Online Games Mobile Internet i-appli

Technology Reports

i-appli online and i-appli call Systems for Real-time Communication with Mobile Terminals

We developed a network and mobile terminal system to extend the i-appli service to provide highly real-time application services such as online games.

Core Network Development Department *Noriko Mizuguchi*
Communication Device Development Department *Makoto Oku*
Services Platform Department *Norio Tanaka*

1. Introduction

With widespread use of the Internet and the evolution of advanced functions for terminals, various business players have appeared to provide applications and content on the terminal platform, and it is expected that the terminal application market will grow continuously in the future. In recent years, those applications have required diversified functionality and multiple levels of quality. Therefore, the protocols should be applied as is appropriate for the applications to satisfy the requirements. For example, online games select a real-time protocol rather than Hyper Text Transfer Protocol (HTTP) because they frequently exchange small amounts of data in Peer-to-Peer (P-to-P)[*1] communication. In the existing NTT DOCOMO packet based network, HTTP is the only available protocol for Web browsing, mail and i-appli communication [1]. Therefore, the NTT DOCOMO packet-based network must offer protocols other than HTTP to accommodate a variety of Internet application programs such as online games.

The only mechanism for Content Providers (CP)[*2] to install application programs onto NTT DOCOMO mobile terminals is the i-appli service. CPs have requested invocation of i-appli with push type instructions from the server and support of protocols other than HTTP.

We therefore developed "i-appli online" functions that enable use of real-time protocols for i-appli, and "i-appli call™[*3]" functions that enable a mobile terminal or CP server to push instructions to invoke i-appli on other mobile terminal via a network.

In this article, we describe the implementation of the i-appli online and i-appli call systems. This implementation achieves highly real-time application services for mobile terminals.

2. Service Overview

The i-appli online is extended service for i-appli that allow multi-person real-time communication. An overview of i-appli online is shown in **Figure 1**. There are two types of communication topology[*4]. One is P-to-P, in which mobile terminals communicate with each other without intermediate servers, and the other is Client-Server (C/S), in which mobile terminals communicate with servers on the Internet. These types of communication are programmed in i-appli programs as socket[*5] communication using Transmission Control Protocol (TCP)[*6] and User Datagram Protocol (UDP)[*7].

The i-appli call is a function in which an i-appli reads information from the phone book of the mobile terminal and pushes instructions to invoke

*1 **P-to-P**: A form of network connection in which computer exchange data as peers, without going through an intermediary server.
*2 **CP**: An enterprise that provides content over the Internet.
*3 **i-appli call™**: A trademark of NTT DOCOMO
*4 **Topology**: Physical interconnections between nodes.
*5 **Socket**: An IP address and port number pair that serves as an interface that hides the details of TCP/IP communication with an application.

