



# OSP Toolkit

## Cisco Interoperability

Release 3.1.2

1 July, 2004

## Revision History

Revision	Date of Issue	Changes
2.7.0	March 12 <sup>th</sup> , 2003	Added revision history. Updated Appendix 'A' – OSP Authorization Token Header for SIP, with the latest RFC.
2.8.0	March 20 <sup>th</sup> , 2003	No Changes
2.8.1	March 25 <sup>th</sup> , 2003	No Changes
2.8.2	April 9 <sup>th</sup> , 2003	No Changes
2.9.0	June 1 <sup>st</sup> , 2003	No Changes
2.9.1	July 7 <sup>th</sup> , 2003	No Changes
2.9.2	July 28 <sup>th</sup> , 2003	No Changes
2.9.3	Sep 15 <sup>th</sup> , 2003	No Changes
2.11.1	Feb 12 <sup>th</sup> , 2004	No Changes
3.0	March 11 <sup>th</sup> , 2004	Updated the SIP Token header RFC with the latest draft. Rectified a mistake in the OSP Authorization Response message.
3.1	April 8 <sup>th</sup> , 2004	No changes
3.1.1	May 12 <sup>th</sup> , 2004	No Changes
3.1.2	July 1 <sup>st</sup> , 2004	No Changes

**Contents**

Revision History .....	2
Contents .....	3
Introduction .....	4
Test Network and Example Call Scenario .....	4
Detailed Message Examples.....	5
Step 1. OSP AuthorizationRequest .....	5
Step 2. OSP AuthorizationResponse .....	5
Step 3A. Q.931 Call Set-up with token .....	6
Step 3B. Q.931 messages from destination gateway .....	12
Step 4A. OSP UsageIndication from Source .....	18
Step 4B. OSP UsageIndication from Destination .....	19
Step 5A. OSP Confirmation to Source .....	19
Step 5B. OSP Confirmation to Destination.....	20
OSP Token Format.....	20
Cisco OSP Token .....	20
OSP token format for SIP .....	22
Appendix A: <i>OSP Authorization Token Header for SIP</i> .....	23

E-mail: [support@transnexus.com](mailto:support@transnexus.com)  
www.transnexus.com  
Copyright © 2003 by TransNexus. All Rights Reserved.

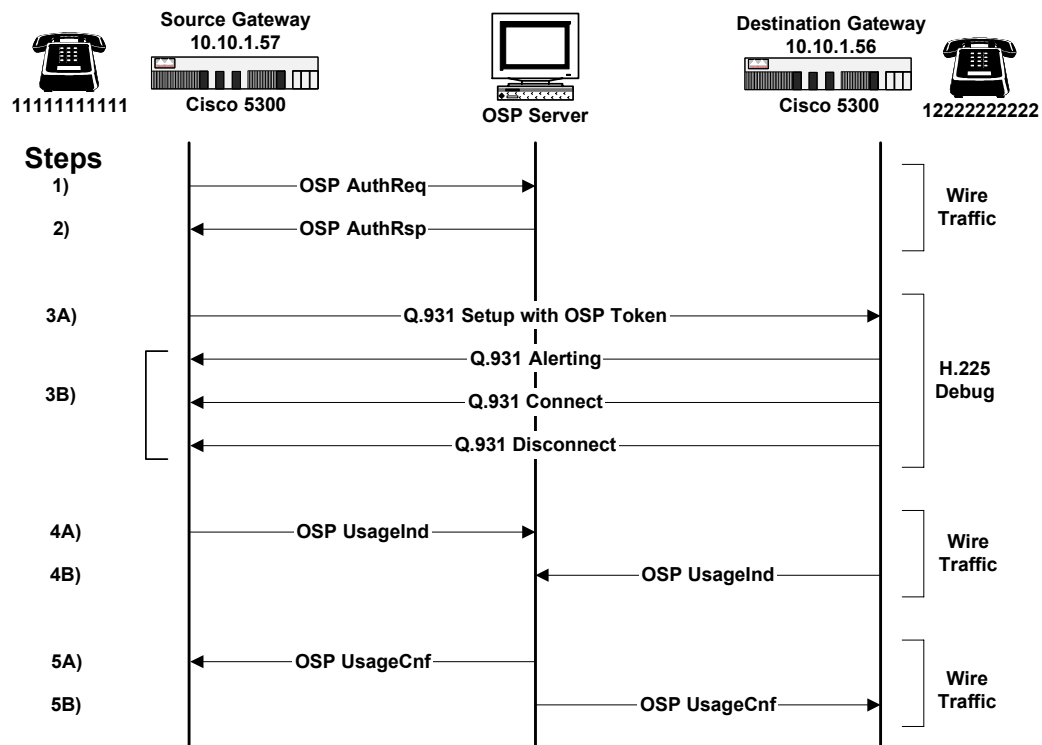
## Introduction

The purpose of this document is to provide an example of how the OSP standard has been implemented in generally available products. Cisco's OSP implementation has been selected for this example. Cisco Systems is one of the co-developers of the OSP standard and the leading provider of VoIP gateways worldwide.

The document begins with a description of the testbed network used to create the example call scenario. The second section describes the messages exchanged among the OSP Server and gateways for the example call scenario. The final section provides information on how the OSP token must be formatted in call set-up message for interoperability with Cisco devices.

