Communication Protocols and Internet Architectures

Harvard University

Lecture #13

Instructor: Len Evenchik
cs40@evenchik.com or evenchik@fas.harvard.edu

ALIGHLSOD1701

Lecture Agenda

- Course Logistics
- Q&A and Topics from Last Week
- Session Initiation Protocol (SIP)
- Software Defined Networks (SDN)
- Network Function Virtualization (NFV)
- One Minute Wrap-Up

Course Logistics

Course Logistics

- Final Exam – Please check the weekly course information sheet for detailed information on the final.
- Upcoming Guest Lecture
- Homework #5 have been posted.
- Always check the weekly course information sheet for any updated schedule information for section meetings.
- **Please submit a one minute wrap-up each week. Thank You!**

Q&A
Topics from Last Week

Structured way to Think about Security: Five Important Elements

- Privacy and confidentiality
- Authentication
- Authorization
- Integrity
- Nonrepudiation

Digital Signatures

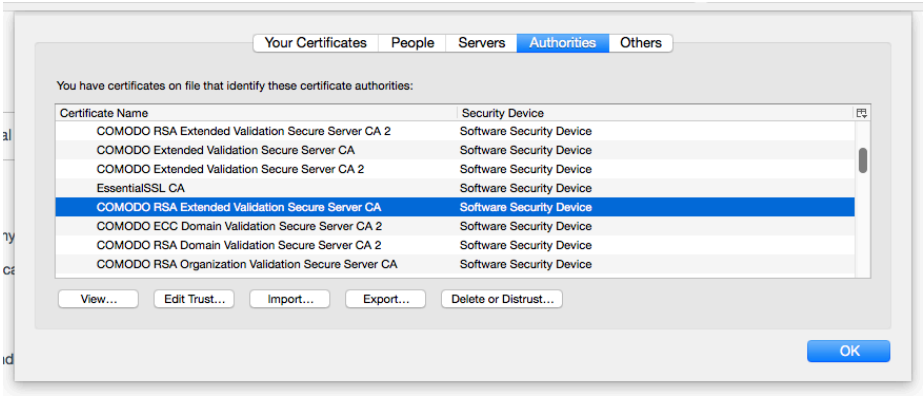
- A digital signature should "prove" that a message came from a specific user (lets call them UserA) and the message has not changed
- One way to produce a digital signature
  - UserA computes a one-way hash function on the contents of the message
  - UserA encrypts the hash code using their private key
  - The encrypted hash code is appended to the message and the combination is sent to UserB
  - UserB computes the same hash function on the contents of the message
  - UserB then decrypts the received hash code with UserA's public key
  - If the hash codes match, the message came from UserA and the message was not changed in transit

X.509 Public Key Infrastructure

X.509 is NOT the same as TLS

- The PKI requires a trusted third party, called a Certificate Authority (CA), and the assumption is that the Public key of the CA has been securely loaded into browsers, clients, and phones.
- A website admin sends their Public key to a CA, and the CA validates the identity of the website and then digitally signs a document called a Certificate that includes the website's Public key.
- Given that the Public key of the CA has been previously and securely loaded into browsers, the validity of a website's Certificate, and hence the identity and Public key of a website, can be verified.
- A Certificate Chain, which is a hierarchical trust model, is common for CAs. The initial CA in the chain is called the root.
- There are 100s of CAs, some more trustworthy than others.
- Websites use a combination of Certificates and TLS to secure traffic. A Certificate is not the same as TLS.

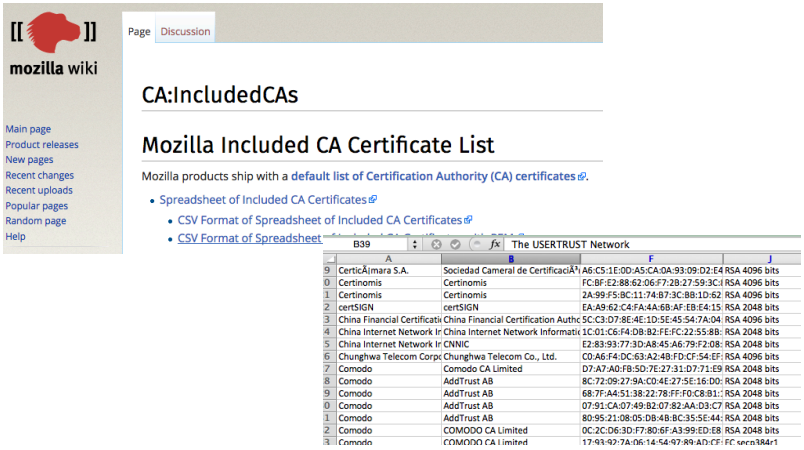
Browser Certificates (3)



Browser Certificate List

Different Browsers and OS Store the Lists in Different Ways

https://wiki.mozilla.org/CA/Included_Certificates



CAs Do Make Errors and Have Other Problems

https://security.googleblog.com/2017/09/chromes-plan-to-distrust-symantec.html

Chrome's Plan to Distrust Symantec Certificates

September 11, 2017

At the end of July, the Chrome team and the PKI community converged upon a plan to reduce, and ultimately remove, trust in Symantec's infrastructure in order to uphold users' security and privacy when browsing the web. This plan, arrived at after significant debate on the blink-dev forum, would allow reasonable time for a transition to new, independently-operated Managed Partner Infrastructure while Symantec modernizes and redesigns its infrastructure to adhere to industry standards. This post reiterates this plan and includes a timeline detailing when site operators may need to obtain new certificates.

https://security.googleblog.com/2015/09/improved-digital-certificate-security.html

Improved Digital Certificate Security

September 18, 2015

Posted by Stephan Somogyi, Security & Privacy PM, and Adam Eijdenberg, Certificate Transparency PM

On September 14, around 19:20 GMT, Symantec's Thawte-branded CA issued an Extended Validation (EV) pre-certificate for the domains google.com and www.google.com. This pre-certificate was neither requested nor authorized by Google.

We discovered this issuance via Certificate Transparency logs, which Chrome has required for EV certificates starting January 1st of this year. The issuance of this pre-certificate was recorded in both Google-operated and DigiCert-operated logs.

https://wiki.mozilla.org/CA:IncludedCAs

Mozilla Security Blog

OCT 24 2016

Distrusting New WoSign and StartCom Certificates

kwilson

Mozilla has discovered that a Certificate Authority (CA) called WoSign has had a number of technical and management failures. Most seriously, we discovered they were backdating SSL certificates in order to get around the deadline that CAs stop issuing SHA-1 SSL certificates by January 1, 2016. Additionally, Mozilla discovered that WoSign had acquired full ownership of another CA called StartCom and failed to disclose this, as required by Mozilla policy. The

© 1998 - 2017 L. Evenchik

Voice and Video Over IP

This is Not VoIP

By courtesy of] [The American Telephone and Telegraph Co.

A LONG-DISTANCE TELEPHONE EXCHANGE.

Radio-telephone switchboard circa 1930. From the left the first four stations are to London, the next Ship to Shore, Buenos Aires, and Rio de Janeiro.

AT&T Photo

Introduction

- Voice and video are both analog signals and must be converted to a digital signal (compressed and coded) for transport over packet switched networks.
- The video (or voice) transport can be one-way or full-duplex, and it can be real-time, or not real-time (i.e., streaming video.)
- Today, IP is the obvious protocol for carrying video and voice, but many other proprietary protocols have been used over the years.
- SIP is the predominant choice today for Voice over IP (VoIP) and newer videoconferencing systems. H.323 is still used for video in some corporate networks, but this is changing rapidly. Skype, now Microsoft, is a proprietary protocol. WebRTC is a browser-focused protocol approach.
- These are all protocol suites, not a single protocol.
- The transport of video or voice over packet based networks requires:
  - protocols for setting up the connection (called signaling)
  - protocols for establishing the capabilities of the end systems
  - protocols for actually sending the video and audio

Audio and Video Codecs
(We'll talk about this more when with discuss QoS.)

- A codec converts an analog signal (either voice or video) to a digital signal (and vice versa)
- Audio Codecs
  - G.711 (8,000 samples per second, 64kbps, 30 msec sample)
  - G.722 (7Khz speech, 48kbps to 64 kbps)
  - G.723.1 (30 msec sample, 6.4kbps)
  - G.728 (16kbps, LD-CELP)
  - G.729 (8kbps, CELP)
  - Plus many proprietary and open source standards
- Video Codecs
  - H.261 (the first packet based video compression standard)
  - H.263, H.263+ and H.263++
  - H.264 (multiple versions and standards)
  - H.265 (most recent standard)
  - Plus many proprietary and open source standards

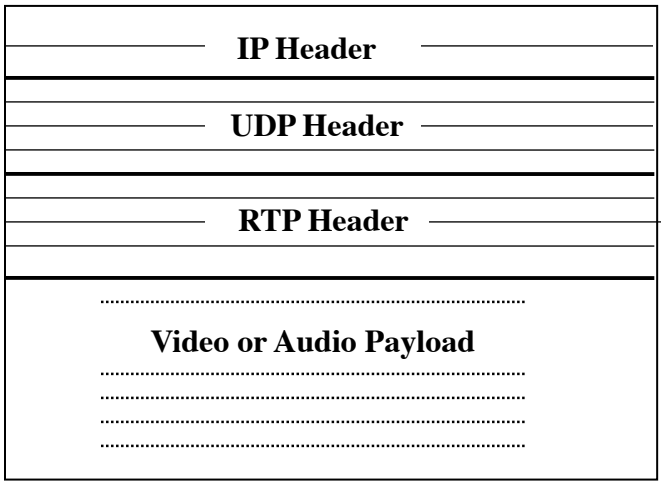
Introduction to RTP (1)

