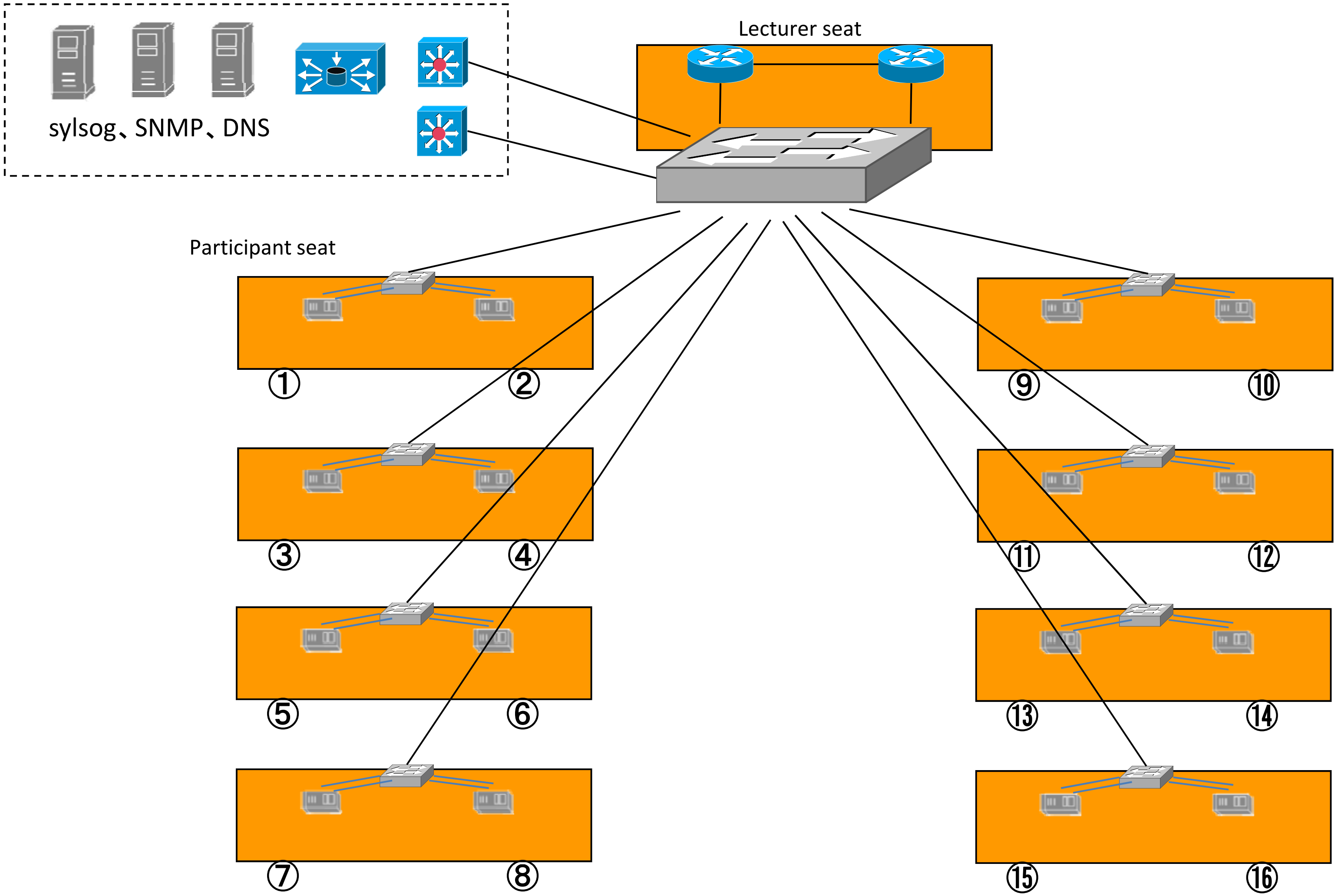
This material is translated and edited based on the
“iDC/ISP/CATV Servers Hands-On Seminar”
by Mr.Koichi Kunitake, BeaconNC Inc.,
on December 17-18, 2009,
organized by
Task Force for IPv4 Exhaustion, Japan
http://www.kokatsu.jp/

