Señalización

Señalización dentro del Núcleo de Red

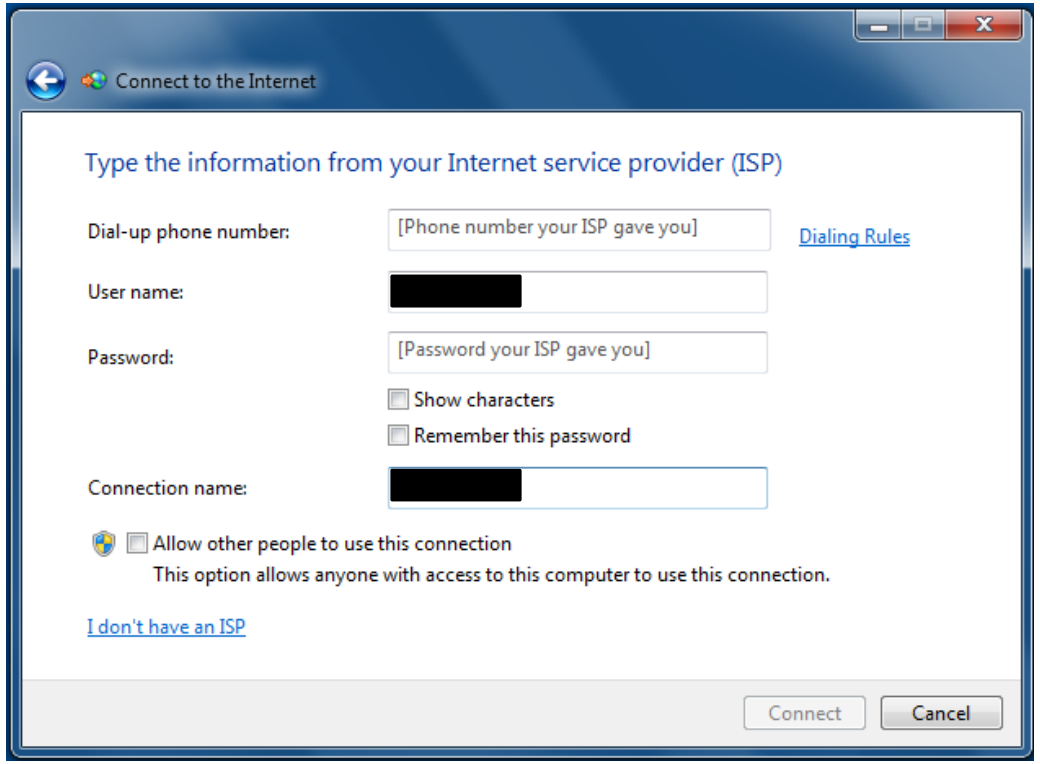
DIAMETER – INTRODUCCIÓN

Introducción

- ¿Qué es Diameter?
  - Es un protocolo de AAA: Authentication, Authorization and Accounting.
  - Trabaja en el layer de Aplicación del modelo OSI
  - Es un protocolo basado en mensajes:
    - Los nodos AAA intercambian información y reaccionan en base a Requests/Answers, y acuses de recibo (ACKs, NACKs)
    - Utiliza SCTP o TCP como protocolo de transporte, que se considera un transporte confiable.
  - Está especificado por la RFC 6733 de la IETF.

Introducción

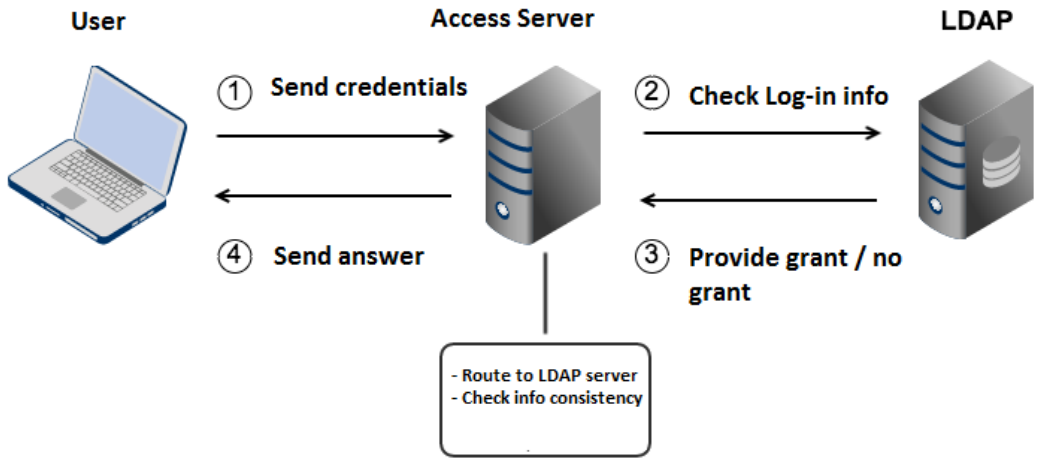
- ¿Para que protocolos AAA?
  - Cada vez que insertamos user/password para poder disponer de cierto servicio, el sistema nos está autenticando y autorizando



Introducción

- Caso de acceso via ISP:
  - ¿Qué es lo que ocurre cuando un ISP nos pide proveer credenciales para acceder a Internet?
  - La información necesaria para autenticar y autorizar al usuario no se encuentra en el poco seguro servidor de acceso.
  - Se elige en su lugar centralizar esta info en servidores LDAP (Ligthweight Directory Access Protocol).
  - Lo que falta en este proceso, son protocolos estandarizados que corran entre el server de acceso y el LDAP para intercambiar la info de AAA.

    → Para ello es que surgen los protocolos AAA como Radius y Diameter.