- Video and voice packets cannot be carried directly by UDP without additional functionality.
- The Real-time Transport Protocol is an IETF transport protocol for real time applications such as voice and video. It is standardized in RFCs 3550 and 3551.
- RTP uses UDP transport, with the inherent and limited functionality provided by UDP. This means error detection but not correction.
- RTP provides data sequencing, timing and synchronization
- RTP is augmented by a "control" protocol called RTCP (Real-Time Transport Control Protocol) which loosely monitors the flow

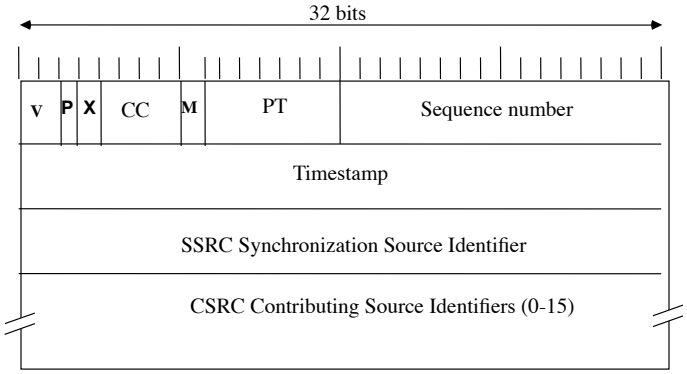
Introduction to RTP (2)

- RTP provides data sequencing, timing and synchronization
- RTCP provides media synchronization, feedback and forward status information
- RTP/RTCP flow uses a pair of UDP channels in each direction
- The Secure Real-time Transport Protocol (SRTP) RTP (RFC 3711) has also been defined and is used. There are other encrypted VoIP protocols such as ZRTP.

Combined IP/UDP/RTP Packet



RTP Header Format



RTP Header Fields

- V: version number
- P: flag to indicate padding bytes are present
- X: header extension flag
- PT: Payload Type
- CC: CSRC count
- M: marker (media dependent, defined in RTP profile)
- timestamp: sampling instant of the first byte, from media encoding clock
- SSRC: Synchronization source, the source of a single stream
- CSRC: Contributing source, a source that contributes to the combined stream produced by an RTP mixer

RTP Packet Flow for Video Call

SIP: Session Initiation Protocol

© 1998 - 2017 L. Evenchik

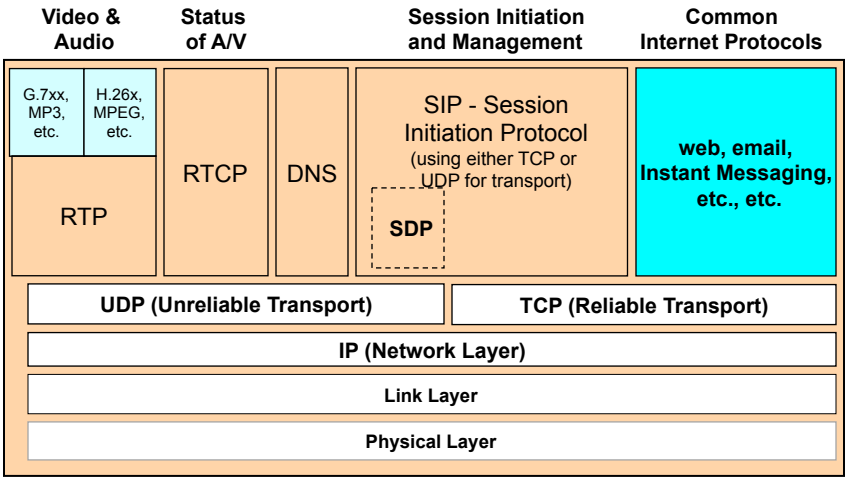
Introduction to SIP

- SIP stands for Session Initiation Protocol: This is the IETF protocol for session initiation and management. SIP does not carry voice or video packets; this is done by RTP.)
- SIP is the dominant protocol for Voice over IP (VoIP) signaling, but sessions can be many different things:
  - Telephone calls (business telephone systems)
  - Video calls
  - IM and chat traffic
  - Multimedia sessions with multiple parties
- SIP leverages other IETF and web protocols. SIP should be considered a suite of protocols.
- SIP's initial goals were simplicity and modularity, but today, it is anything but simple.

Protocols Required for a VoIP Call

- The transport of video or voice over packet switched networks requires protocols to support three different functions:
  - protocols for setting up the connection or session (signaling)
  - protocols for determining and deciding upon the specific voice and video capabilities and parameters that will be used by the end systems during the session (call)
  - protocols for actually sending the video and audio
- We describe these as the **three phases of the call** and it is a great way to understand voice and video protocols. However, it is a simplification in some cases given that protocols that set the capabilities can be piggybacked within the signaling phase.

SIP Protocol Architecture



Introduction to SIP (2)

- SIP is an application layer signaling protocol that looks a lot like a combination of HTTP (web) and SMTP (email).
- SIP messages are text based, comparable to email (no ASN.1 encoding is used as was done in H.323)
- SIP messages are formatted somewhat like email messages
- SIP users are addressed by a SIP URI (sip:alice@harvard.edu)
- Telephone numbers can also be defined and used

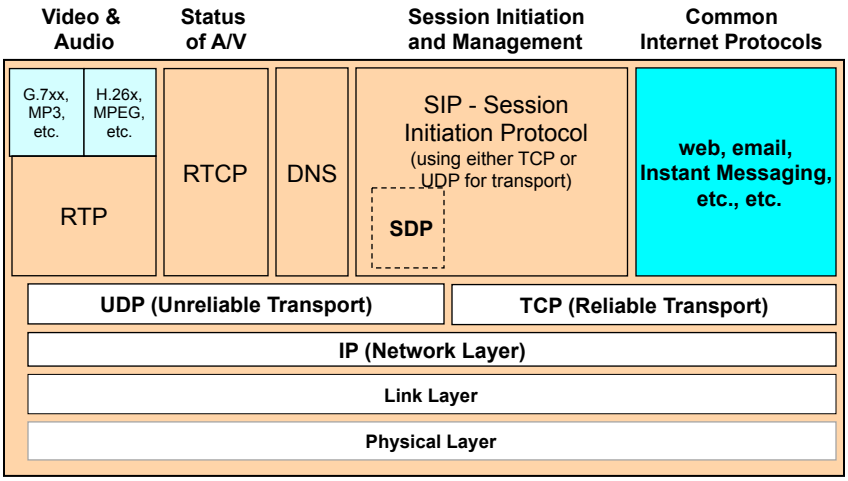
Introduction to SIP (3)

- SIP sessions and media capabilities are described by SDP, Session Description Protocol
- SIP protocol uses a Request / Response approach
- SIP message start with a Method and are followed by multiple headers
  - Methods are the actions to be performed
  - Headers contain the needed parameters and details
- SIP can use TCP, UDP or TLS as the transport protocol
- If you know H.323, it is helpful to compare SIP to H. 323 (but we will not do this here.)

Primary SIP and SIP Related Protocols

- SIP: RFC 3261, core SIP protocol RFC
- SDP: RFC 4566, Session Description Protocol, describes multimedia sessions
- RFC 3263, Locating SIP Servers
- RFC 3264, an Offer/Answer model for SDP
- RTP: RFC 3550, A Transport Protocol for Real-Time Applications
- *Plus dozens of others*. We should have a hitchhikers guide for this journey, so take a look at RFC 5411.

SIP Protocol Architecture



SIP Related IETF Working Groups

- SIPCORE: Session Initiation Protocol
- CLUE: ControLling mUltiple streams for tElepresence
- Plus others including: enum, avt and drinks

SIP Addressing (1)

- SIP uses URI style addressing; the common way that this is explained is to say that SIP uses email style addressing, such as alice@atlanta.com
- URI stands for Uniform Resource Identifier and this approach provides a simple and extensible means for identifying a specific resource. It was introduced for use with the web.
- A URI begins with a scheme (such as http or sip), schemes are defined at:
  - http://www.iana.org/assignments/uri-schemes.html
- Example URIs:
  - http://www.ietf.org/rfc/rfc2396.txt
  - mailto:John.Doe@example.com
  - tel:+1-816-555-1212
  - sip:lensip@harvard.edu

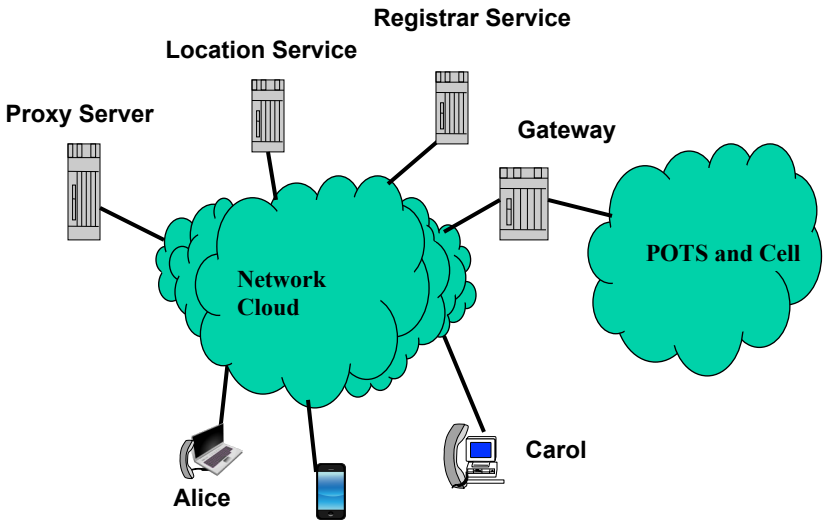
sources rfc3986, rfc4395, rfc3261

SIP Addressing (2)