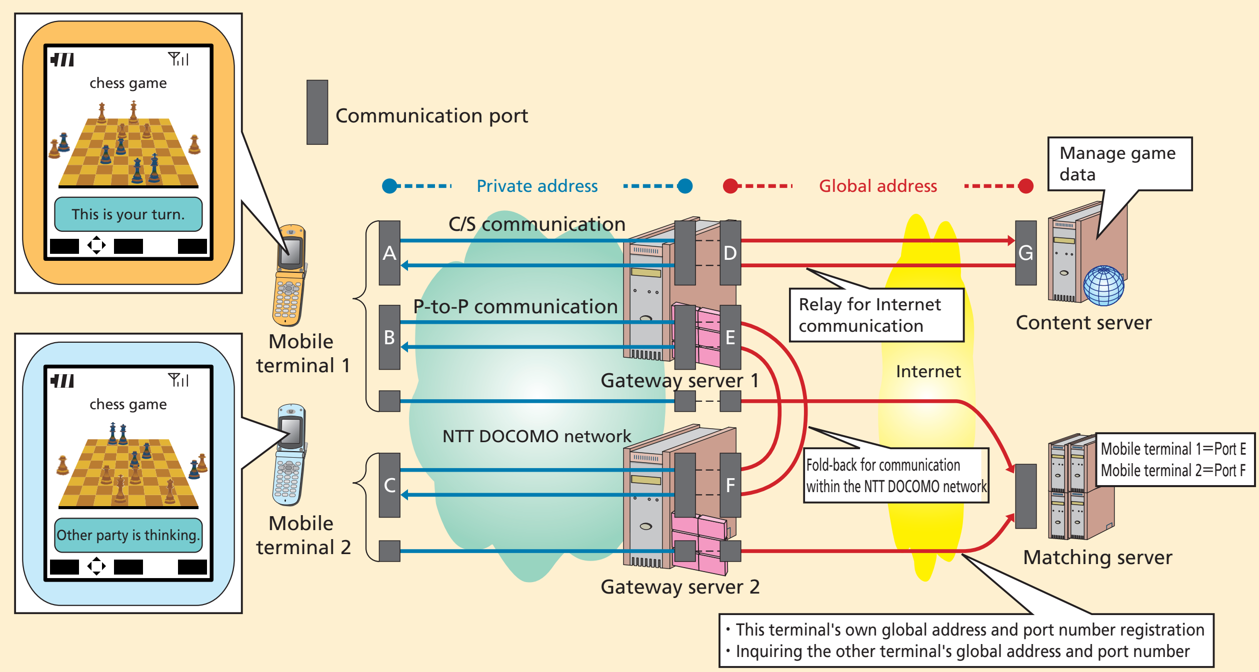


Figure 1 Overview (i-appli online)

an i-appli on another mobile terminal. It also allows a CP server to push-start a mobile terminal i-appli.

Implementing these functions enables i-appli online games and also allows i-appli to be invoked and controlled from the corporate servers.

3. Technical Issues

3.1 i-appli online

1) Issues Concerning Address and Port Notification

Each mobile terminal requires a unique address for packet communication. However, it is difficult to allocate global addresses to all of the terminals connected to the network with Internet Protocol version 4 (IPv4)[*8] due to the limited global address[*9] space. Proxy servers are generally used to economize on addresses to accommodate many terminals. Proxy servers dynamically allocate private addresses[*10] to terminals, and when a terminal accesses an external network, they convert the private address to a global address. The proxy server can economize on addresses with even greater efficiency by using a single global address for multiple terminals and using the port[*11] number to distinguish communication to specific terminals. The NTT DOCOMO network has already implemented this system on the gateway servers.

To achieve P-to-P communication, however, the mobile terminal needs to send the allocated global address and port number to the other mobile terminal. Therefore, the mobile terminal should have the following functions.

- Acquisition of the global address and port allocated from the gateway sever.
- Notification of the global address and the port number to the other terminal.

The gateway servers should also have the following function.

- Relaying of bidirectional data communication among the mobile terminals.

2) Issues Concerning Application Protocols

Depending on the application protocol, the terminal's address may be embedded and used for processing as in

*6 **TCP**: A standard Internet upper-layer protocol above IP. It complements IP by providing functions for confirming the other party in the connection and data arrival, performing flow control, and detecting data duplication or loss to achieve highly reliable communication.

*7 **UDP**: A standard Internet upper-layer protocol above IP. It differs from TCP by not providing functions for confirming the establishment of communication between server and terminal or retransmission of data that does not reach the destination.

*8 **IPv4**: The Internet protocol that is currently used. Address resources are managed as 32-bit numbers.

*9 **Global address**: An IP address that is allocated to a particular terminal connected to the Internet and is unique across all networks.

Session Initiation Protocol (SIP)*12. In networks that perform communication layer address conversion, attention must therefore be given to whether or not the application of each terminal is working properly. In general, some proxy servers are restricted to a certain application protocol to avoid such issues, and NTT DOCOMO network also implemented gateway servers which allow only HTTP transactions. However, to adapt to the diversification of application protocols, the network functions must relay communication at the transport protocol*13 (TCP and UDP) level without depending on the application protocols used in the upper layers.

3.2 i-appli call

In the mobile terminal, Java Application Manager (JAM) is an i-appli control module that invokes the i-appli program. To invoke an i-appli from another mobile terminal or a server, an invocation message is sent to the JAM. To achieve such functions requires two methods. One is to target the mobile terminal to invoke an i-appli from other mobile terminals or servers and the other is the transmission method for sending the invocation message. Furthermore, the invocation message must contain information that specifies the target i-appli program and the parameters, etc. to be passed to the i-appli program at invocation.

4. Implementation

4.1 i-appli online

1) Processing to Establish a Communication Path with the Control Message

To solve the issues related to address notification described above, we created the use of messages to apply to the mobile terminals and the gateway servers to establish a communication path between the two mobile terminals.