AAA e Historia

- Diameter fue inicialmente desarrollado por P. Calhoun, G. Zorn, and P. Pan en la RFC3588 (1998)
- El objetivo: proveer un marco de referencia para aplicaciones AAA que superase las limitaciones de su precursor, Radius
  - Radius era considerado poco flexible, poco seguro y poco confiable.
  - Radius no solucionaba bien los casos de acceso remoto, mobilidad IP, aplicación de políticas dinámicas y escalabilidad (ie roaming)

AAA e Historia

- En 2005, la 3GPP decide tomar a Diameter como uno de los protocolos de referencia para su set de estándares de IMS (IP Multimedia Sub-system)
  - Las interfaces Dx, Cx, Dh, Rf y Ro son implementadas sobre Diameter

- IMS apuntó inicialmente a ser la evolución del Circuit Switched CN para las tecnologías celulares legacy.
  - Sin embargo, logró rápidamente una gran penetración en el Core de las redes fijas
- La popularidad de IMS y la necesidad de mejoras en ciertas aplicaciones del protocolo (roaming, IP mobility) impulsaron una nueva revisión
- La RFC6733 (2012) es la última referente al Diameter Base Protocol

Señalización dentro del Núcleo de Red

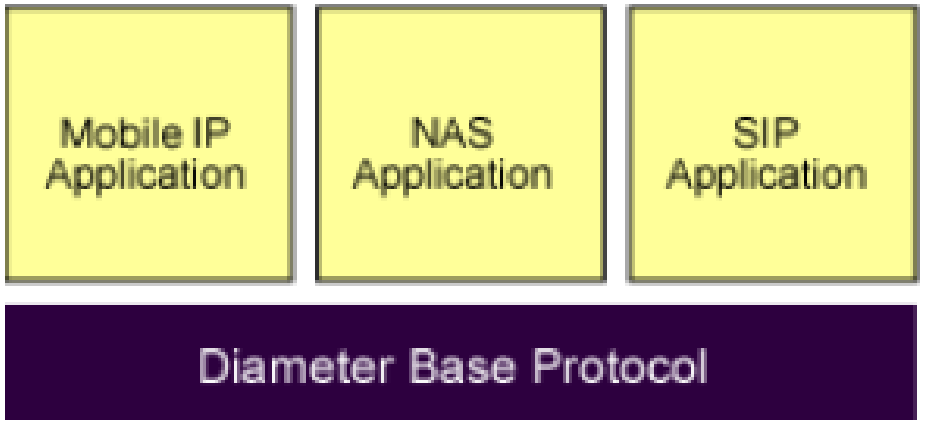
DIAMETER VS. RADIUS

Diameter vs Radius

- Diameter se considera el sucesor de Radius (de ahí el juego de palabras con sus nombres)
- Incluye mejoras en numerosos aspectos de su predecesor:
  - Confiabilidad y seguridad en el layer de transporte
  - Mecanismos de fail-over
  - Soporte de agentes Diameter
  - Soporte de mensajes iniciados por servers (no clientes)
  - Negociación de capacidades
  - Peer discovery y configuración
  - Soporte de transición (desde Radius)
  - Soporte de roaming

Diameter vs Radius

- Diameter es directamente compatible hacia atrás con Radius, pero se prevee un camino de upgrade en la RFC
- Diameter provee un protocolo AAA flexible y seguro, sobre el cual se pueden definir extensiones (aplicaciones) propias



Diameter vs Radius

1. Mecanismos de Fail-over
   - Radius no estandariza mecanismos de fail-over
     - Ni siquiera a nivel de transporte, ya que corre sobre UDP y por tanto no hay procedimientos de hand-shaking definidos
   - Por tanto, los comportamientos en casos de falla dependen de cada implementación y lo hacen poco flexible
   - Diameter soporta ‘acuses de recibo’ (ACKs) a nivel de capa de aplicación y define algoritmos de fail-over asociados a las respectivas máquinas de estado.

Diameter vs Radius

2. Seguridad en la capa de transporte
   - Radius define esquemas de autenticación e integridad a nivel de aplicación, pero solo requerido para mensajes de respuesta
   - Radius define el uso de IPSec, pero su soporte no es mandatorio.
   - Diameter provee soporte para TLS/TCP y DTLS/SCTP, estandarizando el soporte para seguridad en la transmisión, habilitando despliegues intra e inter dominio de AAA.
     → TLS: Transport Layer Security.

Diameter vs Radius

3. Confiabilidad del transporte
   - Radius corre sobre UDP: no hay mecanismo de retransmisión definido
   - Aunque se implementase a nivel de capa de aplicación, el funcionamiento no estaría estandarizado y daría lugar a diferentes comportamientos: es poco confiable
   - Diameter corre sobre TCP o SCTP, ambos considerados protocolos confiables de transporte (orientados a conexión).

Diameter vs Radius

4. Soporte de agentes Diameter
   - Radius no estandariza el soporte de agentes: Proxies, Redirects, Relays
     - La funcionalidad de cada rol de agente y que parte del mensaje Radius que es manipulada no esta definida explícitamente.
   - Como consecuencia, el comportamiento varía entre implementaciones
   - Diameter define el comportamiento de los distintos roles de agentes de forma explícita.
   - Por ejemplo, un Redirect agent no altera ningún AVP del mensaje Diameter

Diameter vs Radius

5. Soporte de mensajes iniciados por servers
   - Un mensaje iniciado por un server, implica que el servidor toma el rol de inciador de comunicación, en vez del cliente
     - Por ejemplo, ¿que tal si la sesión o conexión establecida entre cliente-servidor cae por cualquiera que sea la razón?
     - En dicho caso, el server puede enviar un mensaje al cliente para reconectarse o re-autenticarse.
   - Radius define mensajes iniciados por servers, pero su soporte es opcional.
     - Esto hace complicado el despliegue de funcionalidades como desconexiones/reconexiones/Re-autenticaciones iniciadas por el server
   - Diameter lo hace mandatorio y por tanto extensible a sus implementaciones

Diameter vs Radius

6. Negociación de capacidades
   - Radius no soporta la negociación de capacidades entre peers al momento de establecer la comunicación
   - Por ello, los servers y clientes no tienen forma de conocer de antemano sus respectivas capacidades
   - Pueden entonces no converger al setup de un servicio mutuamente aceptable
   - Diameter estandariza el soporte para negociar las capacidades, al momento de establecer la conexión.