Hands-on Material

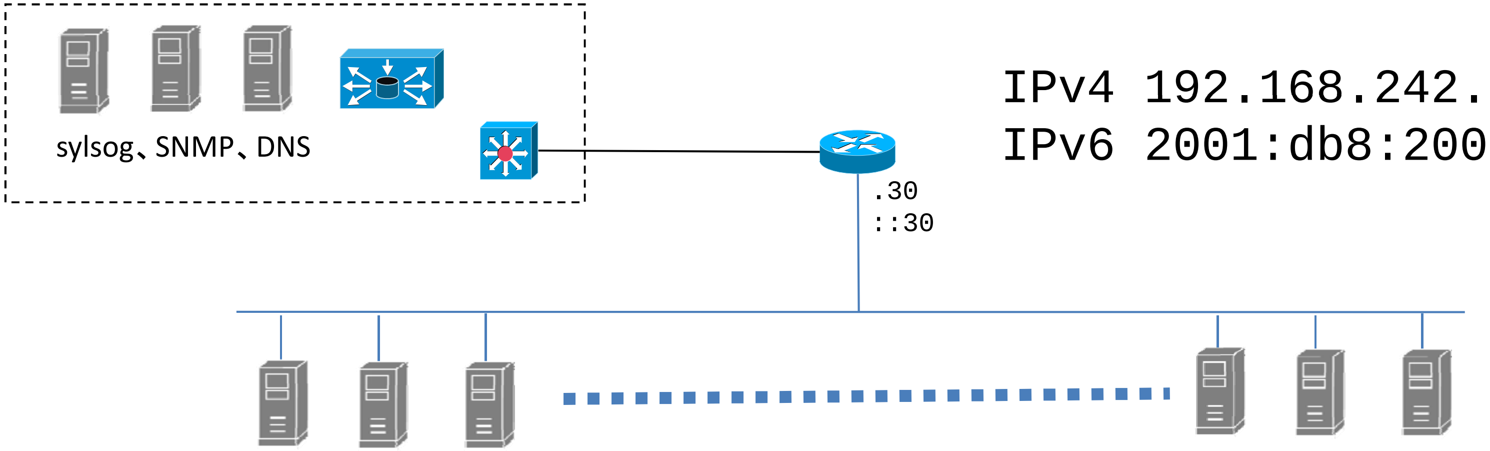
BeaconNC Inc.

Koichi Kunitake

Physical topology for the IPv6 hands-on



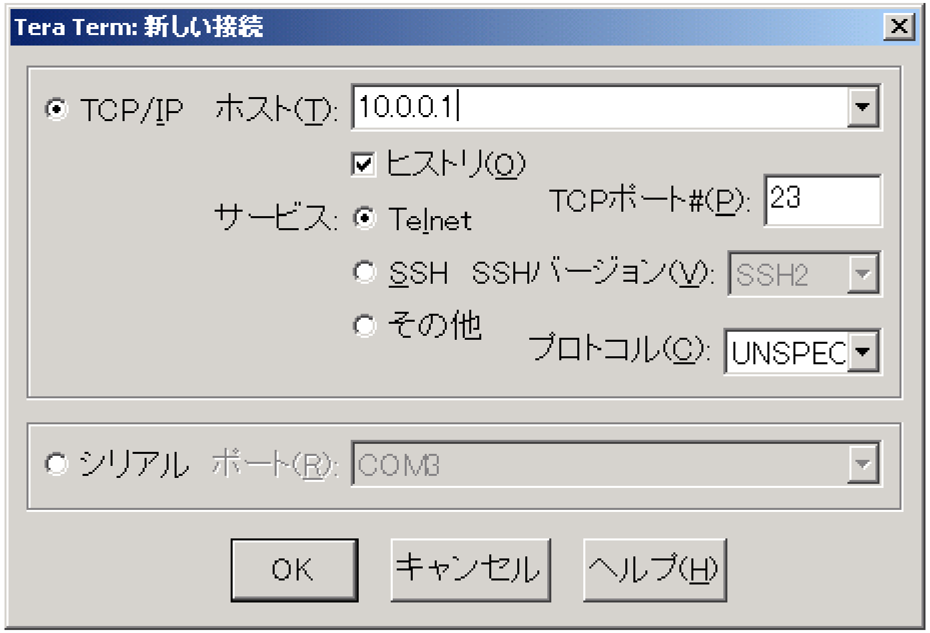
Logical topology for the IPv6 hands-on



IPv4 192.168.242.0/27
IPv6 2001:db8:2000::/64

Logging into the server

- Utilize the TeraTerm shortcut on the desktop, and connect using serial connection
- Account info: (ID/Pw)
  - admin/admin
  - root/root



Tera Term: 新しい接続

● TCP/IP ホスト(T): 10.0.0.1

☑ ヒストリ(O)

サービス: ● Telnet TCPポート#(P): 23

○ SSH SSHバージョン(V): SSH2

○ その他 プロトコル(C): UNSPEC

○ シリアル ポート(R): COM3

* You can save & copy the configuration files in USB memory if you wish.