## Test Network and Example Call Scenario

The following diagram illustrates the test network used to create the example call scenario. The VoIP devices used are two Cisco AS5300 H.323 gateways. The source gateway has address 10.10.1.57, and the destination gateway has address 10.10.1.56. Both gateways are running IOS version 12.2(3), and have been enrolled with an OSP Nexus Server. The call scenario begins with a call made from the telephone connected to the source gateway; The calling number is 11111111111, and the called number is 12222222222.



Step 1: Source gateway sends OSP AuthorizationRequest to OSP Server requesting IP address of gateways that can complete the call to the called number 12222222222.

Step 2: OSP Server sends OSP AuthorizationResponse to source gateway with IP address of the destination gateway and an authorization token.

Step 3: Source gateway sends Q.931 call set-up message to destination gateway. Call is completed and disconnected after 21 seconds.

Step 4: Both the source and destination gateways send OSP UsageIndication messages, reporting call duration, to the OSP server.

Step 5: OSP Server sends OSP UsageConfirmation to both source and destination gateways.

## Detailed Message Examples

This section provides detailed examples of the five steps in the H.323 call scenario described above. Steps 1, 2, 4 and 5 are OSP messages captured off the wire using a protocol analyzer. Steps 3A and 3B are examples of the Q.931 messages exchanged between the source and destination gateways. These messages were captured from H.225 debug out from the Cisco gateways.

### Step 1. OSP AuthorizationRequest

From source gateway 10.10.1.57 to OSP Nexus Server.

```
<Message messageId="23711750491" random="4747">
  <AuthorisationRequest componentId="2371175042768">
    <Timestamp>2001-10-03T22:05:12Z</Timestamp>
    <CallId encoding="base64">jPm35LeBEdWALdS27YGXCQ==</CallId>
    <SourceInfo type="e164">11111111111</SourceInfo>
    <DestinationInfo type="e164">12222222222</DestinationInfo>
    <Service/>
    <MaximumDestinations>3</MaximumDestinations>
  </AuthorisationRequest>
</Message>
```

### Step 2. OSP AuthorizationResponse

From the OSP Nexus Server to source gateway 10.10.1.57.

```
<Message messageId='23711750491' random='8946'>
  <AuthorisationResponse componentId='2371175042768'>
    <Timestamp>2001-10-03T22:05:12Z</Timestamp>
    <Status>
      <Description>SUCCESS</Description>
      <Code>200</Code>
    </Status>
    <TransactionId>4304187353840953091</TransactionId>
    <Destination>
      <CallId encoding='base64'>jPm35LeBEdWALdS27YGXCQ==</CallId>
      <DestinationInfo type='e164'>12222222222</DestinationInfo>
      <DestinationSignalAddress>[10.10.1.56]</DestinationSignalAddress>
      <Token encoding='base64'>
```



```
Oct 3 22:05:12.405: compose_new_style_settlement_token: Building
standard settlement token.
Oct 3 22:05:12.405: H225.0 OUTGOING PDU ::=

value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body setup :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      sourceInfo
      {
        gateway
        {
          protocol
          {
            voice :
            {
              supportedPrefixes
              {
                }
              }
            }
          }
          mc FALSE
          undefinedNode FALSE
        }
        activeMC FALSE
        conferenceID '8CF9B7E4B78111D5802CD4B6ED819709'H
        conferenceGoal create : NULL
        callType pointToPoint : NULL
        sourceCallSignalAddress ipAddress :
        {
          ip '0A0A0139'H
          port 11011
        }
        callIdentifier
        {
          guid '8CF9B7E4B78111D5802DD4B6ED819709'H
        }
        tokens
        {
          {
            tokenOID { 0 4 0 1321 1 2 }
            nonStandard
            {
              nonStandardIdentifier { 0 4 0 1321 1 2 }
              data '308203E106092A864886F70D010702A08203D230...'H
            }
          }
        }
        fastStart
        {
```





```
h323-message-body callProceeding :
{
  protocolIdentifier { 0 0 8 2250 0 2 }
  destinationInfo
  {
    mc FALSE
    undefinedNode FALSE
  }
  callIdentifier
  {
    guid '8CF9B7E4B78111D5802DD4B6ED819709'H
  }
  fastStart
  {
    '0000000D40018011140001000A0A013843D8000A...'H,
    '400000060401004D40018011140001000A0A0139...'H
  }
}
h245Tunneling FALSE
}
```

h323chan\_chn\_process\_read\_socket: fd (1) of type CONNECTED has data

Hex representation of the received  
TPKT030000320802800B013401017E0023052380060008914A000200048011008CF9B7E  
4B78111D5802DD4B6ED81970906800100

Oct 3 22:05:12.973: h225ParseData: Q.931 ALERTING received on socket  
[1]

Oct 3 22:05:12.973: H225.0 INCOMING ENCODE BUFFER ::= 23 80060008  
914A0002 00048011 008CF9B7 E4B78111 D5802DD4 B6ED8197 09068001 00

Oct 3 22:05:12.977:

Oct 3 22:05:12.977: H225.0 INCOMING PDU ::=

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body alerting :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      destinationInfo
      {
        mc FALSE
        undefinedNode FALSE
      }
      callIdentifier
      {
        guid '8CF9B7E4B78111D5802DD4B6ED819709'H
      }
    }
    h245Tunneling FALSE
  }
}
```

h323chan\_chn\_process\_read\_socket: fd (1) of type CONNECTED has data

Hex representation of the received