Diameter vs Radius

7. Peer discovery y configuración
   - Las implementaciones de Radius típicamente requieren que la identidad de los respectivos peers (sea nombre o dirección IP) sea manualmente aprovisionada, junto con los shared-secrets
   - Esto implica una carga de trabajo extra en el despliegue:
     - Aprovisionamientos manuales
     - Compartir las claves de forma segura
     - Aceptar el riego inherente de que las claves no sean seguras (o únicas)
   - Diameter implementa el descubrimiento dinámico de peers a través de registros DNSs.

Diameter vs Radius

8. Soporte de transición (desde Radius)
   - Diameter y Radius no comparten un mismo PDU (Protocol Data Unit) y por ende Diameter no es compatible hacia atrás.
   - Sin embargo, se provee un camino de transición / upgrade para que ambos protocolos puedan coexistir en la misma red
     - Despliegue de Diameter en los nuevos elementos de red
     - Despliegue de Diameter en GWs que hablen ambos protocolos.
   - Esto le permite a un despliegue nuevo de Diameter poder interoperar con redes legacy, agregando un GW que traduzca a Radius.

Diameter vs Radius

9. Soporte de Roaming
   - Por temas de escalabilidad, el concepto de cadenas de proxies (via servers intermedios) es popular entre operadores que brindan servicios de roaming
   - Sin embargo, Radius no define explícitamente el rol de Proxy, y tampoco estandariza esquemas de seguridad en la capa de transporte.
     - Por ello, es vulnerable a ataques externos e internos, y suplantación de identidad
     - No apto para despliegues masivos de Roaming vái transporte no seguro
   - Diameter define no solo el soporte de Proxies, sino que agrega DTLS/TLS en la transmisión.
     - Los despliegues de roaming a gran escala con Diameter son comunes

Señalización dentro del Núcleo de Red

DIAMETER BASE PROTOCOL

Diameter Base Protocol

- Diameter provee las siguientes funcionalidades:
  - Capacidad de intercambiar mensajes y entregar AVPs (Attribute Value Pairs).
  - Negociación de capacidades entre peers.
  - Notificación de errores.
  - Extensibilidad (ver RFC2989).
    - A través de la inclusión de nuevas aplicaciones, comandos, AVPs, etc.
  - Servicios básicos para las aplicaciones, por ejemplo manejo de sesiones, tarificación, etc.

Transporte

- El perfil está definido en la RFC3539, y por cuestiones de confiabilidad solo **TCP y SCTP** son soportados como protocolos de capa de transporte
  - TLS (over TCP) y DTLS (over SCTP) cuando see requiere de seguridad

- La RFC define los puertos 3868 (s/seguridad) y 5868 (c/seguridad) para ser usados durante el establecimiento.

Transporte

- Se pueden iniciar conexiones de transporte desde cualquier puerto, pero es mandatorio estár listo para recibir conexiones en los puertos 3868 & 5868

- Las **conexiones de transporte Diameter no se redundan**, excepto que se trate de instancias distintas del peer Diameter.
  - Una conexión TCP
  - Una conexión SCTP single/multi-homed

Transporte

- Cuando se conoce un peer adyacente pero no existe conexión de transporte válida (por cualquiera sea la razón), se deben realizar intentos periódicos de re-establecerla.

- Una vez establecido el transporte, los peers pasarán a intentar establecer una conexión a nivel de aplicación (ie Diameter).

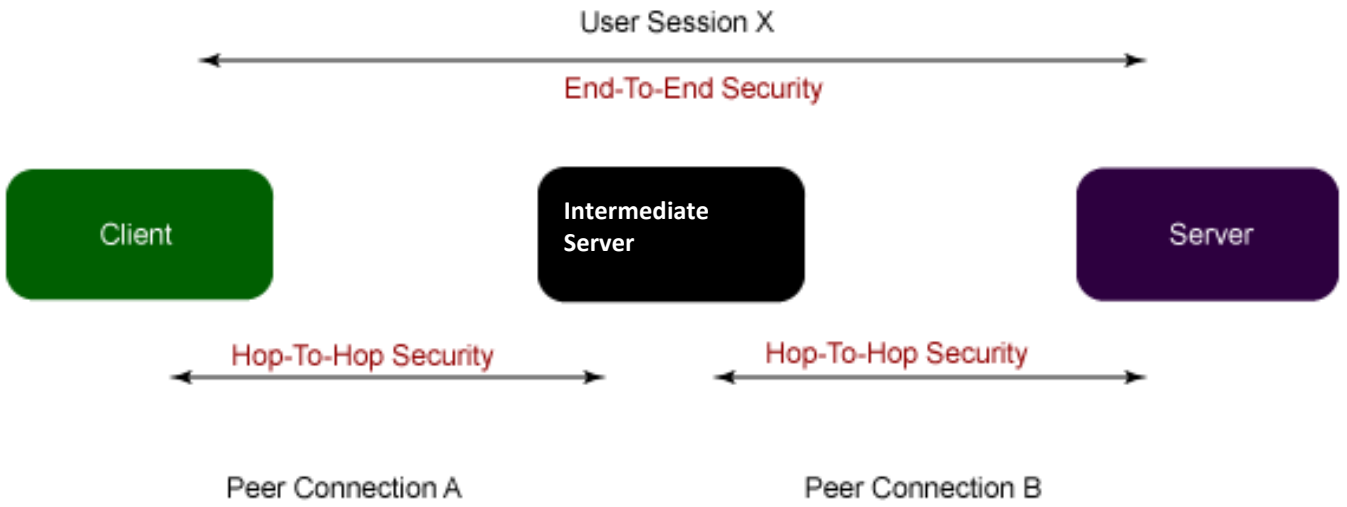
Conexiones vs. Sesiones

- Diameter implementa una arquitectura peer-to-peer

- Una **“Conexión”** Diameter existe a **nivel de aplicación** entre dos peers, sobre la cual se intercambian mensajes Diameter
  - Es mandatorio implementarla sobre transporte TCP o SCTP por confiabilidad
  - Puede existir más de una Conexión Diameter entre el mismo par de peers (por robustez)

