NAT Traversal with GnuGk

Step 1 (Optional)

This step lets us to detect whether we are behind some kind of NAT or not. During endpoint registration process, we fill the **nonStandardData** RRQ field with our local IP address (in the form of a string „**IP=...**"). This will let the gatekeeper to detect that we are beind a NAT box even if the NAT box will replace private addresses with public addresses. This is optional to send this information and not doing so will not affect remaining steps.

Sample RRQ message (fields that are not important are omitted):

```
registrationRequest {
    requestSeqNum = 35543
    protocolIdentifier = 0.0.8.2250.0.4
    nonStandardData = {
      nonStandardIdentifier = object
      data =  17 octets {
        49 50 3d 31 39 32 2e 31  36 38 2e 31 35 2e 31 33   IP=192.168.15.13
        34                                                 4
      }
    }
    discoveryComplete = TRUE
    callSignalAddress = 1 entries {
      [0]=ipAddress {
        ip =  4 octets {
          c0 a8 0f 86                                        ....
        }
        port = 1720
      }
    }
    rasAddress = 1 entries {
      [0]=ipAddress {
        ip =  4 octets {
          c0 a8 0f 86                                        ....
        }
        port = 1256
      }
    }
  }
```

When GnuGk detects that an endpoint is behind NAT (by comparing the address an RRQ has been received from with either the address passed in the nonStandardData field or with addresses passed in the rasAddress RRQ field), it includes a **nonStandardData** field in an RCF message with „**NAT=...**" string embedded. This is a trigger for the endpoint to use the NAT traversal method. The nonStandardData „NAT=" field tells us that we are behind a NAT box and the gatekeeper (GnuGk) supports NAT traversal.

Sample RCF message (fields that are not important are omitted):

```
registrationConfirm {
  requestSeqNum = 35543
  protocolIdentifier = 0.0.8.2250.0.4
  nonStandardData = {
    nonStandardIdentifier = object
    data =  18 octets {
      4e 41 54 3d 32 31 33 2e  31 37 2e xx xx xx xx xx   NAT=213.17.xxx.x
      xx xx                                              xx
    }
  }
  callSignalAddress = 1 entries {
    [0]=ipAddress {
      ip =  4 octets {
        xx xx xx xx                                        xxxx
      }
      port = 1720
    }
  }
  endpointIdentifier =  8 characters {
    0036 0037 0036 0038 005f 0067 006b 0033   6768_gk3
  }
}
```

Step 2

After we have detected that we are behind a NAT box and the gatekeeper supports NAT traversal (either by RRQ/RCF exchange or by manual settings in the application), we start **a new TCP connection to the gatekeeper signaling port**. This connection is like a regular signaling connection, with an exception that the first message is not Q.931 Setup, but **Q.931 Information**. We send an Information message to tell the gatekeeper that this is not a call, but a reverse signaling channel used to reach an endpoint behind a NAT box. This Information message contains an **endpoint identifier** passed in Q.931 Facility Information Element. Thanks to this information, the gatekeeper knows that this NAT connection should be used to contact this particular endpoint. Also, a **Call State** Information Element has to be included and set to 9.

Sample Q.931 Information message (NAT connect):

```
protocolDiscriminator = 8
callReference = 0
from = originator
messageType = Information
IE: Call-State = {
      09                                                .
}
IE: Facility = {
      36 37 36 38 5f 67 6b 33                           6768_gk3
}
```

This message should be resent periodically to maintain NAT port mapping alive and TCP connection alive.

When a new call to such endpoint hits the gatekeeper, the gatekeeper detects presence of reverse signaling connection and chooses this TCP connection (instead of creating a new one) to handle call signaling. Since this moment, both the endpoint and the gatekeeper should treat this connection like a normal signaling connection. Therefore it should be closed after call hungup. The endpoint detects a new incoming call when a Q.931 Setup message is received on the NAT connection. The gatekeeper will not send other messages prior to the Setup. After the Setup is received, the endpoint should establish a new NAT connection, as the connection currently used becomes a normal call signaling connection.

Step 3

This step is executed when an endpoint unregisters itself from the gatekeeper. It can either simply close the NAT traversal connection or send a special message prior to connection closure.

Sample Q.931 Information message (NAT disconnect):

```
protocolDiscriminator = 8
callReference = 0
from = originator
messageType = Information
IE: Call-State = {
  0b                                                    .
}
IE: Facility = {
  36 37 36 38 5f 67 6b 33                               6768_gk3
}
```

This message contains an endpoint identifier and 0x0b value in the Call State IE to signal disconnect request. After this message is sent, the gatekeeper will close the TCP connection.