- SIP or SIPS addresses identify a user, phone, computer, telephone number, service or resource.
- The general form of a SIP URI is,
  sip:user:password@host:port;uri-parameters?headers
  - host can be an IP address, a FQDN (pc1.atlanta.com) or domain (atlanta.com)
  - uri-parameters take the form parameter-name "=" parameter-value
  - a password used in this way would not be secure
- Common form of SIP addresses:
  - sip:alice@atlanta.com
  - sip:bob@biloxi.com;transport=udp
  - sip:+1-212-555-1212:1234@gateway.com:10100;user=phone
  - sips:6100@siplearn.com:5060
  - sip:alice@18.0.2.4
- SIPS specifies a secure channel, but there are problems with this approach. There are other options such as ZRTP.
- AOR means Address of Record and it is intended to be a public SIP user address.

sources rfc3986, rfc4395, rfc3261

SIP Building Blocks



SIP Network Building Blocks

- User Agents (UA)
- Proxies
- Registrar Server
- Redirect Services/Servers
- Location Services/Servers
- Gateways
- Application Services/Servers
- Media Servers
- DNS
- Back-2-Back User Agents (B2BUA)
- Firewalls, Session Border Controllers

SIP User Agent

- User Agent (UA): A logical entity that can act as both a user agent client and user agent server.
- User Agent Client (UAC): A user agent client is a logical entity that creates a new session. The role of UAC lasts only for the duration of that transaction.
- User Agent Server (UAS): A user agent server is a logical entity that generates a response to a SIP request. The response accepts, rejects, or redirects the request. This role lasts only for the duration of that transaction.
- The above is a rather formal way to say that a UA such as phone can create or accept calls (sessions.)
- Consider how this relates to web clients and servers.

(source RFC 3261)

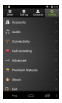
SIP Softphones and Hardware-based Phones
(1,000s of Options Today)

Hardware based SIP Phones and VoIP adapters

SIP Softphones are available for all OS

We will use JITSI for some of our demos but there are many other good options

SIP Softphones for Android, IOS, etc



SIP Server and SIP Proxy

- Server: A server is a network element that receives SIP requests in order to service them and sends back SIP responses to those requests. Examples of servers are registrar servers, location servers, redirect servers, etc.
- Proxy, Proxy Server: An intermediary entity that acts as both a server and a client for the purpose of making requests on behalf of other clients.

(source RFC 3261)

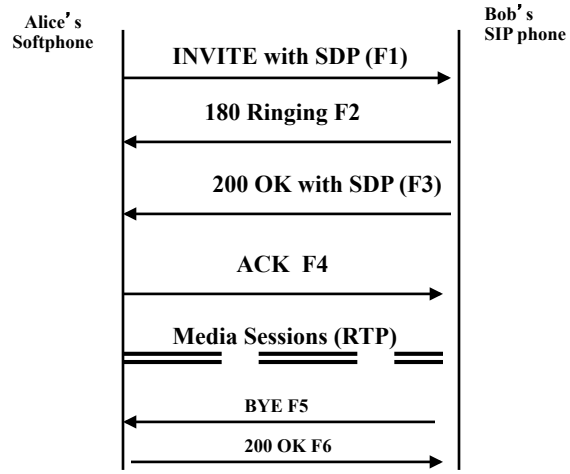
SIP Session Support

SIP supports establishing, managing and terminating multimedia sessions.

- Session setup: calling, ringing, setting session parameters
- Session management: transfer and termination, service invocation, modification of session parameters
- User location: determining the location of the end system
- User availability: willingness of the called party to engage in communications
- User capabilities: determination of the media and media parameters to be used

(source RFC 3261)

Point-to-Point SIP



(source RFC 3261)
F1, F2 etc are in the RFC

SIP Methods

- Six methods are defined in RFC 3261
  - REGISTER
  - INVITE
  - ACK
  - CANCEL
  - BYE
  - OPTIONS
- Additional methods have been defined for uses such as IM and are documented in standards track RFCs
- New methods continue to be defined and debated. The current list can be found at: http://www.iana.org/assignments/sip-parameters

SIP INVITE Message

(Simple form)

INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;
branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
(Alice's SDP not shown)

(source RFC 3261)

Format of SIP Requests and Responses

- SIP is a text-based protocol and it uses a request / response approach
- SIP messages have three parts: first line (or start line), one or more header fields, and message body (optional)
- In a Request, the first line identifies the Method and includes a Request-URI, and then the protocol version
- In a Response, the first line includes the protocol version followed by a numeric status code, and then text that further explains the status code. (Remember SMTP used numeric status codes.)
- Header fields provide required information such as addresses and sequence numbers and have the form -
  field name : field value ; parameters (as needed)
- The message body includes additional information such as SDP data.

SIP Reply Codes

- 1xx: Provisional -- request received, continuing to process the request;
- 2xx: Success -- the action was successfully received, understood, and accepted;
- 3xx: Redirection -- further action needs to be taken in order to complete the request;
- 4xx: Client Error -- the request contains bad syntax or cannot be fulfilled at this server;
- 5xx: Server Error -- the server failed to fulfill an apparently valid request;
- 6xx: Global Failure -- the request cannot be fulfilled at any server.

(source RFC 3261)

Encapsulation of a SIP Invite

Frame (1007 bytes on wire, 1007 bytes captured)

Ethernet II, Src: 00:03:47:8f:ba:dd, Dst: 00:d0:00:db:23:fc

Internet Protocol, Source: 140.247.197.83 Destination: 195.37.77.99

User Datagram Protocol, Src Port: 5060, Dst Port: 5060

Session Initiation Protocol
Request-Line: INVITE sip:cbarkley2442@iptel.org SIP/2.0
Method: INVITE
MUCH MORE DETAIL TO FOLLOW

SIP Invite (part 1)

```
Session Initiation Protocol
   Request-Line: INVITE sip:cbarkley2442@iptel.org SIP/2.0
      Method: INVITE
   Message Header
      Via: SIP/2.0/UDP 140.247.197.83:5060;rport;branch=z9hG4bKDAED….
      From: TestUser1 <sip:bkermit2442@iptel.org>;tag=2672264672
         SIP Display info: TestUser1
         SIP from address: sip:bkermit2442@iptel.org
         SIP tag: 2672264672
      To: <sip:cbarkley2442@iptel.org>
         SIP to address: sip:cbarkley2442@iptel.org
      Contact: <sip:bkermit2442@140.247.197.83:5060
      Call-ID: F68DE3D8-2245-4A16-A820-88752F2222AE@140.247.197.83
      CSeq: 22610 INVITE
  Proxy-Authorization: Digest username="bkermit2442 “ ,realm="iptel.org",
nonce="417ad0b1a61…",response="5595fd",uri="sip:cbarkley2442@iptel.or
g"
      Max-Forwards: 70
      Content-Type: application/sdp
      User-Agent: X-Lite release 1103m
      Content-Length: 302
```


SIP Invite (part 2)

```
Message body
      Session Description Protocol
         Session Description Protocol Version (v): 0
            Owner, Session Id (o): bkermit2442 1803743 1804204
             IN IP4 140.247.197.83
            Owner Username: bkermit2442
            Session ID: 1803743
            Session Version: 1804204
            Session Name (s): X-Lite
       Connection Information (c): IN IP4 140.247.197.83
            Connection Network Type: IN
            Connection Address Type: IP4
            Connection Address: 140.247.197.83
       Media Description, (m): audio 8000 RTP/AVP 0 8 3 98 97 101
            Media Type: audio
            Media Port: 8000
            Media Proto: RTP/AVP
            Media Format: ITU-T G.711 PCMU
            Media Format: ITU-T G.711 PCMA
            Media Format: GSM 06.10
              PLUS Others not shown here
```

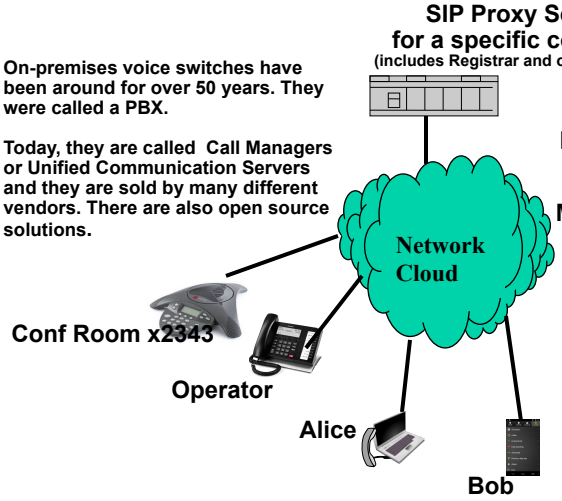
SIP Invite:
the bits and nothing but the bits

```
0000  00 d0 00 db 23 fc 00 03 47 8f ba dd 08 00 45 00   ....#...G.....E.
0010  03 e1 0b 53 00 00 80 11 00 00 8c f7 c5 53 c3 25   ...S.........S.%
0020  4d 63 13 c4 13 c4 03 cd 0f e3 49 4e 56 49 54 45   Mc........INVITE
0030  20 73 69 70 3a 63 62 61 72 6b 6c 65 79 32 34 34    sip:cbarkley244
0040  32 40 69 70 74 65 6c 2e 6f 72 67 20 53 49 50 2f   2@iptel.org SIP/
0050  32 2e 30 0d 0a 56 69 61 3a 20 53 49 50 2f 32 2e   2.0..Via: SIP/2.
0060  30 2f 55 44 50 20 31 34 30 2e 32 34 37 2e 31 39   0/UDP 140.247.19
0070  37 2e 38 33 3a 35 30 36 30 3b 72 70 6f 72 74 3b   7.83:5060;rport;
0080  62 72 61 6e 63 68 3d 7a 39 68 47 34 62 4b 44 41   branch=z9hG4bKDA
0090  45 44 38 35 37 46 34 45 39 44 34 34 36 36 38 44   ED857F4E9D44668D
00a0  42 46 37 42 42 44 44 34 46 31 36 33 34 31 0d 0a   BF7BBDD4F16341..
00b0  46 72 6f 6d 3a 20 74 65 73 74 6f 6e 6c 61 70 74   From: testonlapt
00c0  6f 70 20 3c 73 69 70 3a 62 6b 65 72 6d 69 74 32   op <sip:bkermit2
00d0  34 34 32 40 69 70 74 65 6c 2e 6f 72 67 3e 3b 74   442@iptel.org>;t
00e0  61 67 3d 32 36 37 32 32 36 34 36 37 32 0d 0a 54   ag=2672264672..T
```