Mobile terminals and gateway servers communicate by using the private addresses. The global address is used for communication beyond the gateway servers. The mobile terminal sends a control message to the gateway server as a client using the private address. The control message processing for P-to-P communication and C/S communication is shown in **Figure 2**. The processing that is common to both communications is explained for the example of P-to-P communication using UDP. Mobile terminal 1 sends a global address allocation request to gateway server 1 to get a global address allocated for its own use (Fig. 2 (1)). At that time, the mobile terminal 1 notifies the gateway server 1 of it's own port number, the transport protocol, and the communication topology that will be used in that communication transmission. When the gateway server 1 receives the request, it allocates the global address and port (Fig. 2 (2)), and returns the allocation completed notification to the mobile terminal 1 (Fig. 2 (3)). At the same time, the information is stored in the mapping table maintained by the gateway server 1.

Next, we present the process that is specific to the communication topology. For P-to-P communication, mobile terminal 2, which is the other terminal in the communication, also sends the global address allocation request processing and receives a global address and a port in the same way as the mobile terminal 1. Establishing a P-to-P communication path requires the exchange of global addresses and ports between the two mobile terminals. The two mobile terminals access a matching server*14 placed in the Internet by CP, and the transaction proceeds (Fig. 2 (4)). When TCP is used as the transport protocol, each mobile terminal is notified of its status as client or server when the TCP connection is established. Next, client mobile terminal 1 sends a connection destination set-up request (Fig. 2 (6)-a). When TCP is used as the transport protocol, a TCP connection between the mobile terminal and gateway server 1 is established before the mobile terminal sends the connection destination set-up request (Fig. 2 (5)). After gateway server 1 receives the connection destination set-up request, it establishes a connection to the global address and port of mobile terminal 2 (gateway server 2 in practice) (Fig. 2 (7)-a, (8)). On the other hand, mobile terminal 2 also sends a connection des-

*10 **Private address**: An IP address that is allocated to identify terminals within a local network. Uniqueness across all networks is not guaranteed, so communication over the Internet with that address itself is not possible.

*11 **Port**: In TCP/IP communication, a sub-address below an IP address used to specify different channels for communication on the same terminal.

*12 **SIP**: A call control protocol defined by the Internet Engineering Task Force (IETF) and used for IP-phone with VoIP, etc.

*13 **Transport protocol**: A protocol used in the transport layer. The main Internet transport protocols are the connection-type TCP and the connectionless UDP.