1st day

IPv4/IPv6 address configuration

- Following addresses are assigned to this segment.
  - “192.168.242.0/27”
  - “2001:db8:2000::/64”
- Please use the following addresses (x is the seat number)
  - 192.168.242.x
  - 2001:db8:2000::x

Behavior of IPv6 address duplication

- Check the behavior when you assign duplicated address to your host
  - 2001:db8:2000::17
- Check the DAD message in syslog

IPv4/IPv6 routing configuration

- Default gw is the following
  - 192.168.242.30/27
  - 2001:db8:2000::30/64
- DNS cache server is the following
  - 2001:db8:2000:ffff::250
- Set the configuration to configuration files
- After the configuration, check the connection

```
# traceroute6 ipv6.google.co.jp
# tracepath6 www.nic.ad.jp
```

Try: DAD check

```
# ip addr add 2001:db8:2000::17/64 dev eth0
# ip addr show dev eth0
```

```
# tail /var/log/messages|grep detect
```

Try: address and routing configuration

Address configuration

```
# ip link set eth0 down && ip link set eth0 up
# ip addr add 192.168.242.x/27 dev eth0
# ip addr add 2001:db8:2000::x/64 dev eth0
```

Routing configuration

```
# ip route add 0.0.0.0/0 via 192.168.242.30 dev eth0
# ip route add ::/0 via 2001:db8:2000::30 dev eth0
```

Try: resolv.conf configuration

/etc/resolv.conf

```
search example.jp
nameserver 2001:db8:2000:ffff::250
```

Confirm the behavior

```
$ dig ipv6.google.co.jp AAAA
```

Disabling the address auto-configuration

- Disable the address auto-configuration and reboot the server. Confirm only the manually assigned address is in effect.

Disabling the address auto-configuration

```
/etc/sysconfig/network
IPV6_AUTOCONF=no
```

Activate the configuration

```
# /etc/init.d/network restart
```

Bonding configuration

- After running through the configurations, configure the bonding using eth0/eth1
- Conduct continuous ping6 to default gw and unplug the cable.

Try:Bonding configuration example

ifcfg-bond0

```
DEVICE=bond0
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=xxx:xxx:xxx::x/64
IPADDR=xx.xx.xx
NETWORK=xx.xx.xx
NETMASK=255.255.255.0
```

ifcfg-eth0

```
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

ifcfg-eth1

```
DEVICE=eth1
BOOTPROTO=none
ONBOOT=yes
MASTER=bond0
SLAVE=yes
```

Try:Bonding configuration

/etc/modprobe.conf

```
alias bond0 bonding
options bond0 mode=1 miimon=200
```

Activate the configuration

```
# /etc/init.d/network restart
```

Apache configuration

- IPv6 apache is ready as standard in RHEL5/CentOS5. Place some contents, and access from a browser.
  - Type in http://[2001:db8:2000::x]/ at the address bar
- Please check the log when you access the web site.

VirtualHost configuration

- Configure an IP address base Virtual Host and change the contents for IPv4 and IPv6 access.

Apache ACL configuration

- Let the person sitting next to you to access the web site, and check the IP address he/she was using from the access log. Configure the access denial configuration using ACL.

Try: Apache configuration

Configure following at the /etc/httpd/conf/httpd.conf

```
<VirtualHost 192.168.242.17:80>
        DocumentRoot /var/www/html/ipv4
</VirtualHost>

<VirtualHost [2001:db8:2000::17]:80>
        DocumentRoot /var/www/html/ipv6
</VirtualHost>
```

* Place some contents at /var/www/html/{ipv4|ipv6}

Try: Apache configuration check

```
$ /usr/sbin/httpd –S


$ telnet 192.168.242.17 80
GET /index.html


$ telnet 2001:db8:2000::17 80
GET /index.html
```

Mail Server

- Configure the host name and write it at /etc/hosts
- Send an e-mail to the running mail server using telnet
- Check the mail log after transmission

Try:postfix configuration

/etc/postfix/main.cf

