SIP Auto Peer

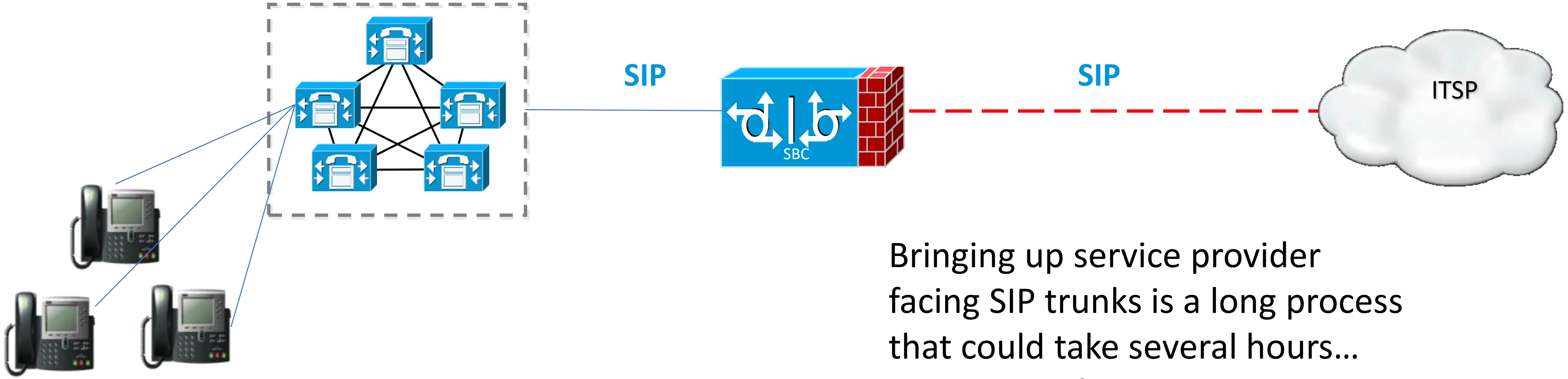
draft-kinamdar-dispatch-sip-auto-peer-01

C. Jennings
K. Inamdar
S. Narayanan

Agenda

- Problem Space
- Solution Overview
- Asks

Problem Space



Bringing up service provider facing SIP trunks is a long process that could take several hours… sometimes days…

- Administrators usually uncover a large number of problems when the trunk is setup.
- Requires time-consuming interaction between the administrator and service provider to ensure everything is working.
- Deployment times increase significantly due to interoperability problems.

Magnitude of Problem

- A total of 6000 support cases opened with Cisco last year for its Enterprise SBC, CUBE
- 22% of these cases were directly related to ITSP interoperability
- Multiply this by the number of enterprise SBC vendors to get a rough estimate of the magnitude of the problem
- Still a significant number of enterprise networks are yet to migrate from TDM/Analog to SIP trunking … the problem isn't going away anytime soon!