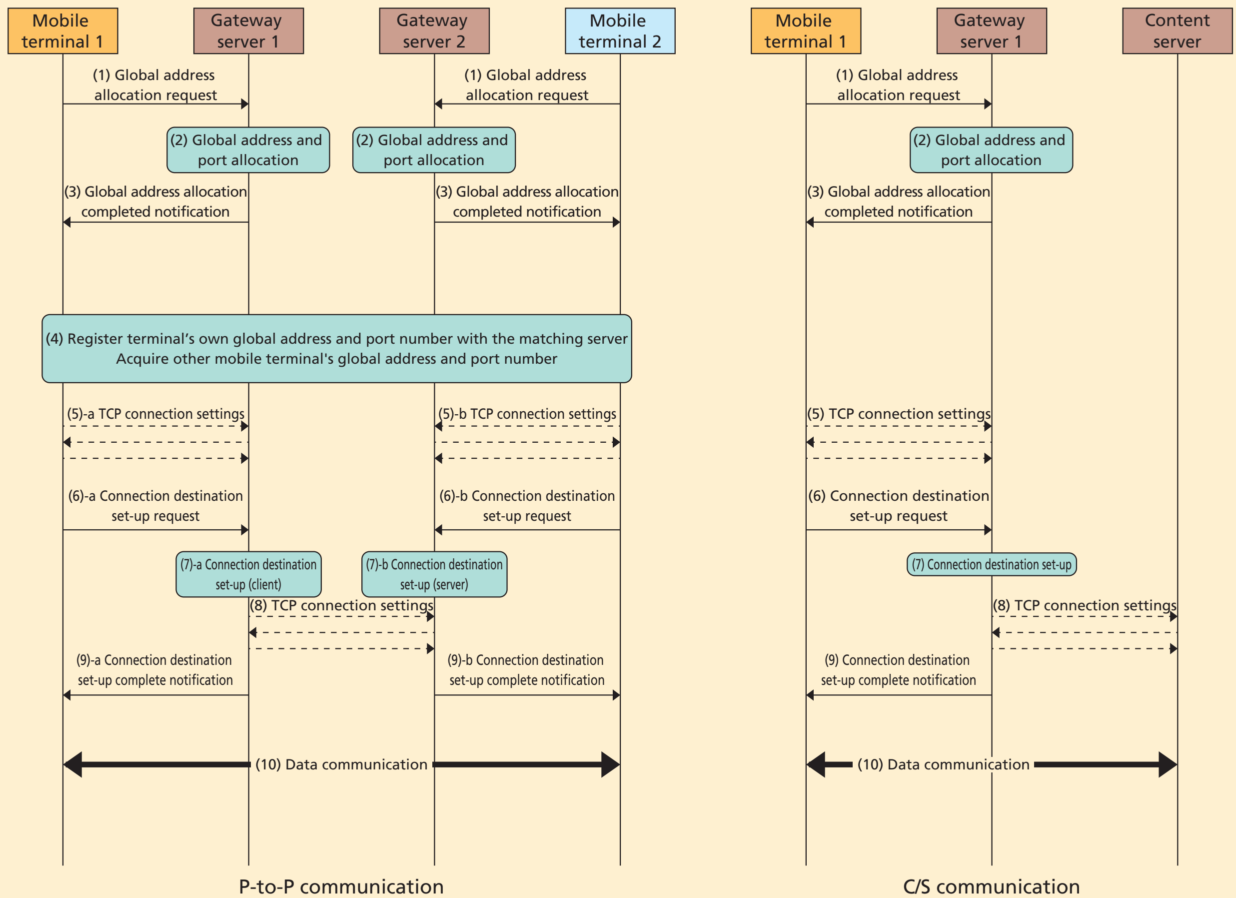


Figure 2 Sequence overview (i-appli online)

tination set-up request (Fig. 2 (6)-b) to gateway server 2 in the same way. If the connection destination set-up request from mobile terminal 2 to gateway server 2 (Fig. 2 (7)-b) is completed, the gateway server on the "client" side of the TCP connection sets up a TCP connection to the gateway server on the "server" side. When the connection has established (Fig. 2 (8)), the two gateway servers send connection set-up completed notifications to the respective mobile terminals (Fig. 2 (9)). The above procedure results in a communication path from mobile terminal 1 to mobile terminal 2 via gateway server 1 and gateway server 2 (Fig. 2 (10)).

In C/S communication, the connection destination set-up request from mobile terminal 1 (Fig. 2 (6)) contains the host name of the content server. When gateway server 1 receives the connection destination set-up request, it resolves the content server's global address from the host name and sets up the connection (Fig. 2 (7), (8)). It then sends a connection destination set-up completed notification to mobile terminal 1 (Fig. 2 (9)). The above procedure results in a communication path from mobile terminal 1 to the content server via gateway server 1 (Fig. 2 (10)).

This control message processing is executed by calls from the i-appli to the Java socket communication API, so the i-appli can be written as an ordinary socket communication program (**Figure 3**).

*14 **Matching server**: A server that has a function for communication with user terminals that access it to broker P-to-P communication among the users.

2) Port forwarding Mechanism Using the Mapping Table

To implement a proxy function that does not depend on the application protocol, we developed a port forwarding function that uses the mapping table.

After the procedure described above is completed, the information required for port forwarding has recorded in the mapping table (**Figure 4**).

The mobile terminal sends data packets to the private address (10.192.50.255) and the UDP listening port (15104) of the gateway server. The source address of the packet is the private address of the mobile terminal that was allocated by the gateway server when the call was established (10.192.50.1), and the source port is the one that was selected by the mobile terminal at the time of the address allocation request (1300). After the gateway server receives this packet, it first finds the relevant connection ID (28) by retrieving the source address and the port number of the packet. Then, the gateway server rewrites the source address and the port number in own global address (xxx.175.175.2) and port number (2006), the destination address