SIP Proxies and SIP Trapezoid

Building Real World SIP-based Networks

SIP Proxy Server supporting a single company



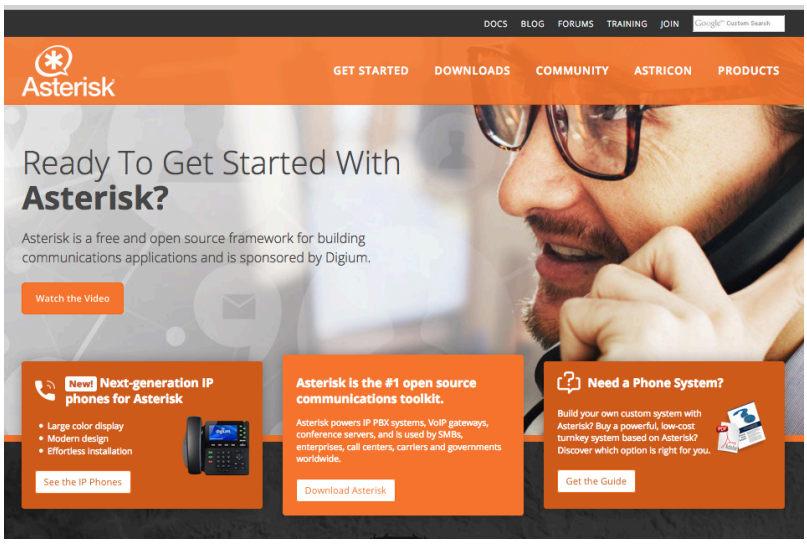
On-premises voice switches have been around for over 50 years. They were called a PBX.

Today, they are called Call Managers or Unified Communication Servers and they are sold by many different vendors. There are also open source solutions.

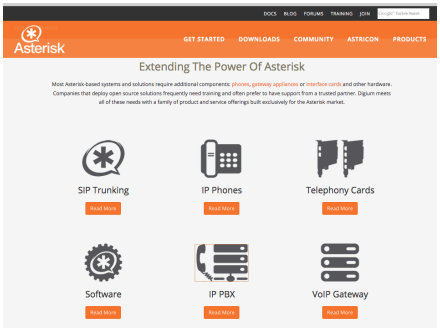
Today, most on-premises phone systems use SIP or a proprietary version of SIP. This includes Cisco, Microsoft and Asterisk voice systems.

Telephone companies also offer SIP based service for use by a company. These are marketed as Cloud Services

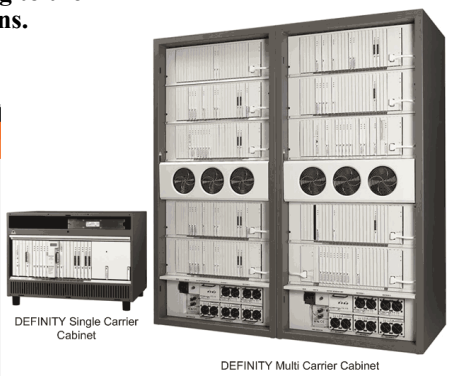
Asterisk is a Very Popular Open Source Solution



The Trend in Unified Communications (UC) for Many Years has been a move to Open Source and Generic Hardware from Proprietary Hardware, Software and Protocols. In addition, voice services are moving to the Cloud versus premises-based solutions.

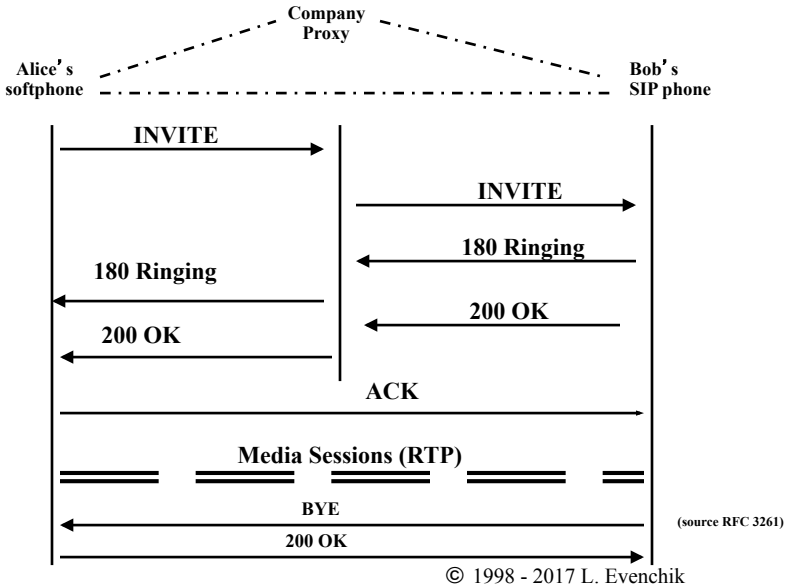


Asterisk



Old Style Proprietary Systems, 1980s

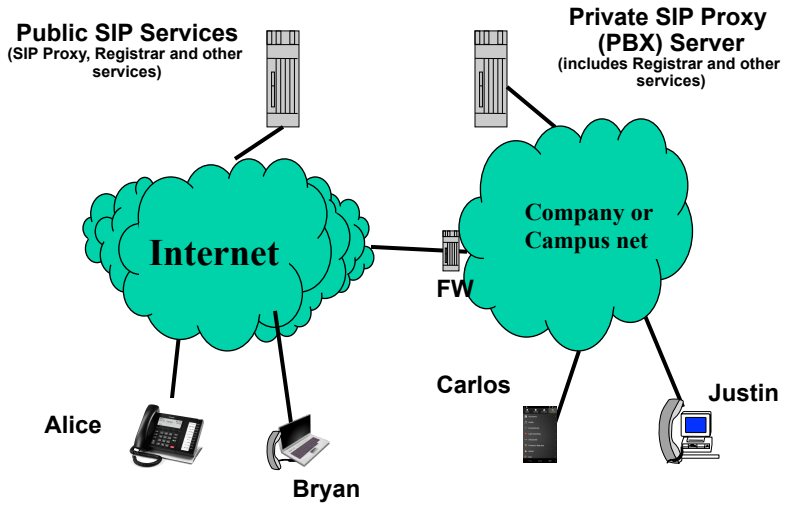
SIP Call via a Single Proxy



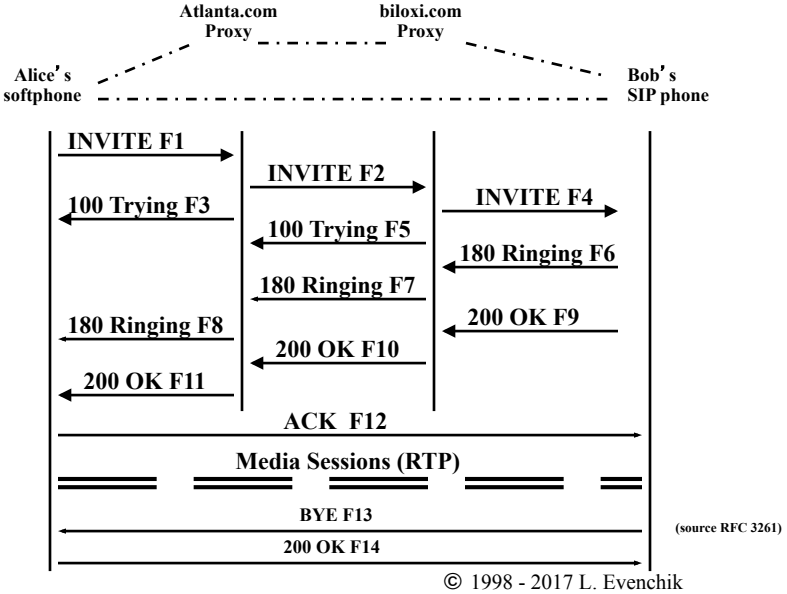
(source RFC 3261)

Public and Private SIP-based Networks

There are 100s of public VoIP service providers that use SIP



SIP Trapezoid



(source RFC 3261)

SIP INVITE Message
(Simple form)

```
INVITE sip:bob@biloxi.com SIP/2.0
Via: SIP/2.0/UDP pc33.atlanta.com;
        branch=z9hG4bK776asdhds
Max-Forwards: 70
To: Bob <sip:bob@biloxi.com>
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:alice@pc33.atlanta.com>
Content-Type: application/sdp
Content-Length: 142
  (Alice's SDP not shown)
```

(source RFC 3261)