Current Solutions

- SIP Connect Technical Recommendations
- Service Provider Documentation

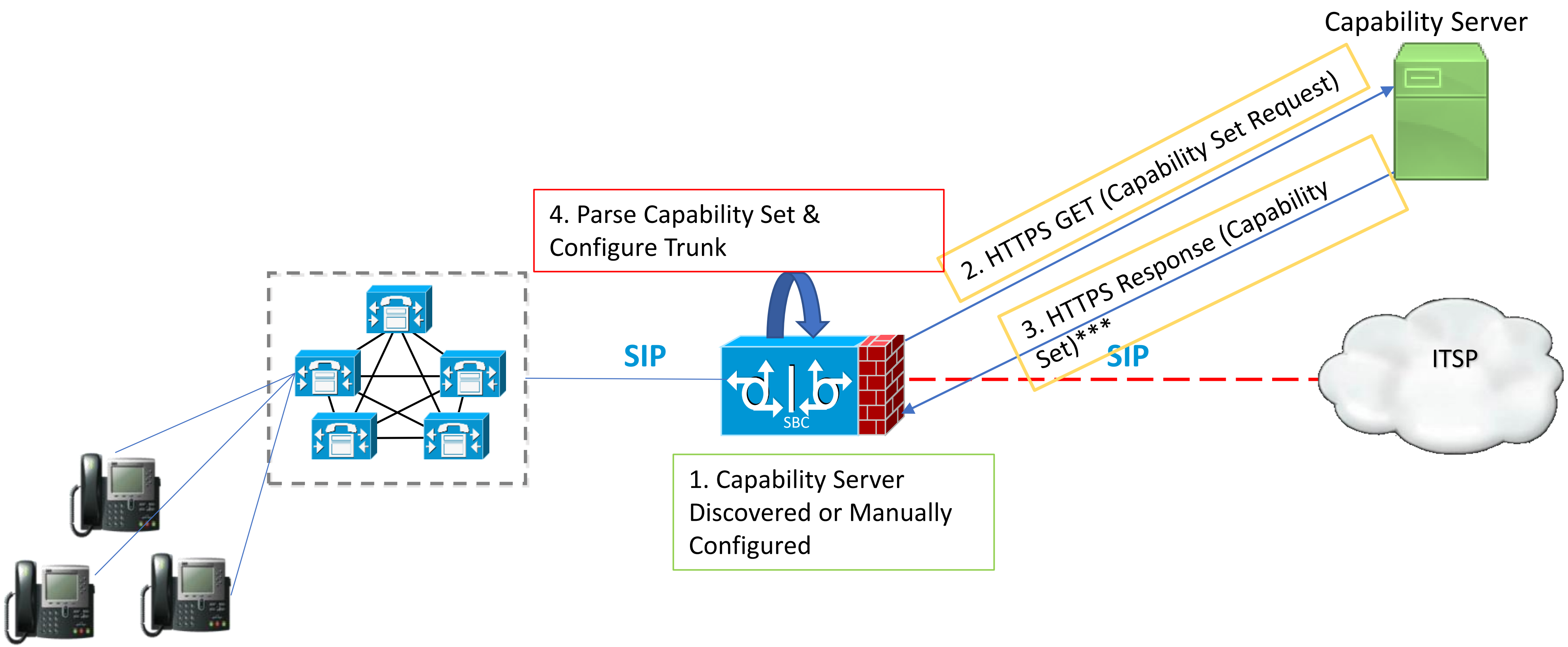
Issues Still Persist…

- What happens if the ITSP forks calls? The SBC has to be configured to either pass through all responses or pass through only the first confirmed dialog.
- What happens for hairpinned calls? Does the ITSP expect the enterprise to send the first RTP packet? Can the enterprise network “hand-off” the call?
- How many re-INVITES with or without SDP change can be sent to the ITSP?
- Is the called party number in the Request-URI? To header field or the P-Called-Party ID?
- How is the user portion of the *From header fie*ld formatted? Is it any number or specifically a number within the DID range allocated to the enterprise?
- Will the ITSP reject calls that have a video media line (along with an audio media line) or follow the offer/answer model and simply set the video media line to 0?

SIP Auto Peer

- SIP Auto Peer is a framework that converts answers to questions typically needed by an administrator to setup a SIP trunk into a machine workable format.
- Parameters are encoded according to the YANG model defined in the draft and communicated to the enterprise edge element.
- Parameters may be subsequently parsed by automaton on the edge element and converted to configuration.
- Parameters are a combination of both basic (e.g. registrar, call control IP) and nuanced (e.g. who sends the first RTP packet for a hairpinned call) parameters.
- Only some parameters are included in Version 00 and 01 of this draft…we hope to work with the larger community to define a robust/complete parameter set.

High-Level Overview



*** Body encoded in XML or JSON

Why HTTPS?

- HTTPS is the path of least disruption – no changes required to SIP stacks.
- Exchange of SIP traffic between enterprise and service provider networks usually requires SIP trunk registration.
- Exchange of the capability set after trunk registration is - well - pointless.
- Most SBCs also have an HTTP stack built-in.

XML or JSON?