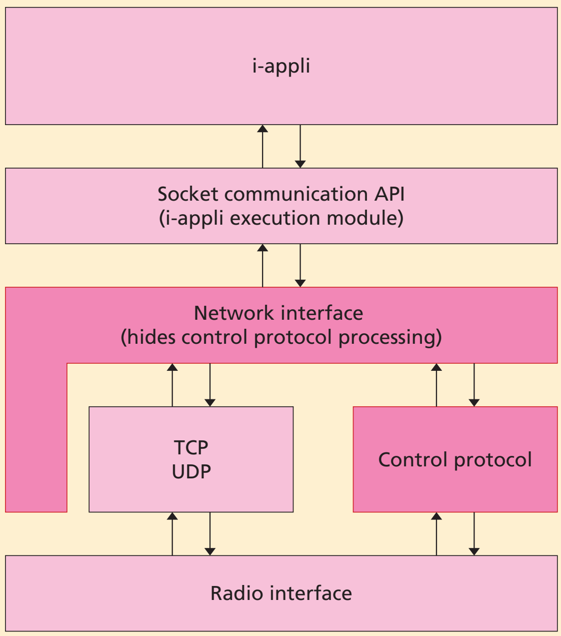
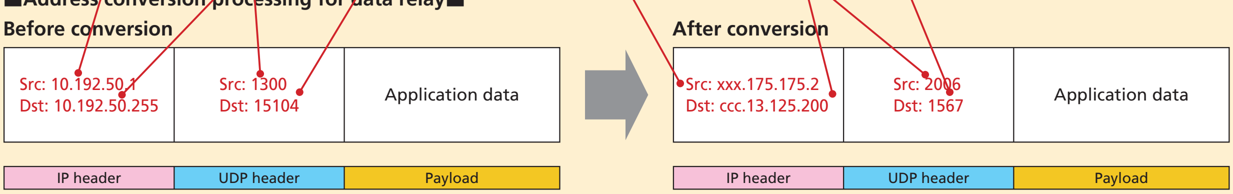


Figure 3 Function blocks in mobile terminal (i-appli online)

■Mapping table■

Mobile terminal	Mobile terminal (private address area)		Gateway server (private address area)		Gateway server (global address area)		Connection destination (global address area)		Transport protocol	Communication topology	Connection ID
	Address	Port	Address	Port	Address	Port	Address	Port			
Mobile terminal 1	10.192.50.1	1100	10.192.50.255	15103	xxx.175.175.2	2004	aaa.233.45.1	1096	TCP	C/S	25
	10.192.50.1	1200	10.192.50.255	15103	xxx.175.175.2	2005	bbb.233.45.1	1097	TCP	C/S	26
	10.192.50.1	1300	10.192.50.255	15104	xxx.175.175.2	2006	ccc.13.125.200	1567	UDP	P-to-P	28
⋮											
Mobile terminal n	10.192.50.3	1280	10.192.50.255	15103	xxx.175.175.2	2584	ddd.23.45.1	1096	TCP	P-to-P	30
	10.192.50.3	18504	10.192.50.255	15103	xxx.175.175.2	2585	eee.1.185.20	6002	TCP	C/S	49

■Address conversion processing for data relay■



Src : Source
Dst : Destination

*All values are examples only

Figure 4 Mapping table and processing for relay data

and port in the connection destination address (ccc.13.125.200) and the port (1567) with information from the mapping table. It then sends the data to the other mobile terminal. This conversion process is also applied in the other direction.

In this way, the gateway server only performs port forwarding by rewriting the packet address and the port number; it does not edit the TCP or UDP packet payload. Any application data that operates on TCP or UDP is sent to the other mobile terminal as shown in **Figure 5**. Because the mobile terminal can know the global address allocated to it by using the control messages in advance, it can use that address within the application data. This eliminates issues such as address inconsistencies in the network and application layers and allows the use of any protocol by the application.

4.2 i-appli call

To implement the i-appli call function described in Section 3.2, the invocation message of the i-appli is sent to the mobile terminal by Short Message Service (SMS)[*15].

The push server[*16] is located on the NTT DOCOMO network. Mobile terminals or servers on the Internet send a invoking request to the push server using HTTP. The push server then performs protocol conversion and notifies the target mobile terminal of the i-appli invocation request by SMS. The SMS which stores the parameters for the invocation message to JAM is sent to the mobile terminal as a control message, which is different from an ordinary short message (**Figure 6**).

Two patterns of the i-appli call functions are shown in **Figure 7**. One is a request from a mobile terminal within the NTT DOCOMO networks and the other is a request from a server on the Internet.