```
TPKT0300004B0802800B0704038090A37E003A052280060008914A00020880013C05010
0008CF9B7E4B78111D5802CD4B6ED819709090011008CF9B7E4B78111D5802DD4B6ED81
970906800100
```

Oct 3 22:05:13.173: h225ParseData: Q.931 CONNECT received on socket  
[1]

```
Oct 3 22:05:13.173: H225.0 INCOMING ENCODE BUFFER ::= 22 80060008
914A0002 0880013C 05010000 8CF9B7E4 B78111D5 802CD4B6 ED819709 09001100
8CF9B7E4 B78111D5 802DD4B6 ED819709 06800100
```

Oct 3 22:05:13.177:

Oct 3 22:05:13.177: H225.0 INCOMING PDU ::=

value H323\_UserInformation ::=

```
{
  h323-uu-pdu
  {
    h323-message-body connect :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      destinationInfo
      {
        gateway
        {
          protocol
          {
            voice :
            {
              supportedPrefixes
              {
            }
          }
        }
      }
      mc FALSE
      undefinedNode FALSE
    }
    conferenceID '8CF9B7E4B78111D5802CD4B6ED819709'H
    callIdentifier
    {
      guid '8CF9B7E4B78111D5802DD4B6ED819709'H
    }
  }
  h245Tunneling FALSE
}
}
```

Oct 3 22:05:13.201: %ISDN-6-CONNECT: Interface Serial0:0 is now  
connected to 1111111111

Oct 3 22:05:19.201: %ISDN-6-CONNECT: Interface Serial0:0 is now

```
connected to 1111111111
Oct  3 22:05:34.722: %ISDN-6-DISCONNECT: Interface Serial0:0
disconnected from 1111111111 , call lasted 21 seconds
Oct  3 22:05:34.726: H225.0 OUTGOING PDU ::=

value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body releaseComplete :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      callIdentifier
      {
        guid '8CF9B7E4B78111D5802DD4B6ED819709'H
      }
    }
    h245Tunneling FALSE
  }
}

Oct  3 22:05:34.730: H225.0 OUTGOING ENCODE BUFFER::= 25 80060008
914A0002 0111008C F9B7E4B7 8111D580 2DD4B6ED 81970906 800100
Oct  3 22:05:34.730:
Hex representation of the RELEASE COMPLETE TPKT to
send.030000310802000B5A080280907E0021052580060008914A00020111008CF9B7E4
B78111D5802DD4B6ED81970906800100
Oct  3 22:05:34.730: h225TerminateRequest: Q.931 RELEASE COMPLETE sent
from socket [1]. Call state changed to [Null].
Oct  3 22:05:34.730:          h323chan_close: TCP connection from socket
[1] closed
```

### Step 3B. Q.931 messages from destination gateway

Cisco H.225 debug output from destination gateway 10.10.1.56.

H.225 Event Messages debugging is on

H.225 ASN1 Messages debugging is on

```
GATEWAY2#h323chan_chn_process_read_socket: fd (0) of type LISTENING has
data
Changing to new event: ACCEPT
h323chan_chn_accept: 0

Oct  3 22:05:12.397:          h323chan_gw_accept: TCP connection accepted
from 10.10.1.57:11011 on socket [1]
Oct  3 22:05:12.397: local(0x0) accepts TCP conn from
10.10.1.57(0xA0A0139) port 11011changing from LISTENING state to
ACCEPTED state
h323chan_chn_process_read_socket: fd (1) of type ACCEPTED has data

Hex representation of the received
```



```
00A08201 39308201 353081E0 02010130 0D06092A 864886F7 0D010104 05003026
3110300E 06035504 03130762 65746162 65313112 30100603 55040A13 094F5350
53657276 6572301E 170D3031 30393230 31313132 30365A17 0D303330 39323131
31313230 365A3026 3110300E 06035504 03130762 65746162 65313112 30100603
55040A13 094F5350 53657276 6572305C 300D0609 2A864886 F70D0101 01050003
4B003048 024100F1 9E1B0578 D442215F 48C40C52 D15D0843 12BE7439 440FABE7
F9E55C10 7744D036 DB34E813 F6F7E522 35D456DD CDC84477 59709742 D4283AF0
C13D7207 AAD21F02 03010001 300D0609 2A864886 F70D0101 04050003 410066FC
45A7FF00 AB8CA823 03D2BAF6 5271A334 139AB20E 82FF0EEA A74145A2 66C67402
B5F56EAD 4D2EF1A8 28F5B7FF 90792152 1FDAFE5B 4DC698EE C7D7C890 F4F23181
9230818F 02010130 2B302631 10300E06 03550403 13076265 74616265 31311230
10060355 040A1309 4F535053 65727665 72020101 300C0608 2A864886 F70D0205
0500300D 06092A86 4886F70D 01010105 0004404C 76E48020 EB109186 347C25FD
C05C3818 9B23E900 92DCF91E 034FF2CE 4FADF0AD 1418753E 3A93C4FB 78B4470F
08512D17 D23D71DD 1D511B82 335E0B22 D4305F32 02120000 000D4001 800A0400
01000A0A 013946E9 1D400000 06040100 4D400180 11140001 000A0A01 3946E800
0A0A0139 46E90100 01000680 0100
Oct 3 22:05:12.497:
Oct 3 22:05:12.497: H225.0 INCOMING PDU ::=
```

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body setup :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      sourceInfo
      {
        gateway
        {
          protocol
          {
            voice :
            {
              supportedPrefixes
              {
            }
          }
        }
      }
      mc FALSE
      undefinedNode FALSE
    }
    activeMC FALSE
    conferenceID '8CF9B7E4B78111D5802CD4B6ED819709'H
    conferenceGoal create : NULL
    callType pointToPoint : NULL
    sourceCallSignalAddress ipAddress :
    {
      ip '0A0A0139'H
      port 11011
    }
    callIdentifier
    {
```

```
    guid '8CF9B7E4B78111D5802DD4B6ED819709'H
  }
  tokens
  {
    {
      tokenOID { 0 4 0 1321 1 2 }
      nonStandard
      {
        nonStandardIdentifier { 0 4 0 1321 1 2 }
        data '308203E106092A864886F70D010702A08203D230...'H
      }
    }
  }
  fastStart
  {
    '0000000D4001800A040001000A0A013946E9'H,
    '400000060401004D40018011140001000A0A0139...'H
  }
  mediaWaitForConnect FALSE
  canOverlapSend FALSE
}
h245Tunneling FALSE
}
```

