a appropriate path and will send path information to a client. Moreover, "qput" and "qget" of a

command send and receive the file immediately after receiving path information from support system, respectively.

Table 6 The average transmission time depends on the number of specified paths.

The number of specification of a path	Average transfer time (second)
0	20.28
1	20.93
2	21.98
3	21.11
4	21.17

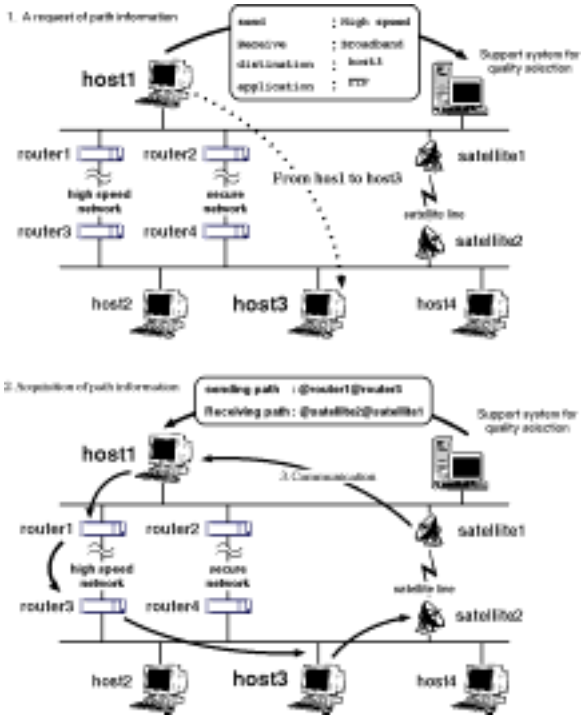

Fig. 6 An example to use the support system for quality selection.

4. Evaluation

We evaluated the path selection by the monitoring of packets. We, also, built two communications paths with different bandwidth, and confirmed the usefulness of path selection using this path. We measured the overhead of a routing header, too.

4.1. Usefulness of path selection

In order to evaluate the usefulness of the path selection application, we built the network shown in Fig. 7. The bandwidths of a communications path are 10Mbps and 100Mbps, respectively. The communication path of 10Mbps is specified as default route. Only when path selection application used, the path of 100Mbps can be used.

The differences in the throughput between 10Mbps and 100Mbps + routing header are shown in Fig. 8. The number of trial is 10 times.

When path selection application is used as shown by Fig. 8, it turns out that a communication band is large. From these results, path selection application works well.

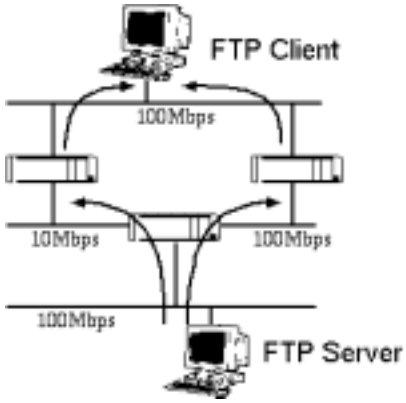

Fig. 7 The topology of the experimental network.

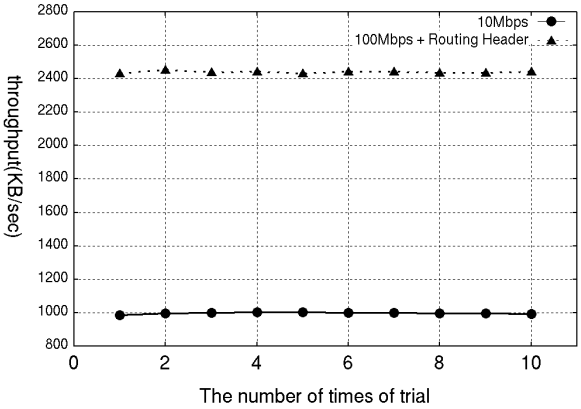

Fig. 8 The throughput of both paths.

4.2. Overhead of a routing header

When a path specifies, the extended header as in Fig. 2 is added to an IPv6 header. The extended headers are processed at each node. It may make an overhead compared with a case of no routing headers. Therefore, we evaluate this overhead. Table 6 shows the transfer rate by the difference in the number of routing headers when sending 50MB file using the network of 100Mbps. The result of this measurement showed that existence of a routing header affected the transfer time about an average of 0.65 seconds. And whenever one node increased, it about an average of 0.08 seconds increased. We think the overhead is small enough. Moreover, the experiment also showed that the effect of a routing header become small

as the bandwidth become large.

As shown above, it is thought that the path selection by the routing header is a useful function[5].

5. Conclusion

We proposed a function of active selection of the path to the destination by users. We implemented the function into the FTP application by using the routing header defined in IPv6 specifications.

We also developed the “Support system for quality selection", which supports the selection by providing information on each path, experimentally.

In the future, this system will be verified using a wide area network. Moreover, we carry out improvement in function of a support system.

References

1) Deering, S. and Hinden, R.: Internet Protocol Version 6 (IPv6) Specification, Request for Comments 2460 (Dec. 1998)
2) KAME Project : http://www.kame.net
3) Stevens, W. and Thomas, M.: Advanced Sockets API for IPv6, Request for Comments 2292 (Feb. 1999)
4) Stevens, W. and Thomas, M.: Advanced Sockets API for IPv6, draft - ietf - ipngwg - 2292bis01 (Dec. 1999)
5) Makoto Otani, Nobuhide Tuda, Kenzi Watanabe and Hiroki kondo: Implementation and verification of an application with an active routing function using IPv6 routing headers, Proceedings of the 60th Annual Meetings of Information Processing Society of Japan (in Japanese) (Mar. 2000)