For invoking requests from mobile terminals, we are considering the case that the mobile terminals know the phone numbers of the mobile terminals that are invited to the online game or other such application. The i-appli calls between mobile terminals are achieved by closed message processing within NTT DOCOMO networks. Mobile terminal 1 uses HTTP to send data to the push server. The push server extracts

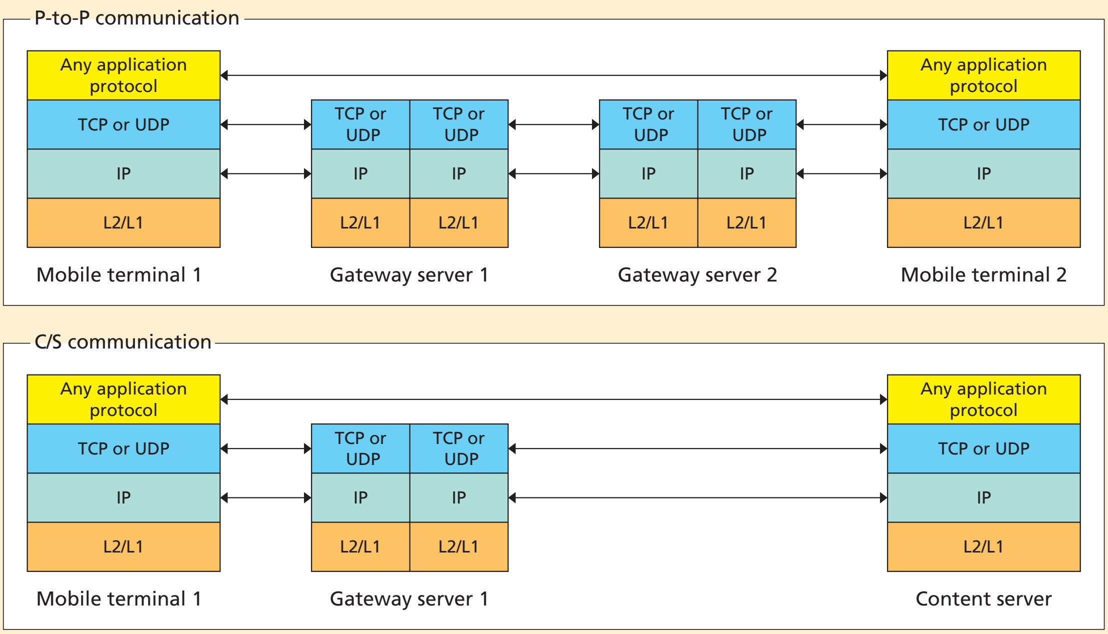


Figure 5 Protocol stack (i-appli online)

*15 **SMS**: A notification method for internal mobile terminal functions that uses a private format to store data in the message portion of an SMS (a service for sending and receiving very short text messages with a mobile terminal) message.

*16 **Push server**: A server that has functions for receiving the destination information and the parameters to be passed to the i-appli and other such required information from the source of the push request and adjusting the format so that the data can be stored in the SMS message and sent to the destination.

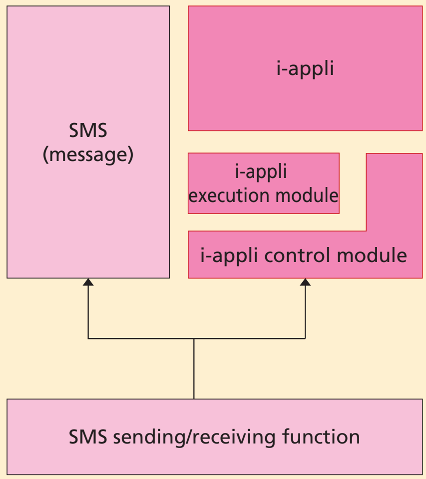


Figure 6 Function blocks in the mobile terminal (i-appli call)

the parameter values from the HTTP data and reformats them for the invoking request. The packaged data is transferred with inter-node protocol between the push server and the switching servers. The switching servers stores that data in the SMS user data area and sends it to mobile terminal 2.