Conexiones vs. Sesiones

- Una **“Sesión”** es una conexión lógica implementada a nivel de aplicación, entre un cliente y un server
  - Puede trascender varias conexiones Diameter
  - Se identifica vía un “Session-Id” único, generado por el cliente al momento del establecimiento



Conexiones vs. Sesiones

- **Sesión Diameter en detalle**
  - Una sesión Diameter comienza con el envío de un mensaje de Request de inicio de sesión desde cliente hacia servidor
    - Se incluye el “Session-Id” como id único, a ser utilizado mientras la sesión esté activa
    - Si el server decide autorizar la sesión, incluirá un timer (Authorization-Lifetime) en su respuesta: tiempo durante el cual la info de auth es válida
  - La re-autenticación del cliente puede ser requerida mientras la sesión este activa
    - Por expiración del timer o por requisito del Server
  - Una sesión puede ser terminada tanto por server como cliente.
    - Fin de sesión requerida por el cliente: “Session-Termination Request”
    - Fin de sesión requerida por el server: “Abort-Session-Request “

Nodos

- De **arquitectura P2P**, cada nodo que implementa el protocolo puede actuar como cliente o servidor.
  - El cliente Diameter
    - Es la interfaz hacia el “end-user”, y por ello recibe el request de acceso inicial (junto con información de autenticación; credenciales por ej)
    - Inicia la sesión hacia el server, para que la info sea validada
    - Debe soportar SCTP o TCP
  - El server Diameter
    - Recibe el request de inicio de sesión desde el cliente Diameter
    - Es el encargado de validar la información de autenticación y proveer al cliente de una respuesta.
    - La respuesta puede ser de éxito o rechazo (cualquiera fuese)
    - Debe soportar ambos, TCP & SCTP

Nodos

- La arquitectura es de cliente-servidor en un sentido amplio:
  - Un nodo puede actuar como cliente en algunos escenarios (o hacia algunos peers), y como server en otros
- Pero además de los roles tradicionales, se definen los **Agentes Diameter**.
- ¿Con que propósito?
  - Distribuir o agregar tráfico
  - Balancear carga
  - Esconder la topología interna por cuestiones de seguridad
  - Traducir protocolos AAA, etc

Agentes

- El requisito básico para un **Agente Diameter**, es mantener el estado de la transacción (para fail-over)
  - Transacción: Request > Answer
  - El “id” del mensaje de Request (Hop-By-Hop Identifier) es reemplazado por un id de transacción local único
- Un agente **“stateless”** solo mantiene el estado de la transacción (ie Request & Answer)
  - Al recibir el mensaje de Respuesta se limpia la transacción de la tabla
- Un agente **“stateful”** mantiene la información de sesión a través de las diferentes transacciones.
  - La sesión se considera activa (Session-Id válido) hasta que se libere explícitamente, mediante una notificación del cliente o server al agente.

Agentes

- Por sobre el requisito anterior, se pueden clasificar los agentes según sus "habilidades" adicionales: hay cuatro roles de agentes definidos en la RFC6733:
  - Relay
  - Proxy
  - Redirect
  - Translation

Agentes

- **Diameter Relay Agent**
  - Un Relay Agent rutea mensajes Diameter basado en información de destino contenida en el mensaje
  - La información de destino es:
    - AVP de Host de Destino
    - AVP de Dominio (Realm) de Destino
  - Cada RA tiene una tabla de ruteo para interpretar esta info
  - Siendo básicamente un router Diameter, sus principales usos son la agregación de mensajes de diferentes regiones/dominios, y seconder topología.

Agentes

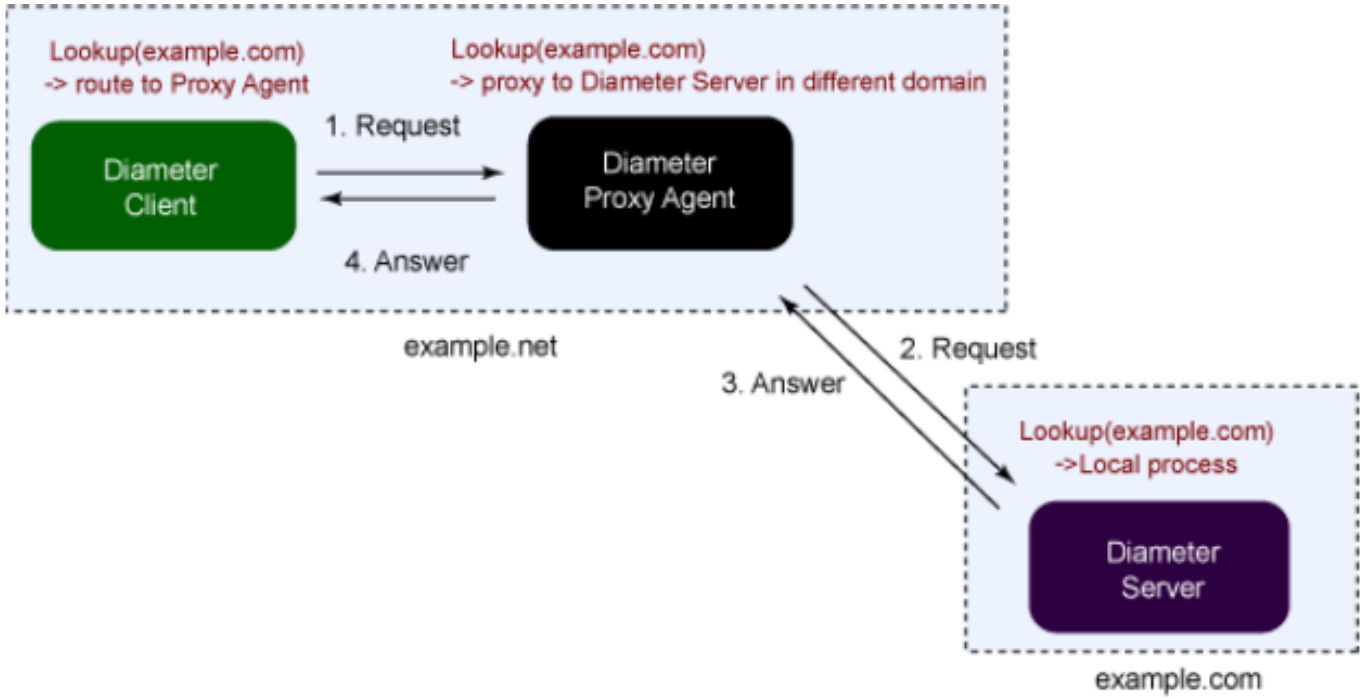
- **Diameter Relay Agent**
  - Un Relay modifica los mensajes Diameter solo insertando o removiendo información de ruteo
    - No puede modificar ningún otro campo del mensaje
  - Es un agente “stateless”
    - Solo mantiene estado de transacciones
    - No mantiene el estado de la sesión