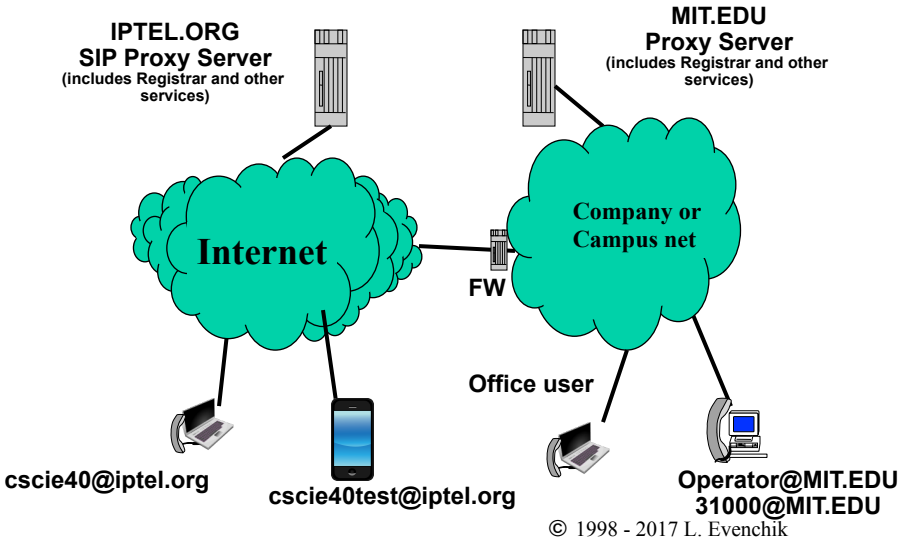
SIP Session INVITE
200 OK Response

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server10.biloxi.com
      ;branch=z9hG4bKnashds8;received=192.0.2.3
Via: SIP/2.0/UDP bigbox3.site3.atlanta.com
      ;branch=z9hG4bK77ef4c2312983.1;received=192.0.2.2
Via: SIP/2.0/UDP pc33.atlanta.com
      ;branch=z9hG4bK776asdhds ;received=192.0.2.1
To: Bob <sip:bob@biloxi.com>;tag=a6c85cf
From: Alice <sip:alice@atlanta.com>;tag=1928301774
Call-ID: a84b4c76e66710@pc33.atlanta.com
CSeq: 314159 INVITE
Contact: <sip:bob@192.0.2.4>
Content-Type: application/sdp
Content-Length: 131
(Bob's SDP not shown)
```

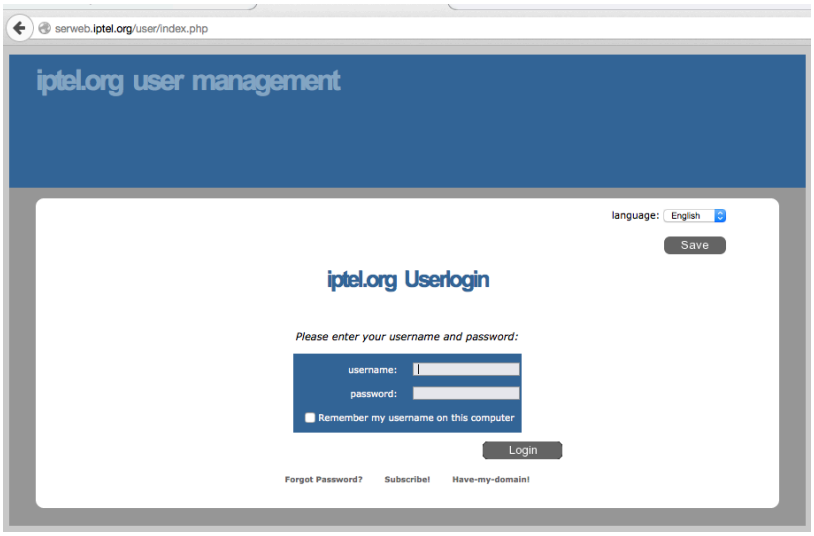
(source RFC 3261)

SIP Call Demonstration

SIP Call Demonstration



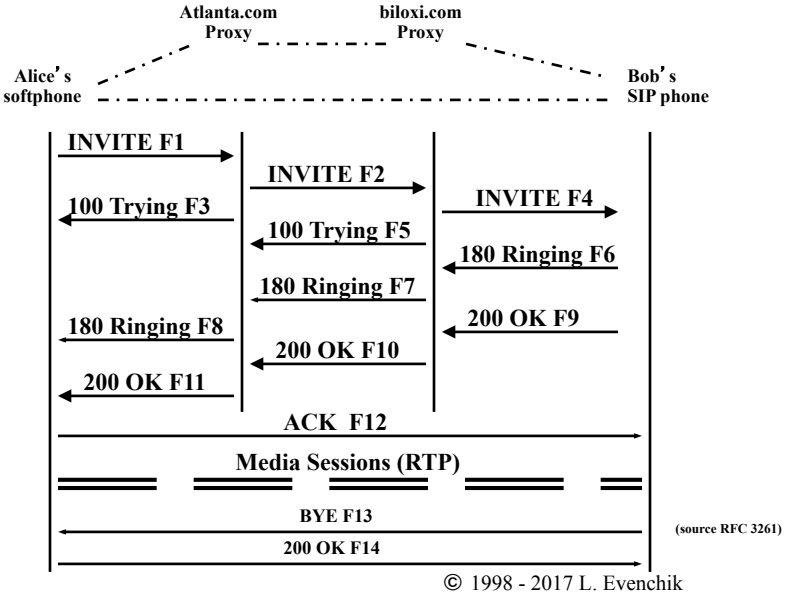
WWW.IPTEL.ORG Public SIP Service



SIP Registration and User Authentication

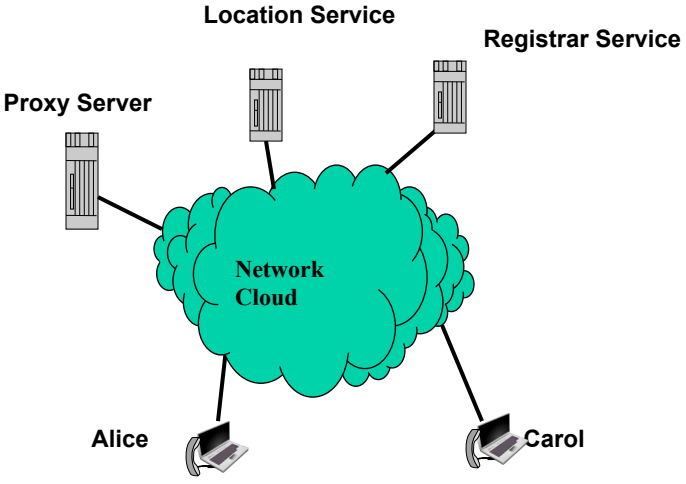
(Authentication uses the Hashing and Message Digests we learned in the previous lecture on security.)

SIP Trapezoid

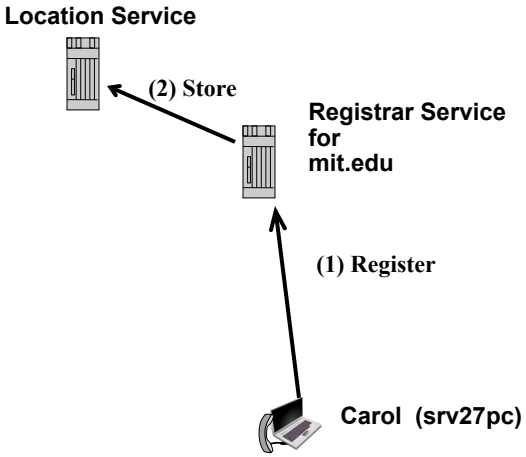


(source RFC 3261)

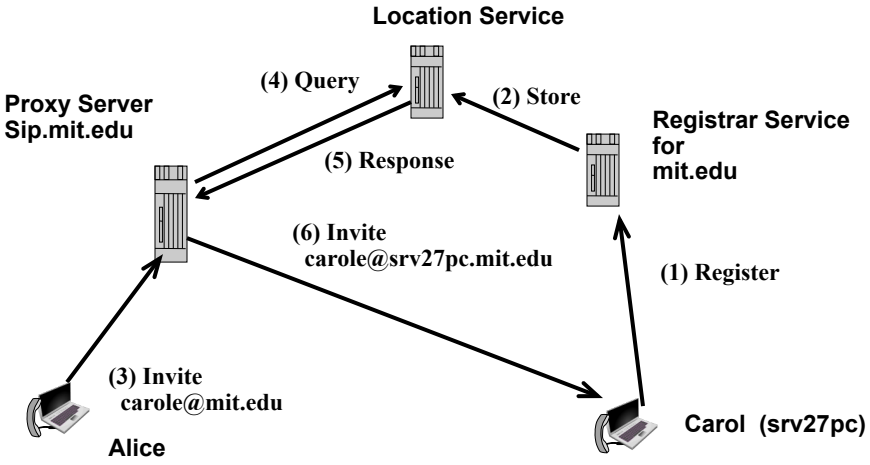
SIP Registration Building Blocks



SIP Registration
Carol registers in mit.edu



INVITE after SIP Registration



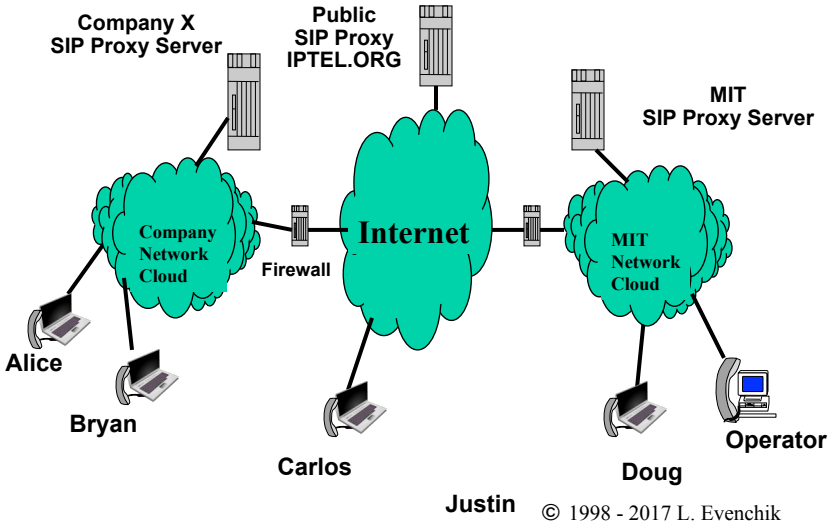
Domain Name System (DNS) and SIP

Finding the Proxy Server for a Remote Domain (not your own domain)

- Imagine you want to call bill@siplearn.com. How does your SIP proxy server find the IP address of the proxy that supports the siplearn.com domain.
- The approach is comparable to finding the foreign mail server for a domain.
- In other words, this will be done with DNS.

SIP Services Today

How does the SIP proxy at Company X find the Proxy for MIT?



DNS Resource Records (partial listing)

- A - specifies 32 bit IPv4 address
- AAAA – IPv6 address record
- MX - mail exchange record
- NS - specifies authoritative name server for a domain
- CNAME - canonical name, provides alias functionality
- HINFO - specifies limited host information
- SRV – identifies a specific service, such as SIP Server
- NAPTR – Name Authority Pointer, points to more info

Finding a Network Resource via DNS

- "A" records provide a mapping between names and addresses. This is what you would expect the DNS to handle.
- But how do you find a resource such a mail server when you don't know the name of the server?
- For example, email to webmaster@harvard.edu must be delivered to the mail server for Harvard, even though you do not know the name (or IP address) of the server that handles incoming mail.

Harvard DNS MX Query

```
fas% dig harvard.edu mx
;; QUESTION SECTION:
;harvard.edu.                IN      MX

;; ANSWER SECTION:
harvard.edu.         10800   IN      MX      20 mail.br.harvard.edu.
harvard.edu.         10800   IN      MX      10 netopc.harvard.edu.
harvard.edu.         10800   IN      MX      0 netop3.harvard.edu.