Oct 3 22:05:12.505: parse\_ClearTokenNonstd: Decoding settlement clear token using standard format, len=997

Oct 3 22:05:12.541: H225.0 OUTGOING PDU ::=

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body callProceeding :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      destinationInfo
      {
        mc FALSE
        undefinedNode FALSE
      }
      callIdentifier
      {
        guid '8CF9B7E4B78111D5802DD4B6ED819709'H
      }
      fastStart
      {
        '0000000D40018011140001000A0A013843D8000A...'H,
        '400000060401004D40018011140001000A0A0139...'H
      }
    }
  }
  h245Tunneling FALSE
}
```

```
    }
  }

Oct  3 22:05:12.545: H225.0 OUTGOING ENCODE BUFFER ::= 21 80060008
914A0002 00048811 008CF9B7 E4B78111 D5802DD4 B6ED8197 09390219 0000000D
40018011 14000100 0A0A0138 43D8000A 0A013843 D91D4000 00060401 004D4001
80111400 01000A0A 013946E8 000A0A01 3843D906 800100
Oct  3 22:05:12.549:
Hex representation of the CALL PROCEEDING TPKT to
send.030000690802800B027E005D052180060008914A000200048811008CF9B7E4B781
11D5802DD4B6ED8197093902190000000D40018011140001000A0A013843D8000A0A013
843D91D400000060401004D40018011140001000A0A013946E8000A0A013843D9068001
00
Oct  3 22:05:12.549: h225CallProcRequest: Q.931 CALL PROCEEDING sent
fromsocket [1].
Oct  3 22:05:12.961: %ISDN-6-CONNECT: Interface Serial0:22 is now
connected to 12222222222
Oct  3 22:05:12.961: H225.0 OUTGOING PDU ::=

value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body alerting :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      destinationInfo
      {
        mc FALSE
        undefinedNode FALSE
      }
      callIdentifier
      {
        guid '8CF9B7E4B78111D5802DD4B6ED819709'H
      }
    }
    h245Tunneling FALSE
  }
}
```

```
Oct  3 22:05:12.965: H225.0 OUTGOING ENCODE BUFFER ::= 23 80060008
914A0002 00048011 008CF9B7 E4B78111 D5802DD4 B6ED8197 09068001 00
Oct  3 22:05:12.965:
Hex representation of the ALERTING TPKT to
send.030000320802800B013401017E0023052380060008914A000200048011008CF9B7
E4B78111D5802DD4B6ED81970906800100
Oct  3 22:05:12.965: h225AlertRequest: Q.931 ALERTING sent from socket
[1]. Call state changed to [Call Received].
Oct  3 22:05:12.973: H225.0 OUTGOING PDU ::=

value H323_UserInformation ::=
```

```
{
  h323-uu-pdu
  {
    h323-message-body connect :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      destinationInfo
      {
        gateway
        {
          protocol
          {
            voice :
            {
              supportedPrefixes
              {
                }
              }
            }
          }
          mc FALSE
          undefinedNode FALSE
        }
        conferenceID '8CF9B7E4B78111D5802CD4B6ED819709'H
        callIdentifier
        {
          guid '8CF9B7E4B78111D5802DD4B6ED819709'H
        }
      }
      h245Tunneling FALSE
    }
  }
}
```

```
Oct 3 22:05:12.977: H225.0 OUTGOING ENCODE BUFFER::= 22 80060008
914A0002 0880013C 05010000 8CF9B7E4 B78111D5 802CD4B6 ED819709 09001100
8CF9B7E4 B78111D5 802DD4B6 ED819709 06800100
```

```
Oct 3 22:05:12.977:
```

```
Hex representation of the CONNECT TPKT to
```

```
send.0300004B0802800B0704038090A37E003A052280060008914A00020880013C0501
00008CF9B7E4B78111D5802CD4B6ED819709090011008CF9B7E4B78111D5802DD4B6ED8
1970906800100
```

```
Oct 3 22:05:12.981: h225SetupResponse: Q.931 CONNECT sent from socket
[1]
```

```
Oct 3 22:05:18.961: %ISDN-6-CONNECT: Interface Serial0:22 is now
connected to 1222222222 h323chan_chn_process_read_socket: fd (1) of
type ACCEPTED has data
```