Agentes

- **Diameter Proxy Agent**
  - Al igual que un Relay, un Proxy rutea mensajes Diameter en base a info de destino y tablas de rutas propias.
  - La diferencia esencial es que un Proxy puede modificar cualquier campo del mensaje Diameter.
  - Su principal uso es la aplicación de políticas o reglas en forma dinámica y servicios de valor agregado en general
    - Ej: Modificar destino en base a carga de tráfico o estado de la red

Agentes

- **Diameter Proxy Agent**
  - Para hacer esto efectivo, un Proxy debe ser “stateful” y llevar registro de las sesiones activas
  - Ejemplo:



Agentes

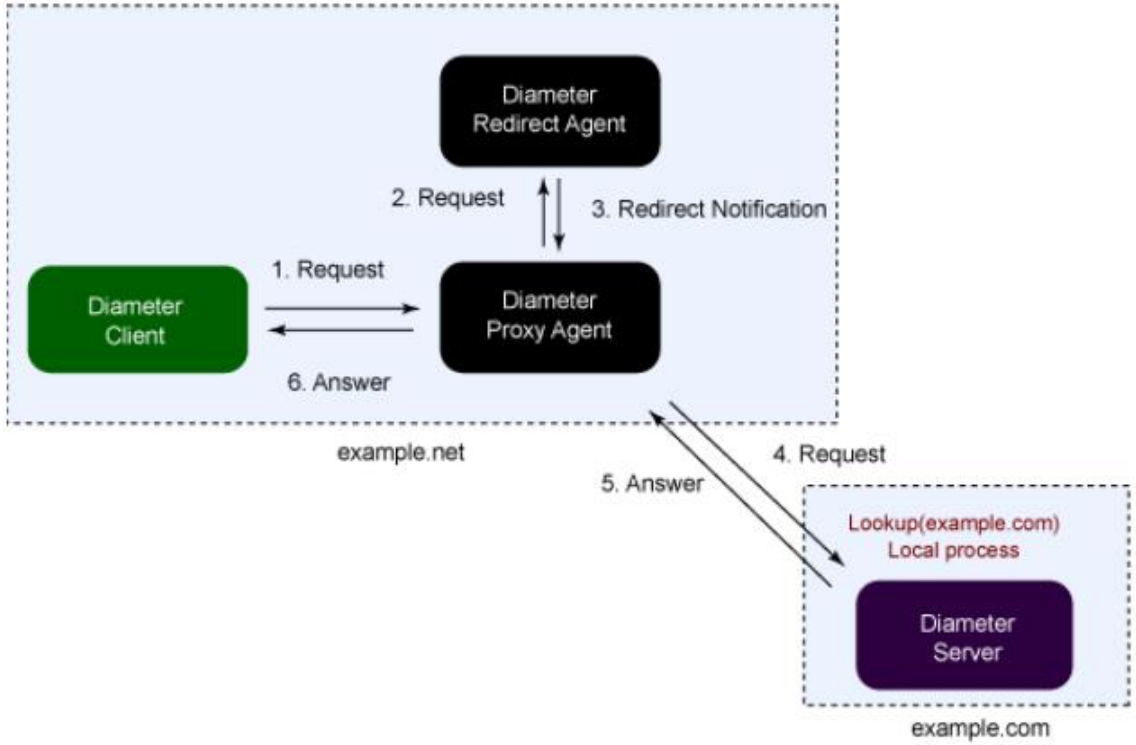
- **Diameter Redirect Agent**
  - El rol de Redirect es útil en escenarios en los cuales la información necesaria para rutear un mensaje Diameter precisa ser centralizada.
  - Un Redirect no forwardea el mensaje, solo provee la información de ruteo necesaria en su respuesta para que otro lo haga.
    - La información de ruteo está contenida en una tabla de Destino (Realm/Application) vs. Peer de salida, al igual que los casos anteriores.

Agentes

- **Diameter Redirect Agent**
  - El Redirect Agent centraliza la información de ruteo de su zona de influencia.
  - De esta forma evita la necesidad de mantenter actualizadas las tablas de ruteo de los restantes nodos.
  - Dado que un Redirect no reenvía mensajes (solo provee info de ruteo), no puede modificar los mensajes Diameter
  - Dado que un Redirect solo recibe Requests (y no Respuestas), solo puede ser “stateless”
    - No puede mantener estado de sesiones al no tener la transacción completa

Agentes

- **Diameter Redirect Agent**
  - Ejemplo:



Agentes

- **Diameter Translation Agent**
  - Como el nombre lo evidencia, este rol de agente tiene la capacidad de traducir Radius <> Diameter
  - Típicamente usados como servidores de agregación e integración con sistemas legacy (no migradas a Diameter aún)
  - Ejemplo:

Peer Discovery

- En Radius era necesario aprovisionar manualmente el hostname/IP address del nodo peer, así como el shared secret.
  - Costoso en términos operativos e inseguro
- Diameter estandariza el procedimiento de *Peer Discovery*, que como su nombre lo presenta, permite simplicar el procedimiento permitiendo descubrir al peer de forma dinámica
  - La opción de aprovisionamiento manual sigue estando disponible y debe ser soportada de todas formas
- Peer Discovery está basado en dos métodos
  - SRVLOC (Service Location Protocol)
  - DNS Queries