```
myhostname = dns.17-handson.example.jp
mydomain = 17-handson.example.jp

inet_interfaces = all

mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain

home_mailbox = Maildir/
```

Creating a posting address user

```
#  useradd user1
```

Email transmission example

```
$ telnet fe80::aaa:dead:beaf%bond0  smtp
Trying fe80::aaa:dead:beaf%bond0...
Connected to fe80::aaa:dead:beaf%bond0.
Escape character is '^]'.
220 asteroid ESMTP Postfix (Ubuntu)
HELO foo
250 asteroid
MAIL FROM: kunitake@example.jp
250 2.1.0 Ok
RCPT TO: user1@17-handson.example.jp
250 2.1.5 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
From: kunitake@example.jp
Subject: from handson!
Hello IPv6 world
.
250 2.0.0 Ok: queued as 22945DD71
QUIT
221 2.0.0 Bye
Connection closed by foreign host.
```

NTP server

- Configure the NTP server
  - 2001:3a0:0:2001::27:123
  - 2001:db8:2000::x (Address of the node person next to you is using）
- Confirm the synchronization using ntpq

ACL in NTP server

```
server 2001:3a0:0:2001::27:123
server 2001:db8:2000::X

#restrict default ignore
restrict -4 default nomodify notrap
restrict -6 default nomodify notrap
```

- Confirm the IPv4 and IPv6 ACL configurations are independent by enabling / disabling the restrict configuration

DNS server

- Develop a DNS cache server that supports IPv6 transport
- Utilize the developed DNS cache server with the person sitting next to you

```
# dig @2001:db8:2000::x ipv6.google.co.jp AAAA
```

- After confirming you can access to the server, deny the query from the node of person next to you using ACL

Try:DNS cache server

```
# cd /var/named/chroot/etc
# wget ftp://ftp.rs.internic.net/domain/named.root
```

/var/named/chroot/etc/named.conf

```
acl handson-net {
        2001:db8::/32;
};
options {
        directory “/etc”;
        version “”;
        alllow-query { handson-net; 127.0.0.1; ::1; };
        listen-on-v6 {any; };
};
zone “.” {
        type hint;
        file “/etc/named.root”;
};
```

Try: DNS server

How to confirm

```
# dig @::1 ipv6.google.co.jp AAAA
```

ZONE registration

- Add the “x-handson.v4exh-testbed.jp” zone and register your server’s A and AAAA RR (x is the seat number)

Try:DNS cache server

/var/named/chroot/etc/named.conf

```
// Refer the following
zone “17-handson.example.jp” {
        type master;
        file “master/example.jp”;
        allow-transfer { localhost; handson-net;};
        allow-query { any ;};
};
```

Try: DNS authoritative Server

```
/var/named/chroot/etc/master/ipv4exh-testbed.jp
```

```
;;
$TTL 3600
@       IN      SOA     17-handson.example.jp.  root.example.jp.
 (
                2009082601      ; Serial
                7200            ; Refresh 2hrs
                1800            ; Retry 30mins
                604800          ; Expire 1 weeks
                86400   )       ; Minimum 1 days

                IN      NS      dns.17-handson.example.jp.
                IN      MX  10  dns.17-handson.example.jp.

dns             IN      A       192.168.242.17
dns             IN      AAAA    2001:db8:2000::17
```

Confirmation

$ dig @::1 17-handson.ipv4exh-testbed.jp SOA

2nd day

tcp_wrappers

- sshd access is limited. Disable the access control for the hands-on network only, and ask the person next to you to connect to the server

Try: tcp_wrappers

/etc/hosts.deny

```
ALL: ALL
```

/etc/hosts.allow

```
sshd: 192.168.242. [2001:db8:2000:ffff::]/64
```

- Make sure “SSH” is in green status at Nagios

Packet filter

- Configure the filter CERT is showing as an example using ip6tables, and see the effect (See the separate sheet: ip6tables_rules.txt)

Input

```
# Allow some ICMPv6 types in the INPUT chain
# Using ICMPv6 type names to be clear.
ip6tables -A INPUT -p icmpv6 --icmpv6-type destination-unreachable -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type packet-too-big -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type time-exceeded -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type parameter-problem -j ACCEPT

# Allow others ICMPv6 types but only if the hop limit field is 255.
ip6tables -A INPUT -p icmpv6 --icmpv6-type router-advertisement -m hl --hl-eq 255 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type neighbor-solicitation -m hl --hl-eq 255 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type neighbor-advertisement -m hl --hl-eq 255 -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type redirect -m hl --hl-eq 255 -j ACCEPT
```

Specify the ICMP required for normal behavior such as Path MTU Discovery. Specify the hoplimit explicitly to To the packets whose hoplimit have to be 255

Input（Cont)

```
# Allow some other types in the INPUT chain, but rate limit.
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-request -m limit --limit 900/min -j ACCEPT
ip6tables -A INPUT -p icmpv6 --icmpv6-type echo-reply -m limit --limit 900/min -j ACCEPT