```
Hex representation of the received
```

```
TPKT030000310802000B5A080280907E0021052580060008914A00020111008CF9B7E4B
78111D5802DD4B6ED81970906800100
```

```
Oct 3 22:05:34.725: h225ParseData: Q.931 RELEASE COMPLETE received on
socket [1]
```

```
Oct 3 22:05:34.729: H225.0 INCOMING ENCODE BUFFER::= 25 80060008
```

```
914A0002 0111008C F9B7E4B7 8111D580 2DD4B6ED 81970906 800100
Oct 3 22:05:34.729:
Oct 3 22:05:34.729: H225.0 INCOMING PDU ::=
```

```
value H323_UserInformation ::=
{
  h323-uu-pdu
  {
    h323-message-body releaseComplete :
    {
      protocolIdentifier { 0 0 8 2250 0 2 }
      callIdentifier
      {
        guid '8CF9B7E4B78111D5802DD4B6ED819709'H
      }
    }
    h245Tunneling FALSE
  }
}
```

```
h323chan_chn_process_read_socket: fd (1) of type ACCEPTED has data
```

```
Oct 3 22:05:34.733: h323chan_recvdata:Connection lost socket [1]
Oct 3 22:05:34.733: h323chan_close: TCP connection from socket
[1] closedh225TerminateRequest: Unidentifiable socket -1
Oct 3 22:05:34.753: %ISDN-6-DISCONNECT: Interface Serial0:22
disconnected from 1222222222 , call lasted 21 seconds
```

#### Step 4A. OSP UsageIndication from Source

From source gateway, 10.10.1.57, to OSP Nexus Server.

```
<Message messageId="2597352024114" random="4809">
  <UsageIndication componentId="2597352025057">
    <Timestamp>2001-10-03T22:05:34Z</Timestamp>
    <Role>source</Role>
    <TransactionId>4304187353840953091</TransactionId>
    <CallId encoding="base64">jPm35LeBEdWALdS27YGXCQ==</CallId>
    <SourceInfo type="e164">1111111111</SourceInfo>
    <SourceAlternate type="transport">[10.10.1.57]</SourceAlternate>
    <DestinationInfo type="e164">1222222222</DestinationInfo>
    <DestinationAlternate type="transport">[10.10.1.56]</DestinationAlternate>
    <UsageDetail>
      <Service/>
      <Amount>21</Amount>
      <Increment>1</Increment>
      <Unit>s</Unit>
    </UsageDetail>
    <gric.com:TransactionStartTime critical="False">2001-10-03T22:05:13Z
      </gric.com:TransactionStartTime>
    <gric.com:TransactionStatus critical="False">
      <gric.com:TransactionCode critical="False">1016
        </gric.com:TransactionCode>
```

```

<gric.com:Description critical="False">normal call clearing
                                </gric.com:Description>
</gric.com:TransactionStatus>
<transnexus.com:Statistics critical="False">
  <transnexus.com:LossSent critical="False">
    <transnexus.com:Packets critical="False">0</transnexus.com:Packets>
    <transnexus.com:Fraction critical="False">0</transnexus.com:Fraction>
  </transnexus.com:LossSent>
  <transnexus.com:LossReceived critical="False">
    <transnexus.com:Packets critical="False">0</transnexus.com:Packets>
    <transnexus.com:Fraction critical="False">0</transnexus.com:Fraction>
  </transnexus.com:LossReceived>
</transnexus.com:Statistics>
</UsageIndication>
</Message>

```

#### Step 4B. OSP UsageIndication from Destination

From destination gateway, 10.10.1.56, to OSP Nexus Server.

```

<Message messageId="2597369534664" random="2857">
  <UsageIndication componentId="2597369532132">
    <Timestamp>2001-10-03T22:05:34Z</Timestamp>
    <Role>destination</Role>
    <TransactionId>4304187353840953091</TransactionId>
    <CallId encoding="base64">jPm35LeBEdWALdS27YGXCQ==</CallId>
    <SourceInfo type="e164">1111111111</SourceInfo>
    <SourceAlternate type="transport">[10.10.1.57]</SourceAlternate>
    <DestinationInfo type="e164">1222222222</DestinationInfo>
    <DestinationAlternate
type="transport">[10.10.1.56]</DestinationAlternate>
    <UsageDetail>
      <Service/>
      <Amount>21</Amount>
      <Increment>1</Increment>
      <Unit>s</Unit>
    </UsageDetail>
    <gric.com:TransactionStartTime critical="False">
      2001-10-03T22:05:12Z</gric.com:TransactionStartTime>
    <gric.com:TransactionStatus critical="False">
      <gric.com:TransactionCode critical="False">1016
                                </gric.com:TransactionCode>
    <gric.com:Description critical="False">normal call clearing
                                </gric.com:Description>
  </gric.com:TransactionStatus>
</UsageIndication>
</Message>

```

#### Step 5A. OSP Confirmation to Source

Acknowledgement message from OSP Nexus Server to source gateway 10.10.1.57.

```

<Message messageId='2597352024114' random='18941'>
  <UsageConfirmation componentId='2597352025057'>

```

```

    <Timestamp>2001-10-03T22:05:34Z</Timestamp>
    <Status>
      <Description>SUCCESS</Description>
      <Code>200</Code>
    </Status>
  </UsageConfirmation>
</Message>

```

### Step 5B. OSP Confirmation to Destination

Acknowledgement message from OSP Nexus Server to destination gateway, 10.10.1.56.