Capability Negotiation

- Previo a que dos peers establezcan una conexión Diameter, deben intercambiar las prestaciones (aka Capabilities) de cada uno
  - Por ej., para entender si ambos soportan la aplicación de interés

- Parte de la información intercambiada contiene:
  - Identidad
  - Versión de protocolo soportada
  - Aplicaciones Diameter soportadas
  - Mecanismos de seguridad soportados,
  - etc

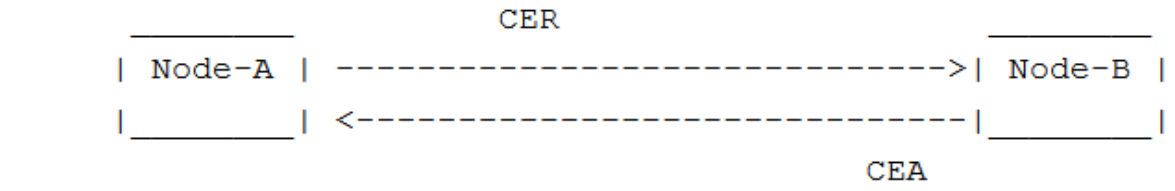
Capability Negotiation

- Este proceso se implementa mediante el intercambio de mensajes de Capabilities Exchange
  - El cliente envía un CER (Capability Exchange Request)
  - El server responde con un CEA (Capability Exchange Answer)

- Los mensajes de CE se intercambian entre peers directamente conectados
  - El objetivo: validar los parámetros de conexión Diameter y establecerla
  - Sobre esta conexión luego pueden establecerse sesiones entre Peers a más de un salto de distancia.
  - Por tanto, si un agente recibe un CER, NO debe restransmitirlo/reenviarlo

Capability Negotiation

- Ejemplo de flujo de CE




```
<CER> ::= < Diameter Header: 257, REQ >
    { Origin-Host }
    { Origin-Realm }
    { Host-IP-Address }
    { Vendor-Id }
    { Product-Name }
    [ Inband-Security-Id =s1 ]
    [ Inband-Security-Id =s2]
    [ Vendor-Specific-Application-Id =X]
    [ Vendor-Specific-Application-Id =Y]
    [ Vendor-Specific-Application-Id =Z]
```

```
<CEA> ::= < Diameter Header: 257 >
    { Result-Code =SUCCESS}
    { Origin-Host }
    { Origin-Realm }
    { Host-IP-Address }
    { Vendor-Id }
    { Product-Name }
    [ Inband-Security-Id =s1 ]
    [ Vendor-Specific-Application-Id =X]
```

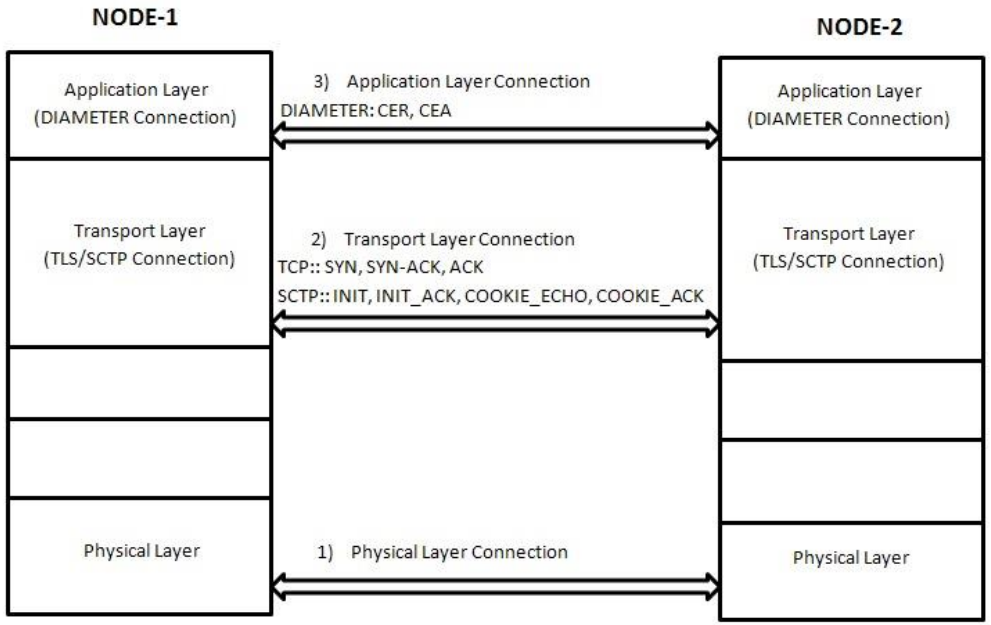
Capability Negotiation

- En casos en que peers no dispongan de aplicaciones o mecanismos de seguridad en común, el Server es el encargado de
  - Devolver un CEA no exitoso con su correspondiente causa de falla
    - “Diameter_no_common_application”, “Diameter_no_common_Security”
  - Terminar la conexión de transporte

- En caso de que el CER provenga de un peer aún no descubierto, el server puede
  - Descartar el mensaje sin responder al cliente
  - Descartar el mensaje con un CEA no-exitoso hacia el cliente
  - Agregar al peer a la tabla de peers y responder con un CEA exitoso

Capability Negotiation

1. Previo al CER/CEA que da inicio a la conexión Diameter, ambos peers deben intentar establecer el transporte
2. Mandatoriamente, se debe intentar con TLS o/TCP & DTLS o/SCTP, es decir, intentar establecer con el máximo grado de seguridad posible
   - El objetivo es proteger los CER/CEA que seguirán
   - Para ser compatibles con versiones anteriores del protocolo, los nodos pueden luego seguir negociando durante el CE
3. Luego del CER/CEA exitoso la conexión Diameter está establecida. Otras transacciones pueden dar lugar