For invoking requests from CP servers on the Internet, the target mobile terminals are identified by its user IDs. The user ID is an identification used in i-mode. It is related to the phone number that is used to identify

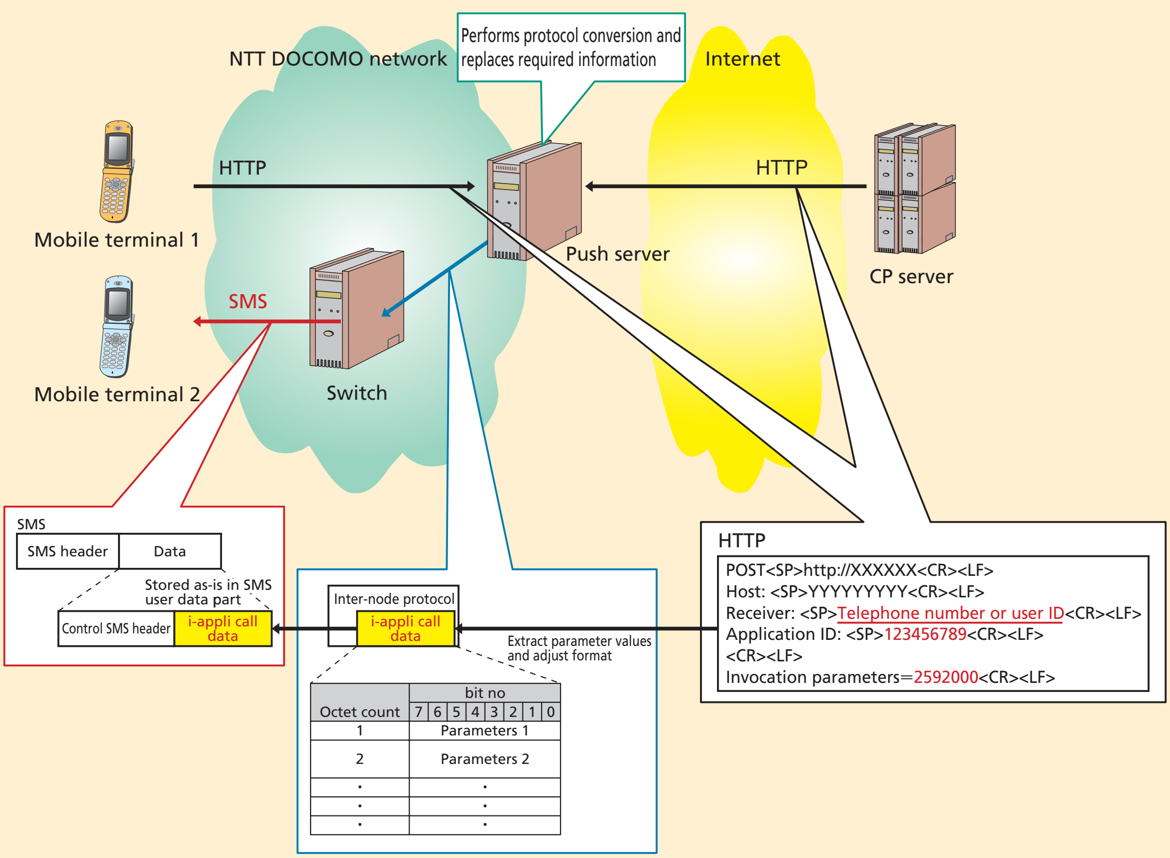


Figure 7 Network diagram (i-appli call)

users within NTT DOCOMO networks by the push server. First, the CP servers send data and the target user ID over HTTP. Next, the push server converts the user ID into the phone number and identifies the mobile terminal in the NTT DOCOMO network. Then, the data are sent to mobile terminal 2 by the same procedure as when the request comes from mobile terminal 1. This mechanism allows mobile terminal 1 to send a push invocation request that specifies the user ID of the mobile terminal 2 via the CP server.

5. Conclusion

We developed the i-appli online and the i-appli call functions with the objective of providing highly real-time application services for mobile terminals. These functions make it possible to provide online games with i-appli service. Combining these functions makes it possible to provide enterprise-oriented business applications as i-appli as well as consumer-oriented applications.

The network functions described here are not limited to i-appli. In future work, we will study use from mobile terminal applications other than i-appli and continue to expand the network functions to provide communication with an even higher real-time quality.

References

[1] M. Jinguji et. al: "Gateway Technologies —WPCG, TCPGW, ExGW—," NTT DoCoMo Technical Journal, Vol. 3, No. 3, pp. 49-61, Dec. 2001.