;; ADDITIONAL SECTION:
mail.br.harvard.edu.    10066   IN      A       128.119.3.169
netopc.harvard.edu.     10800   IN      A       128.103.1.37
netop3.harvard.edu.     10800   IN      A       128.103.208.29

ns1.harvard.edu.        10800   IN      A       128.103.200.101
ns.harvard.edu.         10800   IN      A       128.103.201.100
ns2.harvard.edu.        10800   IN      A       128.103.1.1
```

Finding a SIP Proxy Server

- Finding a SIP proxy server for a specific domain is comparable to finding a mail server for a specific domain.
- SRV records configured by the administrator of the domain are used by other proxy servers on the Internet to locate the domains SIP proxy server.
- NAPTR records are also used and they provide added flexibility to the type of SRV record
- For example, a SIP call to bill@siplearn.com must be sent to the proxy server for siplearn.com domain, even though the user (or the user's proxy server) does not know the name (or IP address) of the proxy server. DNS provides this needed address information.

DNS SRV Query for domain SIPLEARN.COM

```
cmd% dig _sip._udp.siplearn.com SRV
;; QUESTION SECTION:
;_sip._udp.siplearn.com.              IN      SRV

;; ANSWER SECTION:
_sip._udp.siplearn.com. 3600    IN      SRV   1 1 5060 asterisk.siplearn.com.

*** Then Another DNS lookup

cmd% dig asterisk.siplearn.com
;; QUESTION SECTION:
;asterisk.siplearn.com.     IN      A

;; ANSWER SECTION:
asterisk.siplearn.com. 3600 IN      A       aa.bb.cc.dd
```

DNS SRV Query
(Use DIG is Available)

```
nslookup
> set type=srv
> _sip._udp.siplearn.com

Answer is asterisk.siplearn.com
>
```


Session Description Protocol (SDP)

Example of SDP

Message body
Session Description Protocol
Session Description Protocol Version (v): 0
Owner, Session Id (o): bkermit2442 1803743 1804204
IN IP4 140.247.197.83
Owner Username: bkermit2442
Session ID: 1803743
Session Version: 1804204
Session Name (s): X-Lite
Connection Information (c): IN IP4 140.247.197.83
Connection Network Type: IN
Connection Address Type: IP4
Connection Address: 140.247.197.83
Media Description, (m): audio 12312 RTP/AVP 0 8 3 98 97 101
Media Type: audio
Media Port: 12312
Media Proto: RTP/AVP
Media Format: ITU-T G.711 PCMU
Media Format: ITU-T G.711 PCMA
Media Format: GSM 06.10
PLUS Others not shown here

Session Description Protocol (SDP)

- SDP is a general purpose protocol used to describe multimedia sessions. It is defined in RFC 4566.
- SDP is a format for session description, it is not a transport protocol and hence it must be carried by SIP messages.
- SDP is used to describe:
  - Session name and purpose
  - Contact and user information
  - Time information
  - Types of media to be used
  - Specific details for each media stream, including the Port #
- RFC 3264 defines offer / answer model for agreeing on media parameters.
- SDP was originally designed for describing multimedia session on the MBone test network. Some of the parameters you see in the spec relate to this history.
- SDP is being used for new protocols such as WebRTC.

SDP (2)

- As with SIP, SDP is text based (not ASN.1 encoded.)
- Each line of an SDP message is of the form
  <type> = <value>
  - <type> MUST be exactly one case-significant character
  - <value> is structured text whose format depends on <type>
- Common Session description lines
  - v= (protocol version)
  - o= (originator and session identifier)
  - s= (session name)
  - C= (connection/address information -- not required if included in all media)
- Common Media description lines
  - m= (media name and transport address)
  - c=* (connection/address information -- optional if included at session level for all streams)
  - b=* (zero or more bandwidth information lines)
  - a=* (zero or more media attribute lines)
  - *Plus many more*

*** means optional**

(source RFC 4566)

SDP Attributes and Media lines (3)

- Attributes ("a=")
  a=<attribute>
  a=<attribute>:<value>
  – Attributes are the primary means for extending SDP
- Media Descriptions ("m=")
  m=<media> <port> <proto> <fmt> ...
  – <media> is the media type
  – Media includes "audio", "video", "text", "application", and "message"
  – <port> is the transport port to which the media stream is sent
  – <fmt> is a media format description.
- We also need to look at MIME types

(source RFC 4566)

SDP for H.264 per Internet Draft
(abridged)

- SDP for codecs such as H.264 can have a large number of attributes

Offerer -> Answerer SDP message:
m=video 49170 RTP/AVP 100 99 98
a=rtpmap:98 H264/90000
a=fmtp:98 profile-level-id=42A01E; packetization-mode=0;
sprop-parameter-sets=<parameter sets data#0>
a=rtpmap:99 H264/90000
a=fmtp:99 profile-level-id=42A01E; packetization-mode=1;
sprop-parameter-sets=<parameter sets data#1>
a=rtpmap:100 H264/90000
a=fmtp:100 profile-level-id=42A01E; packetization-mode=2;
sprop-parameter-sets=<parameter sets data#2>;
sprop-interleaving-depth=45; sprop-deint-buf-req=64000;
sprop-init-buf-time=102478; deint-buf-cap=128000

SIP Offer/Answer

Offer/Answer Example 1 (What is the outcome?)

OFFER

```
v=0
o=alice 2890844526 2890844526
 IN IP4 host.atlanta.example.com
c=IN IP4 host.atlanta.example.com
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 132
a=rtpmap:31 H261/90000
a=rtpmap:132 H264/90000
```

ANSWER

```
v=0
o=bob 2808844564 2808844564
IN IP4 host.biloxi.example.com
c=IN IP4 host.biloxi.example.com
m=audio 49174 RTP/AVP 0
a=rtpmap:0 PCMU/8000
m=video 53456 RTP/AVP 132
a=rtpmap:132 H264/90000
```

(source RFC 4317)

Offer/Answer Example 2 (What is the outcome?)

OFFER

```
v=0
o=alice 2890844526 2890844526 IN
  IP4 host.atlanta.example.com
c=IN IP4 host.atlanta.example.com
m=audio 49170 RTP/AVP 0 8 97
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
a=rtpmap:97 iLBC/8000
m=video 51372 RTP/AVP 31 32
a=rtpmap:31 H261/90000
a=rtpmap:32 MPV/90000
```

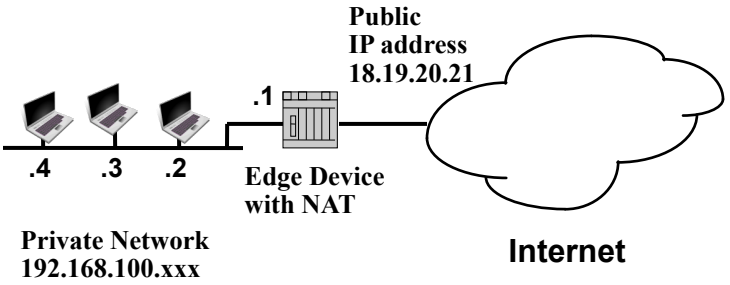
ANSWER