DIAMETER Connection Establishment Process

Mensajes & Estructura

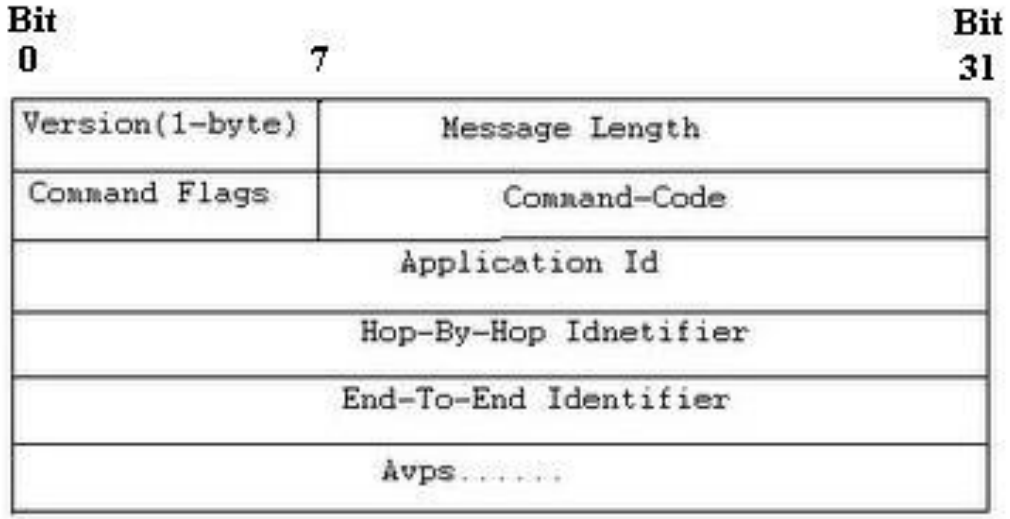
- Diameter es un protocolo basado en mensajes.
  - La información se intercambia en base transacciones (Request/Answer) entre cliente y servidor.

- Cada mensaje contiene un encabezado y un payload de datos variable.

- El encabezado identifica, entre otras cosas, el propósito de mensaje.
  - Diameter define varios tipos de mensajes Diameter, y el campo “Command Code” los identifica unívocamente.

Mensajes & Estructura: Header

- El header contiene además los siguientes campos:
  - Version: 1 Byte.
  - Message Length: 3 Bytes. Largo total del mensaje (incluyendo “payload”)
  - Command Flag: 1 Byte. R (Request), P (Proxiable), E (Error Response), T (Re-Transmission of RequesT)
  - Command Code: 3 Bytes. Utilizado para identificar el propósito del mensaje.
  - Application ID: 4 Bytes. Utilizado para identificar la aplicación para la cual este mensaje es aplicable.
  - Hop-by-Hop ID: 4 Bytes. Id único de transacción, utilizado para mapear respuesta con request.
  - End-to-End ID: 4 Bytes. Utilizado para detección de duplicados.



Mensajes & Estructura: Header

- Command codes (Diameter Base Protocol)

Message name	Abbreviation	Command code
Abort-Session-Request	ASR	274
Abort-Session-Answer	ASA	274
Accounting-Request	ACR	271
Accounting-Answer	ACA	271
Capabilities-Exchanging-Request	CER	257
Capabilities-Exchanging-Answer	CEA	257
Device-Watchdog-Request	DWR	280
Device-Watchdog-Answer	DWA	280
Disconnect-Peer-Request	DPR	282
Disconnect-Peer-Answer	DPA	282
Re-Auth-Request	RAR	258
Re-Auth-Answer	RAA	258
Session-Termination-Request	STR	275
Session-Termination-Answer	STA	275

Mensajes & Estructura: Header

- Command codes
  - Diameter es un protocolo extensible
  - No está atado a un aplicación específica corriendo sobre él.
  - En particular, **los mecanismos de Autenticación y Autorización varían en gran medida entre aplicaciones**.
    - Por ello, el Base Protocol no define Command Codes con estos fines: es reponsabilidad de cada aplicación introducir sus propios mensajes.
  - Diameter Base Protocol solo define entonces Command Codes de accounting (Tarificación).

Mensajes & Estructura: Header

- Algunos Command Codes introducidos por aplicaciones externas.

Command-Name	Abbr.	Code	Application
AA-Request	AAR	265	Diameter base
AA-Answer	AAA	265	Diameter base
Diameter-EAP-Request	DER	268	Diameter base
Diameter-EAP-Answer	DEA	268	Diameter base
Credit-Control-Request	CCR	272	Diameter Credit Control (4)
Credit-Control-Answer	CCA	272	Diameter Credit Control (4)
User-Data-Request	UDR	306	Sh (16777217)
User-Data-Answer	UDA	306	Sh (16777217)
Profile-Update-Request	PUR	307	Sh (16777217)
Profile-Update-Answer	PUA	307	Sh (16777217)
Subscribe-Notifications-Request	SNR	308	Sh (16777217)
Subscribe-Notifications-Answer	SNA	308	Sh (16777217)
Push-Notification-Request	PNR	309	Sh (16777217)
Push-Notification-Answer	PNA	309	Sh (16777217)
Update-Location-Request	ULR	316	S6a/S6d (16777251)
Update-Location-Answer	ULA	316	S6a/S6d (16777251)
Cancel-Location-Reqeust	CLR	317	S6a/S6d (16777251)
Cancel-Location-Response	CLA	317	S6a/S6d (16777251)
Authentication-Information-Request	AIR	318	S6a/S6d (16777251)
Authentication-Information-Answer	AIA	318	S6a/S6d (16777251)
Insert-Subscriber-Data-Request	ISD	319	S6a/S6d (16777251)
Insert-Subscriber-Data-Response	ISD	319	S6a/S6d (16777251)
Delete-Subscriber-Data-Request	DSD	320	S6a/S6d (16777251)
Delete-Subscriber-Data-Response	DSD	320	S6a/S6d (16777251)
Purge-UE-Request	PER	321	S6a/S6d (16777251)
Purge-UE-Response	PEA	321	S6a/S6d (16777251)
Notify-Request	NR	323	S6a/S6d (16777251)
Notify-Answer	NA	323	S6a/S6d (16777251)
Provide-Location-Request	PLR	8388620	3GPP-LCS-SLg (16777255)
Provide-Location-Answer	PLA	8388620	3GPP-LCS-SLg (16777255)
Routing-Info-Request	RIR	8388622	3GPP-LCS-SLg (16777255)
Routing-Info-Answer	RIA	8388622	3GPP-LCS-SLg (16777255)