XML

Pros:

- XML and configuration files are synonymous.
- SBCs usually have XML parsers built-in
- Strong schema constraint languages for XML

Cons:

- Bulky, verbose
- Requires translation of doc to structure

JSON

Pros:

- Compact and more readable
- Data formatted as a map

Cons:

- Can have issues with weakly structured data

Yang Model

- Parameters applicable to any vendor SBC for SIP trunk registration
- Significant media parameters
- Security specifications
- SIP extensions supported by SP

```
module: ietf-sip-auto-peering
  +--rw peering-info
     +--rw variant              string
     +--rw transport-info
     |  +--rw transport?         enumeration
     |  +--rw registrar*         host-port
     |  +--rw registrarRealm?    string
     |  +--rw callControl*       host-port
     |  +--rw dns*               inet:ip-address
     |  +--rw outboundProxy?     host-port
     +--rw call-specs
     |  +--rw earlyMedia?          boolean
     |  +--rw signalingForking?    boolean
     |  +--rw supportedMethods?    string
     +--rw media
     |  +--rw mediaTypeAudio
     |  |  +--rw mediaFormat*    string
     |  +--rw fax
     |  |  +--rw protocol*    enumeration
     |  +--rw rtp
     |  |  +--rw RTPTrigger?      boolean
     |  |  +--rw symmetricRTP?    boolean
     |  +--rw rtcp
     |     +--rw symmetricRTCP?    boolean
     |     +--rw RTCPfeedback?     boolean
     +--rw dtmf
     |  +--rw payloadNumber?   int8
     |  +--rw iteration?       boolean
     +--rw security
     |  +--rw signaling
     |  |  +--rw type?      string
     |  |  +--rw version?   string
     |  +--rw mediaSecurity
     |     +--rw keyManagement?    string
     +--rw extensions?        string
```

Example Capability Set

```xml
<peering-info xmlns="urn:ietf:params:xml:ns:yang:ietf-peering"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:ietf:params:xml:ns:yang:ietf-peering ietf-peering.xsd">
      <variant>1.0</variant>
      <transport-info>
        <transport>TCP;TLS;UDP</transport>
        <registrar>registrar1.voip.example.com:5060</registrar>
        <registrar>registrar2.voip.example.com:5060</registrar>
        <registrarRealm>voip.example.com</registrarRealm>
        <callControl>callServer1.voip.example.com:5060</callControl>
        <callControl>192.168.12.25:5065</callControl>
        <dns>8.8.8.8</dns>
        <dns>208.67.222.222</dns>
        <outboundProxy>0.0.0.0</outboundProxy>
      </transport-info>
      <call-specs>
        <earlyMedia>true</earlyMedia>
        <signalingForking>false</signalingForking>
        <supportedMethods>INVITE;OPTIONS;BYE;CANCEL;ACK;PRACK;SUBSCRIBE;NOTIFY;REGISTER</supportedMethods>
      </call-specs>
      <media>
        <mediaTypeAudio>
          <mediaFormat>PCMU;rate=8000;ptime=20</mediaFormat>
          <mediaFormat> G729;rate=8000;annexb=yes</mediaFormat>
          <mediaFormat>G722;rate=8000;bitrate=56k,64k</mediaFormat>
        </mediaTypeAudio>
        <fax>
          <protocol>pass-through</protocol>
          <protocol>t38</protocol>
        </fax>
        <rtp>
          <RTPTrigger>true</RTPTrigger>
          <symmetricRTP>true</symmetricRTP>
        </rtp>
        <rtcp>
          <symmetricRTCP>true</symmetricRTCP>
          <RTCPFeedback>true</RTCPFeedback>
        </rtcp>
      </media>
      <dtmf>
        <payloadNumber>101</payloadNumber>
        <iteration>0</iteration>
      </dtmf>
      <security>
        <signaling>
          <type>TLS</type>
          <version>1.0;1.2</version>
        </signaling>
        <mediaSecurity>
          <keyManagement>SDES;DTLS-SRTP,version=1.2</keyManagement>
          </mediaSecurity>
        </security>
      <extensions>timer;rel100;gin;path</extensions>
</peering-response>
```

The way forward…

- Objections / Challenges?

- Refining the capability set.   Additional parameters?
    - URI Scheme of calls – TEL or SIP URIs?
    - Full or Compact form headers?
    - DID block allocated by ITSP?
    - Emergency Numbers?

- How should the draft be taken forward?
    - Mini Workgroup
    - SIP Core
    - AD Sponsored

Thank You