```
v=0
o=bob 2808844564 2808844564
IN IP4 host.biloxi.example.com
c=IN IP4 host.biloxi.example.com
m=audio 49172 RTP/AVP 0 8
a=rtpmap:0 PCMU/8000
a=rtpmap:8 PCMA/8000
m=video 0 RTP/AVP 31
a=rtpmap:31 H261/90000
```

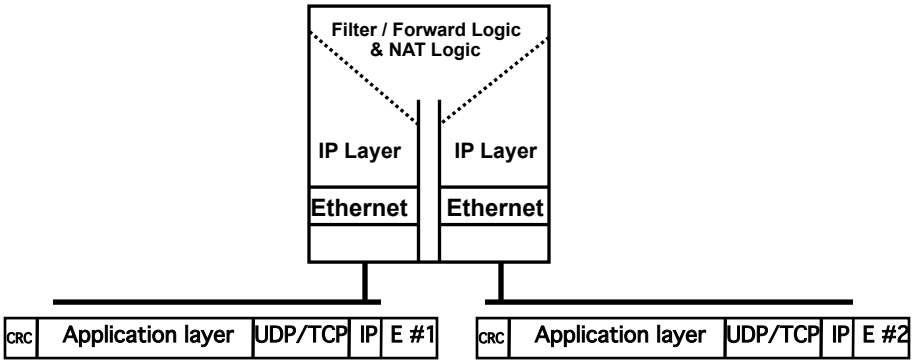
(source RFC 4317)

SIP and NAT and Firewalls

Network Address Port Translation (NAPT) Block Diagram



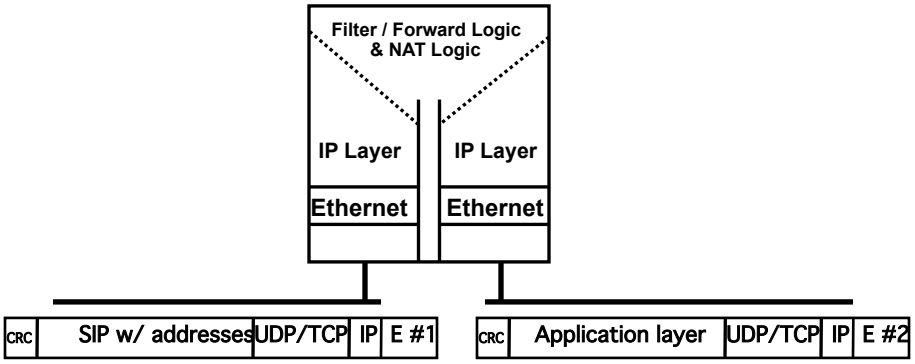
NAT and Port-Mapping Functionality Implemented by a Router or Firewall



IP Source Addr – Private
IP Dest Addr – Public
TCP Source Port – Private
TCP Dest Port - Public

IP Source Addr – Mapped (Public)
IP Destination Addr – Public
TCP Source Port – Mapped (Public)
TCP Dest Port - Public

NAT and Port-Mapping with Embedded Addresses in Application Layer



IP Source Addr – Private
IP Dest Addr – Public
TCP Source Port – Private
TCP Dest Port – Public
SIP – media addresses and ports in SDP, also addresses in contact field, etc.

IP Source Addr – Mapped (Public)
IP Destination Addr – Public
TCP Source Port – Mapped (Public)
TCP Dest Port – Public
SIP – what addresses are changed in the SDP and other fields?

SIP Use of STUN, TURN and ICE

- SIP clients and proxies use STUN, TURN and ICE to overcome the problems caused by NAT. Proprietary protocols are also used by many systems.
- STUN – protocol used by a client to determine the presence and type of NAT
- TURN – protocol for working with a media relay located on the Internet. A TURN relay replaces the need for an inbound call through at NAT. All of the clients place outbound calls to the TURN server instead.
- ICE – complex protocol for managing NAT traversal in protocols such as SIP (for VoIP) that use the offer/answer model.

WebRTC / RTCweb

WebRTC / RTCweb

- WebRTC is W3C work, RTCweb is IETF work
- Provides real time communication between HTML5-based browsers without the need for plugins. Also no need to install standalone VoIP app.
- Signaling for call setup and functionality for capability exchange/agreement are required and are typically implemented within the browser.
- Uses Secure RTP (SRTP) for media exchange
- Uses ICE/TURN/STUN for NAT traversal
- WebRTC is implemented today in most browsers and on mobile devices.
- “Click to Talk” customer service is a proposed use case.

Software Defined Networks (SDN)

Software Defined Networks (SDN) Characteristics (1)

- An SDN architecture explicitly separates the control plane and data plane. The data plane is responsible for packet forwarding.
- The control plane manages the functions of the data plane. The logical control system should provide centralized functionality, but should be physically decentralized, which is very difficult.
- There should be a well defined, standards-driven, interface between the control plane and data plane. (There are many standards being proposed for this interface today.)
- SDN supports virtualized network elements, including NFV for the switching and packet forwarding engine.
- There should be a well defined, standards driven, API between the control system and the application level, and between one SDN control system and other control systems. (Many different vendor and open standards are being proposed for this today.)

Software Defined Networks (SDN) Characteristics (2)

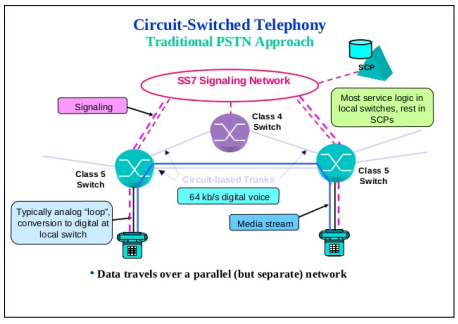
- Note that much of this system functionality, including the term SDN, and many of the individual building blocks, have been developed and implemented in various forms over the past 25 years (both proprietary vendor-specific approach and some open systems)
- The increasing availability of merchant-silicon chipsets for packet processing makes the separation between the control plane and data plane very cost effective today.
- Some approaches include a management plane which is different than the application level.
- An excellent reference on SDN is:
  http://arxiv.org/pdf/1406.0440v3.pdf

Standards Groups and Industry Associations Working on SDN and NFV

- IETF
- Open Networking Foundation (ONF), Industry consortium for OpenFlow
- Open vSwitch, Focus on production quality open virtual switches
- European Telecommunications Standards Institute (ETSI), NFV work
- OpenDaylight, Linux Foundation project
- Open Platform for NFV (opnfv.org)
- This is just a partial listing of the various groups!

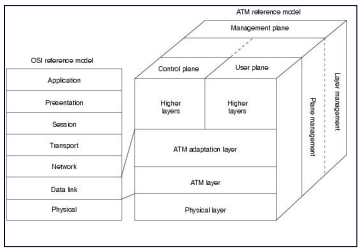
Previous Examples of Separating the Data and Control/Signaling Planes

Signaling System 7 (SS7), 1970s



Source: unknown

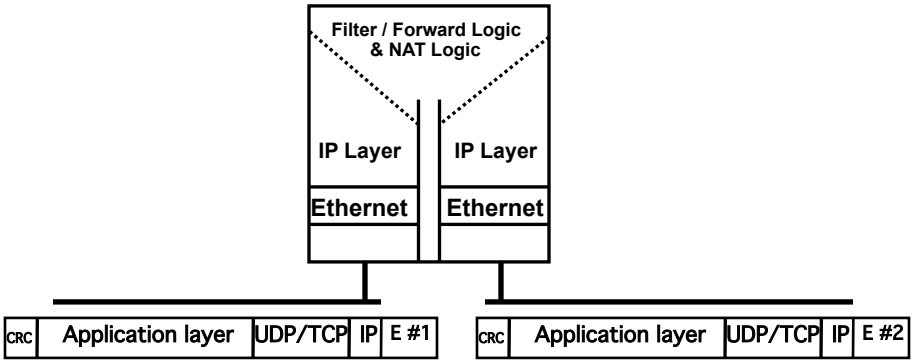
ATM Reference Model, early 1990s



Source: docwiki.cisco.com

- The Tymnet packet switching network had centralized routing and management control, circa 1980s
- Active Networks was a research effort in this area about about 20 years ago.

Our VERY Simplified Logical Diagram of Router Functionality
(Router providing NAT and Port-Mapping)

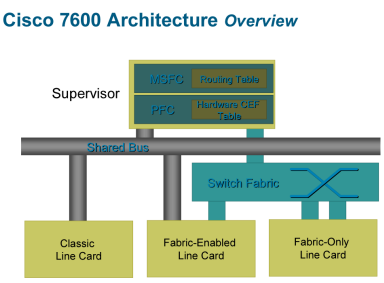


IP Source Addr – Private
IP Dest Addr – Public
TCP Source Port – Private
TCP Dest Port - Public

IP Source Addr – Mapped (Public)
IP Destination Addr – Public
TCP Source Port – Mapped (Public)
TCP Dest Port - Public

Common Router and Switch Architecture

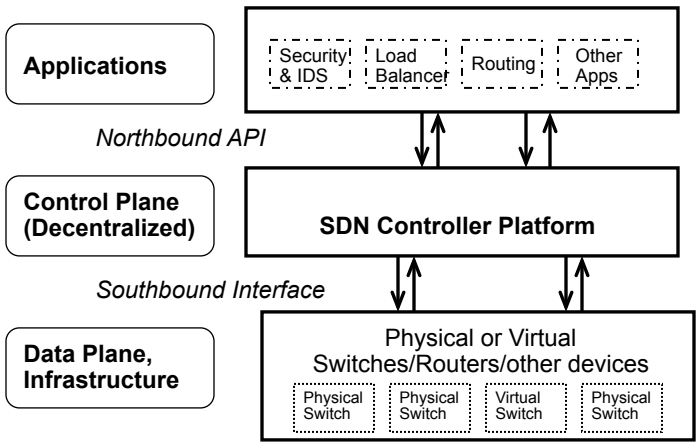
Routers have separated control and packet processing functionality for many years, but both functions typically resided in the same box