# When there isn't a match, the default policy (DROP) will be applied.
# To be sure, drop all other ICMPv6 types.
# We're dropping enough icmpv6 types to break RFC compliance.

ip6tables -A INPUT -p icmpv6 -j LOG --log-prefix "dropped ICMPv6"
ip6tables -A INPUT -p icmpv6 -j DROP
```

Limit the Echo/reply, and log the packet drop

Output

```
# Allow ICMPv6 types that should be sent through the Internet.
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type destination-unreachable -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type packet-too-big -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type time-exceeded -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type parameter-problem -j ACCEPT


# Limit most NDP messages to the local network.
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type neighbour-solicitation -m hl --hl-eq 255 -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type neighbour-advertisement -m hl --hl-eq 255 -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type router-solicitation -m hl --hl-eq 255 -j ACCEPT
```

Specify the ICMP required for normal behavior such as Path MTU Discovery. Specify the hoplimit explicitly to To the packets whose hoplimit have to be 255

Output（Cont)

```
# If we're acting like a router, this could be a sign of problems.

ip6tables -A OUTPUT -p icmpv6 --icmpv6-type router-advertisement -j LOG --log-prefix "ra ICMPv6 type"
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type redirect -j LOG --log-prefix "redirect ICMPv6 type"
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type router-advertisement -j REJECT
ip6tables -A OUTPUT -p icmpv6 --icmpv6-type redirect -j REJECT


# Accept all other ICMPv6 types in the OUTPUT chain.

ip6tables -A OUTPUT -p icmpv6 -j ACCEPT
```

Effective configuration when used as a router

Configuration of service port, etc

```
# Enough ICMPv6!   :-D
# Some sample TCP rules. <These are for example purposes only.>
# The REJECT is for politeness on the local network.
ip6tables -A INPUT -m multiport -p tcp --dport $blocked_tcp_ports -m hl --hl-eq 255 -j REJECT
ip6tables -A OUTPUT -m multiport -p tcp --dport $blocked_tcp_ports -m hl --hl-eq 255 -j REJECT
ip6tables -A INPUT -m multiport -p tcp --dport $blocked_tcp_ports -m hl --hl-lt 255 -j DROP
ip6tables -A OUTPUT -m multiport -p tcp --dport $blocked_tcp_ports -m hl --hl-lt 255 -j DROP

# Stateful matching to allow requested traffic in.
ip6tables -A OUTPUT -p tcp -j ACCEPT
ip6tables -A OUTPUT -p udp -j ACCEPT
ip6tables -A INPUT -p tcp -m state --state ESTABLISHED,RELATED -j ACCEPT
ip6tables -A INPUT -p udp -m state --state ESTABLISHED,RELATED -j ACCEPT

# Drop NEW,INVALID probably not needed due to the default drop policy.
ip6tables -A INPUT -m state --state NEW,INVALID -j DROP
```

POP server configuration

- Receive emails using the developed POP server

Dovecot install

- Utilize yum to install dovecot

```
# yum install dovecot
```

- Configure /etc/dovecot.conf

Try:Dovecot configuration

/etc/dovecot.conf

```
protocols = imap pop3

protocol lda {
   postmaster_address = root@17-handson.example.jp
}

ssl_disable = yes

auth default {
  passdb passwd-file {
       args = /etc/dovecot.passwd
 }
  userdb passwd-file {
       args = /etc/dovecot.passwd
 }
}
```

Try: Dovecot configuration

/etc/dovecot.passwd

```
user1:{plain}user1:501:501::/home/user1::userdb_mail=maildir:/home/user1/Maildir
```

Email reception example

```
$  telnet 2001:db8:2000::17 pop3
Trying 2001:db8:2000::17...
Connected to 2001:db8:2000::17.
Escape character is '^]'.
+OK Dovecot ready.
USER user1
+OK
PASS user1
+OK Logged in.
LIST
+OK 1 messages:
1 554
.
RETR 1
+OK 554 octets
Return-Path: <kunitake@example.jp>
X-Original-To: user1@17-handson.example.jp
Delivered-To: user1@17-handson.example.jp
Received: from fo (localhost.localdomain [127.0.0.1])
        by dns.17-handson.example.jp (Postfix) with SMTP id 092E862C109
        for <user1@17-handson.example.jp>; Wed, 16 Dec 2009 15:29:12 +0900 (JST)
From: kunitake@example.jp
Subject: test mail
Message-Id: <20091216062926.092E862C109@dns.17-handson.example.jp>
Date: Wed, 16 Dec 2009 15:29:12 +0900 (JST)
To: undisclosed-recipients:;

Hello
```