```

<Message messageId='2597369534664' random='18941'>
  <UsageConfirmation componentId='2597369532132'>
    <Timestamp>2001-10-03T22:05:34Z</Timestamp>
    <Status>
      <Description>SUCCESS</Description>
      <Code>200</Code>
    </Status>
  </UsageConfirmation>
</Message>

```

## OSP Token Format

OSP interoperability between different VoIP devices is dependent on the formatting of the OSP token passed in the H.323 call set-up or SIP INVITE message from the source device to the destination device. The source device must format the OSP token so that the destination device can recognize and validate the OSP Token.

Annex D of ETSI TS 101 321 V2.1.1 defines OSP token object identifiers when the token is carried as part of a call signaling message of an ASN.1-based protocol. Cisco has implemented OSP tokens as clear tokens in an XML format. The osp-token-xml-format OID used by Cisco is: 040132112. A description of the Cisco OSP token is provided below.

### Cisco OSP Token

1. Token should be present in the AuthorisationResponse message, one token for each Destination element.
2. Token uses XML format, with the following fields present:

TokenInfoRandom:	provided by the settlement server
SourceInfo:	same as in AuthorisationRequest
DestinationInfo:	same as in AuthorisationResponse
CallId:	same as in AuthorisationRequest
TransactionId:	same as in AuthorisationResponse
ValidUntil:	provided by the settlement server
ValidAfter:	provided by the settlement server
UsageDetail:	provided by the settlement server

3. Token is signed, but not encrypted, conforming to PKCS#7 standard for Signed Data.

Section 7 in DTS03004 describes in details the

necessary steps to create a signature.

The signer certificate should be sent as part of the Signed Data content, in the "certificates" fields so that the router can verify the token without having to store the server's certificate.

The following is an example of the token in PKCS#7 signed data format:

```

ContentInfo ::= SEQUENCE {
    contentType { pkcs-7 signedData(2) }
    content      tokenSigned
}

tokenSigned SignedData ::= {
    version 1
    digestAlgorithms { iso(1) member-body(2) US(840) rsadsi(113549)
        digestAlgorithm(2) 5}

    contentInfo {
        contentType { pkcs-7 1} -- data identifier
        content token          -- octet string representing the OSP
                                -- token in XML format. This token
                                -- is created by the settlement server
    }

    certificates { -- settlement server certificate chain
        certificate {
            version 3
            serialNumber          -- the settlement server certificate
                                -- serial number
            signature { pkcs-1 4 } -- md5WithRSAEncryption
            issuer                -- the certificate authority issuer name
            validity {
                notBefore         -- UTC time
                notAfter          -- UTC time
            }
            subject                -- the settlement server subject name
                                -- as given in PKCS#10
            subjectPublicKeyInfo {
                algorithm { pkcs-1 1}
                subjectPublicKey -- a BER encoding of the settlement server
                                public key as given in PKCS#10
            }
            extensions            -- the extensions as given in PKCS#10
            signatureAlgorithm { pkcs-1 4 }
        }
        certificate              -- the certificate authority certificate
    }

    signerInfo { -- including the digest of the token as
                -- the authenticated attributes
        version 1
        issuerAndSerialNumber {
            issuer                -- the certificate authority issuer name
            serialNumber          -- the CA's certificate serial number
        }
        digestAlgorithm { iso(1) member-body(2) US(840) rsadsi(113549)
            digestAlgorithm(2) 5}

        authenticateAttributes {
            contentType { {pkcs-9 3} {pkcs-7 1}}
            messageDigest { {pkcs-9 4} -- an octet string }
        }