SOURCE: https://www.cisco.com/web/YU/events/expo_08/pdfs/Arhitektura_C7600_Aleksandar_Vidakovic.pdf

SOURCE: Juniper Network website, model MX480

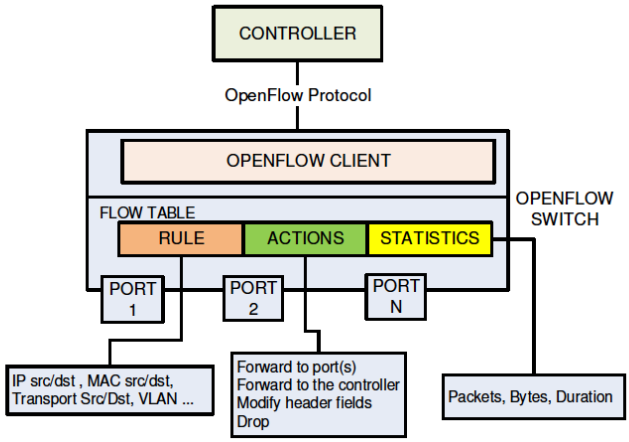
Basic SDN Architecure



OpenFlow Features

- OpenFlow is one example of a Southbound interface in SDN.
- There have been been multiple versions of the specification to date and the recent one is v1.5. (v1.6 is posted.) There were significant changes between versions in the earlier releases.
- In addition to their proprietary interfaces, many router vendors also support OpenFlow. (Note that the details vary by vendor.)
- OpenFlow defines tables, with entries such as source and destination addresses, and specifies the actions that should be taken by the forwarding device depending on the matches that are found when the packet is compared to the table.
- Longest match should be used first within a table, and tables entries can include * for wildcard.
- Packets are processed through a pipeline of tables.
- The tables do not cover everything. For example, there is no deep packet inspection (in v1.3), such as looking at different MIME types.

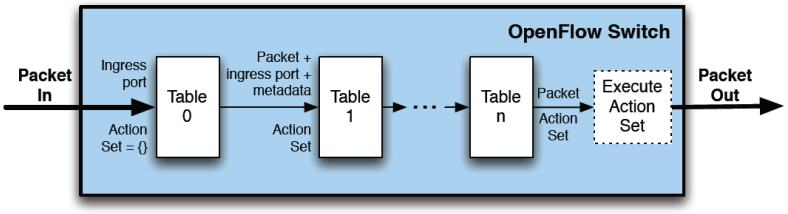
OpenFlow Model



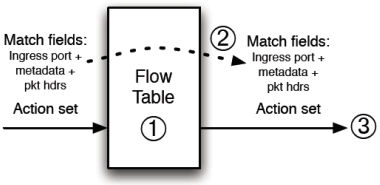
Source: IEEE Communications Surveys and Tutorials, 2014, 16 (3), pp.1617 - 1634, figure 2

Multiple Tables are Available

OpenFlow Switch Specification Version 1.4.0



(a) Packets are matched against multiple tables in the pipeline



① Find highest-priority matching flow entry

② Apply instructions:
- i. Modify packet & update match fields (apply actions instruction)
- ii. Update action set (clear actions and/or write actions instructions)
- iii. Update metadata

③ Send match data and action set to next table

Source: OpenFlow Switch spec V1.4

OpenFlow Required Match Fields

OpenFlow Switch Specification Version 1.4.0

Field		Description
OXM_OF_IN_PORT	*Required*	Ingress port. This may be a physical or switch-defined logical port.
OXM_OF_ETH_DST	*Required*	Ethernet destination address. Can use arbitrary bitmask
OXM_OF_ETH_SRC	*Required*	Ethernet source address. Can use arbitrary bitmask
OXM_OF_ETH_TYPE	*Required*	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_IP_PROTO	*Required*	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	*Required*	IPv4 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV4_DST	*Required*	IPv4 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_SRC	*Required*	IPv6 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_DST	*Required*	IPv6 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_TCP_SRC	*Required*	TCP source port
OXM_OF_TCP_DST	*Required*	TCP destination port
OXM_OF_UDP_SRC	*Required*	UDP source port
OXM_OF_UDP_DST	*Required*	UDP destination port

Table 11: Required match fields.

Source: OpenFlow Switch spec V1.4

NFV and SDN Model

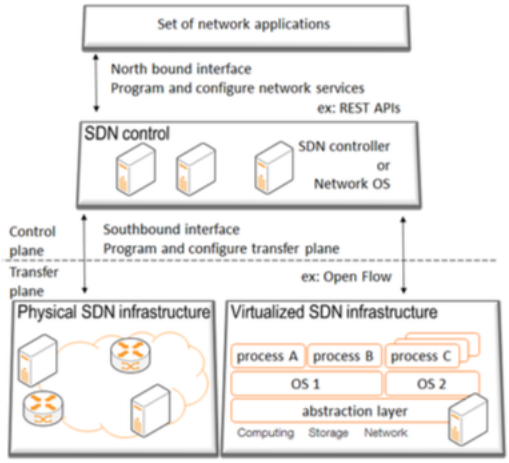


Figure 3 Software Defined Networking (SDN) overview

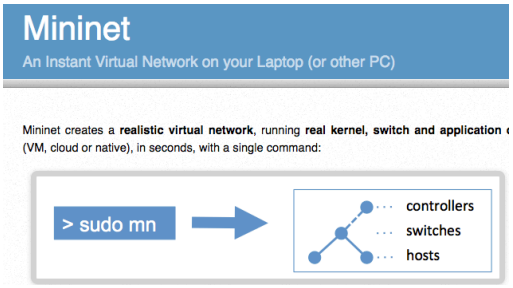
Source:
2015 18th International Conference on Intelligence in Next Generation Networks
978-1-4799-1866-9/15/$31.00 ©2015 IEEE

Network Virtualization and NFV

- Network Virtualization (NV) and Network Functions Virtualization (NFV) are not the same.
- Network virtualization separates the logical network from the physical network; it is an abstraction of the physical network. This concept predates SDN. For example, VLANs,VPNs and the MBone.
- Today, cloud data center providers use network virtualization to allow different customers to share a common network infrastructure. For example, VXLAN allows each user in the data center (a tenant) to have a virtual switched network.

Network Virtualization and NFV (2 of 2)

- NFV means that the network function (such as switching) is not being run on a network appliance designed and built specifically for that function.
- For example, NFV is used on VMs within data centers to allow clusters of VMs to communicate.
- If you want to create your own virtual SDN, please take a look at mininet.org.

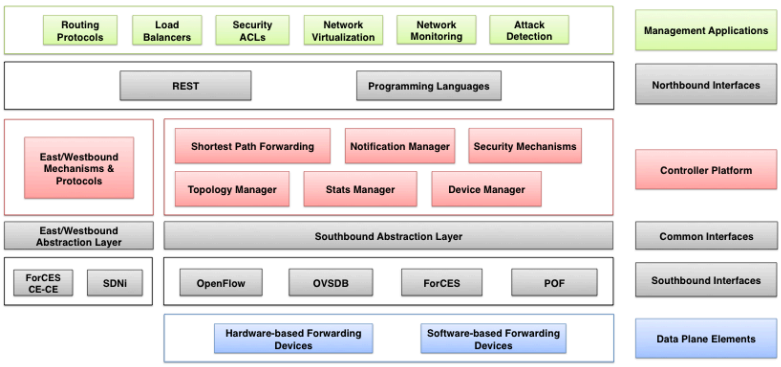


Models and Standards Groups

Standards Groups and Industry Associations Working on SDN and NFV

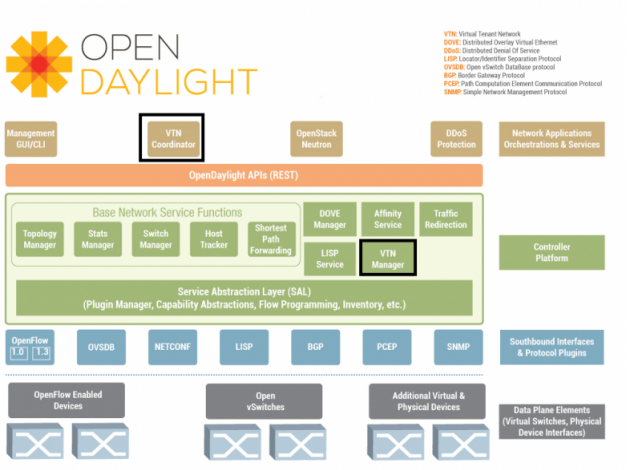
- IETF
- Open Networking Foundation (ONF), Industry consortium for OpenFlow
- Open vSwitch, Focus on production quality open virtual switches
- European Telecommunications Standards Institute (ETSI), NFV work
- OpenDaylight, Linux Foundation project
- Open Platform for NFV (opnfc.org)
- This is just a partial listing of the various groups!

SDN Architectural Model



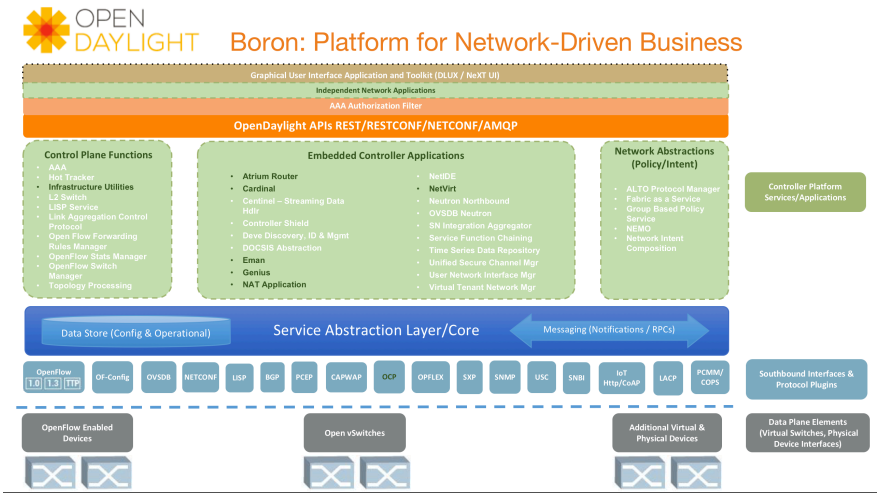
Source: IEEE, SDN Survey, Figure 8, http://arxiv.org/pdf/1406.0440v3.pdf

OpenDaylight SDN Architectural Model



Source: Open DayLight.org website, Nitrogen is the Sept. 2017 release

OpenDaylight SDN Architectural Model



Source: Open DayLight.org website

One Minute Wrap-Up

- Please do this Wrap-Up at the end of each lecture.
- Please fill out the form on the website.
- The form is anonymous (but you can include your name if you want.)
- Please answer three questions:
  - What is your grand "Aha" for today's class?
  - What concept did you find most confusing in today's class?
  - What questions should I address next time
- **Thank you!**