        digestEncryptionAlgorithm {pkcs-1 1}

```

```
        encryptedDigest "encrypted digest of the message using the
                          server private key"
    }
}
```

## OSP token format for SIP

OSP, which is based on XML message transmitted via HTTP fits easily in the SIP architecture. OSP has been implemented have been implemented numerous SIP devices including the Vovida Policy Server, Commworks SIP proxy and Nuera's ORCA softswitch. As of this writing, the URL below references the IETF draft describing how an OSP token should be formatted in a SIP INVITE message:

<http://www.ietf.org/internet-drafts/draft-johnston-sip-osp-token-05.txt>

The title of the document is Session Initiation Protocol Private Extension for an OSP Authorization Token written by A. Johnston, D. Rawlins, H. Sinnreich, and S. Thomas on October, 2003. This draft proposes a new SIP (Session Initiation Protocol) header OSP-Authorization-Token for carrying an OSP (Open Settlements Protocol) authorization token between domains.

A copy of this draft is included for your convenience in Appendix A.

## Appendix A: *OSP Authorization Token Header for SIP*

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

Internet Engineering Task Force      A. Johnston  
Internet Draft      D. Rawlins  
Document: draft-johnston-sip-osp-token-05.txt      H. Sinnreich  
October 2003      MCI  
Expires: April 2004      Stephen Thomas  
      Wave7 Optics  
      Richard Brennan  
      Telxxis LLC

Session Initiation Protocol Private Extension for an  
OSP Authorization Token

### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/lid-abstracts.txt>  
The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

### Abstract

This document discusses a private extension to the Session Initiation Protocol (SIP) for carrying OSP (Open Settlements Protocol) authorization tokens in applications such as clearinghouses.

Johnston, et al.

Expires - April 2004

[Page 1]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

## Table of Contents

1. Introduction.....	2
2. Terminology.....	3
3. Design Alternatives.....	3
4. Header Field Definition.....	4
5. Protocol Semantics.....	5
5.1 User Agents.....	5
5.2 Proxies.....	5
6. Example Message.....	5
7. IANA Considerations.....	6
Security Considerations.....	6
Normative References.....	6
Informative References.....	7
Authors' Addresses.....	7

## 1. Introduction

The problem of interdomain IP telephony calls with QoS is an important problem being addressed using AAA protocols. The new private SIP [1] header field proposed here is part of an approach to solving this problem, which is summarized briefly here.

Interdomain IP telephony is accomplished today using clearinghouse services and a mix of proprietary and standard AAA protocols. Making calls with AAA support between service providers that are affiliated to different clearinghouses is a difficult problem.

Beyond IP telephony it is also desirable to have a consistent AAA approach for all applications on the Internet.

Work on a general architecture for AAA is proceeding in the IETF AAAArch research group. A framework and examples have been developed for various Internet applications. At the same time, Internet telephone calls can be set up with QoS and security. Since QoS is a valuable network resource, it requires AAA and possibly payments.

This draft documents a proprietary SIP extension header field that may be used to exchange open settlements protocol [4] information in the context of a SIP session establishment. The approach outlined here may be useful later for developing a uniform AAA architecture and protocols for other application layer services.

Figure 1 shows the model for an interdomain phone call across the Internet with the various entities having business relationships, but not necessarily trust relationships with their correspondents:

Johnston, et al.

Expires - April 2004

[Page 2]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

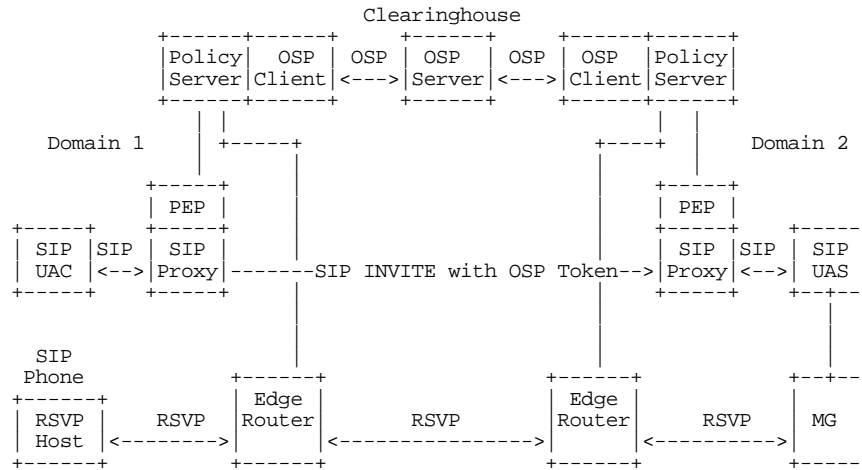


Figure 1: Model for interdomain QoS phone call

While this approach to interdomain authorization is not a complete one, it is currently used today by IP telephony carriers and is useful in limited applications such as in a clearinghouse. As such, it is appropriate for the header field extension to SIP be registered as a private SIP header field per the SIP change process [5]. Note that while RSVP [6] is shown, its use is not required by this extension.

## 2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in RFC 2119 [2] and indicate requirement levels for compliant SIP caller preferences implementations.

## 3. Design Alternatives

The OSP Token is an opaque string to SIP which must be carried in the INVITE passed between domains. As such, the Token could be carried as a MIME attachment. However, there are three issues with this:

Johnston, et al.

Expires - April 2004

[Page 3]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

- Since the Token must be carried with the SDP, the INVITE would need to have a multipart MIME message body. If either User Agents do not support multipart MIME, the call will fail.
- The Token is used by both proxies and User Agents. As such, the proxy would have to decode the multipart MIME message body to extract the token. The general design of SIP is for message bodies to contain information of interest to end-points only, with information needed by proxies contained in header fields.
- Multipart MIME encoding/decoding adds more delay to an already lengthy call setup procedure, as compared to header field processing.

For these reasons, a new SIP header field is proposed instead of a new MIME type for OSP authorization tokens.

Note that since OSP tokens are commonly constructed according to Cryptographic Message Syntax [3], their size may depend on the size of X.509 certificates embedded in the CMS format. For this reason, entities using this header field MUST NOT use UDP for transport. Instead TLS SHOULD be used. In addition, it is recommended that systems use the abbreviated token format described in Annex D of [4].

#### 4. Header Field Definition

The table below specifies an extension of Table 2 in RFC 3261 [1] for the new header field defined here.

	where	proxy	ACK	BYE	CAN	INV	OPT	REG
P-OSP-Auth-Token	R	ad	-	-	-	o	-	-
P-OSP-Auth-Token	18x,2xx	ad	-	-	-	o	-	-

The "where" column describes the request and response types with which the header field can be used. "R" indicates a request header field, a numeric value in the "where" column indicates the status code the header field is used with. The "proxy" column describes whether this message header field MAY be added, "a", or deleted, "d", by a proxy server. In the method columns, "o" means optional and "-" means not applicable.

The Augmented BNF for the header field (using the form and definitions in Section 25 of RFC 3261) is:

```
P-OSP-Auth-Token = "P-OSP-Auth-Token" HCOLON token *(SEMI osp-param)
osp-param       = realm / generic-param
realm           = "realm" EQUAL realm-value
```

Johnston, et al.

Expires - April 2004

[Page 4]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

realm-value      = quoted-string

## 5. Protocol Semantics

The OSP Token is always encoded per base64 and only allowed in INVITE requests, 200 OK responses to INVITES, and reliable provisional responses to INVITES.

### 5.1 User Agents

A UAC MAY include the header field an INVITE requesting QoS using AAA.

If present in an INVITE, an AAA/QoS UAS MAY validate the token.

If it is absent or present in the INVITE, an AAA/QoS UAS MAY include the header field in a reliable provisional response or 200 OK answer.

A UAC MAY validate the token received in a response to an INVITE.

### 5.2 Proxies

A proxy participating in the AAA exchange may add, delete, examine or validate the token.

Otherwise, the header field is ignored.

## 6. Example Message

This SIP INVITE message is an example exchange between the two domains as shown in Figure 1:

```
INVITE sips:+1-972-555-5555@domain2.example.com;user=phone SIP/2.0
Via: SIP/2.0/TLS proxy.domain1.example.com:5061;branch=z9hG4bK3a5d3.1
Via: SIP/2.0/TLS phonel.domain1.example.com:5061;branch=z9hG4bK3a5654
;received=192.0.2.1
Max-Forward: 69
From: Alice <sips:alice@phonel.domain1.example.com>;tag=3
To: <sips:+1-972-555-5555@domain2.example.com;user=phone>
Call-ID: 123456@domain1.example.com
CSeq: 1 INVITE
Contact: <sips:alice@phonel.domain1.example.com>
Record-Route: <sips:proxy.domain1.example.com;lr>
P-OSP-Auth-Token: "YT64VqpFyF467GhIGfHfYT6jH77n8HHGgHyHhHUuJhJh756t
HGTrfvbnjn8HHGTrfvhJhJh776tbB9HG4VQbnj7567GhIGfH
6ghyHhHUujpFyF47GhIGfHfYT64VQbnj";realm="domain1.example.com"
Content-Type: application/sdp
Content-Length: 184
```

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

```
v=0
o=alice 9735285123 9721273312 IN IP4 phone1.domain1.example.com
s=-
c=IN IP4 phone1.domain1.example.com
t=0 0
m=audio 9876 RTP/AVP 0
a=rtpmap: 0 PCMU/8000
a=qos:mandatory recv confirm
```

## 7. IANA Considerations

Registration of "P-OSP-Auth-Token" SIP header field

This document defines a new private SIP header field, "P-OSP-Auth-Token". As recommended by the policy of the Transport Area [5], this header field should be registered by the IANA in the SIP header field registry, using the RFC number of this document as its reference.

Name of Header field:	P-OSP-Auth-Token
Short form:	None
Registrant:	Alan Johnston alan.johnston@mci.com
Normative description:	This document

## Security Considerations

The security and handling of OSP tokens is covered in [4] which includes encryption and use of IPsec.

The P-OSP-Auth-Token header field may be protected using standard SIP mechanisms such as TLS transport and/or S/MIME encryption as detailed in [1].

Since the threats analyzed in the OSP document include ones in which the token is carried in plain text and available to an attacker, carrying the token in SIP does not introduce any new attacks.

## Normative References

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, and E. Schooler, "SIP: Session Initiation Protocol," Request for Comments (Proposed Standard) 3261, Internet Engineering Task Force, June 2002.

Johnston, et al.

Expires - April 2004

[Page 6]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

- [2] S. Bradner, "Key words for use in RFCs to indicate requirement levels," Request for Comments (Best Current Practice) 2119, Internet Engineering Task Force, March 1997.
- [3] R. Housley, "Cryptographic Message Syntax", RFC 2630, June 1999.
- [4] European Telecommunications Standards Institute.  
"Telecommunications and Internet Protocol Harmonization Over Networks (TIPHON); Open Settlement Protocol (OSP) for Inter-domain pricing, authorization, and usage exchange". Technical Specification 101 321. Version 2.1.0.

#### Informative References

- [5] A. Mankin, S. Bradner, R. Mahy, D. Willis, J. Ott, and B. Rosen, "Change Process for the Session Initiation Protocol (SIP)," Request for Comments (Proposed Standard) 3427, Internet Engineering Task Force, December 2002.
- [6] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification," Request for Comments (Proposed Standard) 2205, Internet Engineering Task Force, October 1997.

#### Authors' Addresses

Alan Johnston  
MCI  
100 S. 4th Street  
St. Louis, Missouri 63102  
alan.johnston@mci.com

Henry Sinnreich  
MCI  
400 International Parkway  
Richardson, Texas 75081  
USA  
henry.sinnreich@mci.com

Diana Rawlins  
MCI  
901 International Parkway  
Richardson, Texas 75081  
USA  
diana.rawlins@mci.com

Stephen Thomas  
Wave7 Optics

Johnston, et al.

Expires - April 2004

[Page 7]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

1075 Windward Ridge Parkway  
Alpharetta, GA 30005  
USA  
stephen.thomas@wave7optics.com

Richard Brennan  
Telxxis LLC  
1670 South Amphlett Blvd.  
Suite 214, No. 1018  
San Mateo, CA 94402-2511  
USA  
rbrennan@telxxis.com

#### Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

#### Full Copyright Notice

"Copyright (C) The Internet Society 2003. All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other

Johnston, et al.

Expires - April 2004

[Page 8]

Internet-Draft      SIP Extension for OSP Auth Token      October 2003

Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.