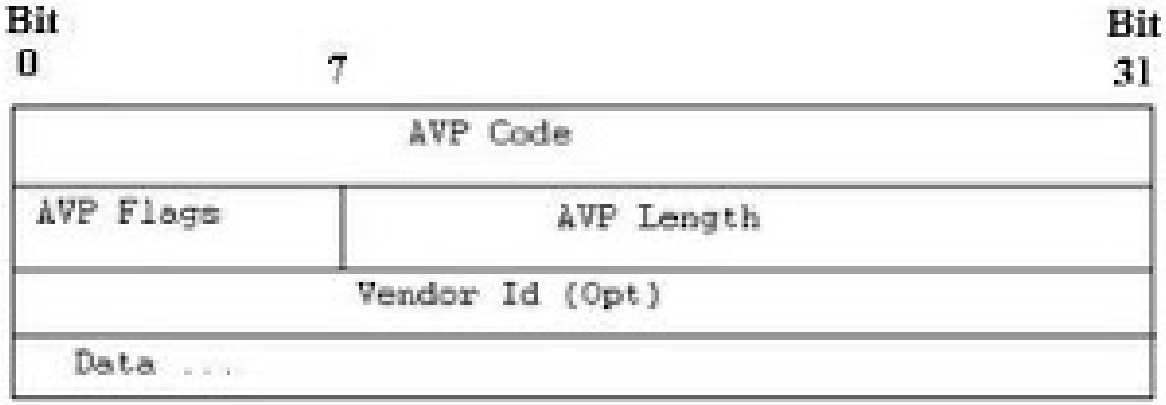
Mensajes & Estructura: AVPs

- ¿Que es un AVP?
  - Un Attribute-Value-Pair
  - Es la únidad básica de información de un mensaje Diameter.
  - Transporta tanto info del Protocolo base en sí, como para las aplicaciones que corren encima de Diameter.

- Mientras la intención del mensaje se identifica vía los Command Codes, son los AVPs los que entregan la **TODA** la información útil

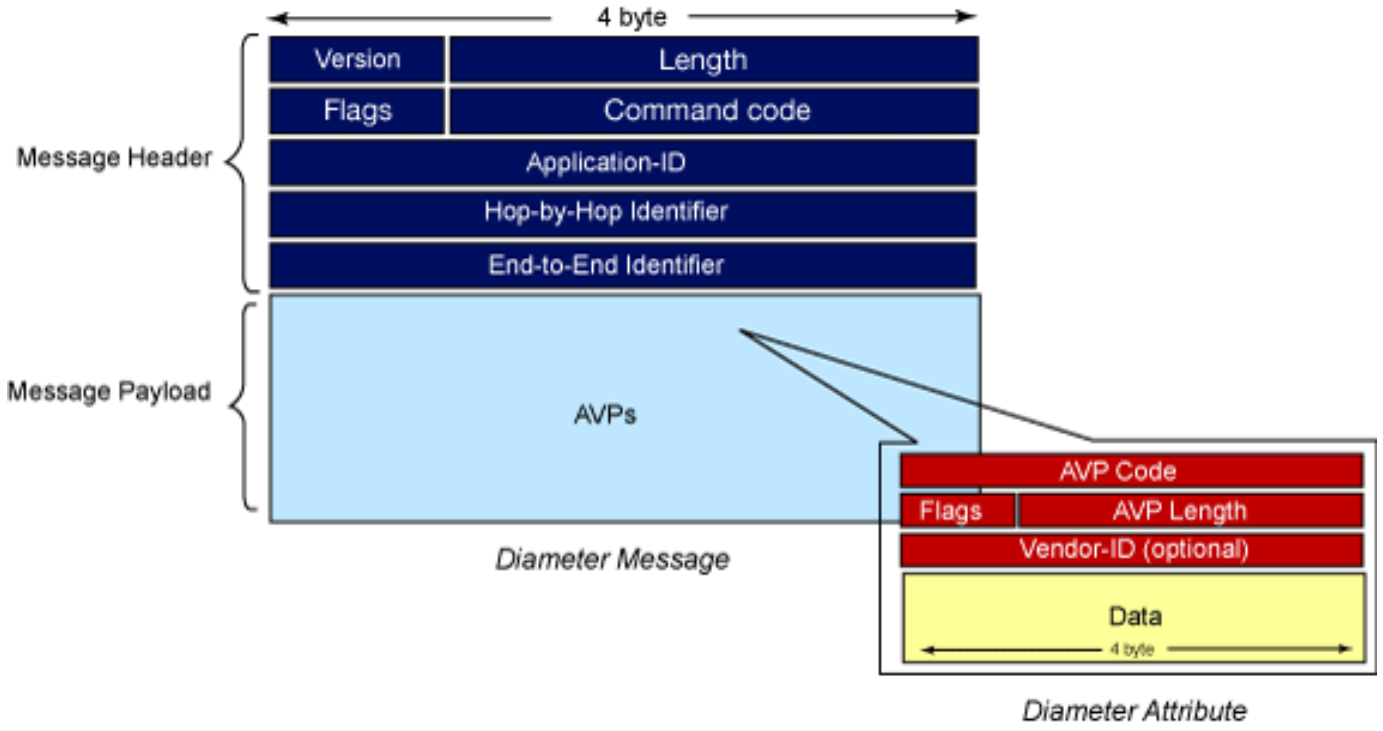
Mensajes & Estructura: AVPs

- Diameter define nativamente un conjunto de atributos comunes al protocol.
- Por otro lado, nuevas aplicaciones pueden introducir (definir) nuevos AVPs.
  - La única restricción es respetar el formato de AVP especificado en la RFC
  - El AVP Code junto con el Vendor ID identifican el atributo unívocamente.



Mensajes & Estructura: AVPs

- Inserción de AVPs dentro de un mensaje Diameter:



Señalización dentro del Núcleo de Red

APLICACIONES Y EJEMPLOS

Aplicaciones

- Hay numerosas aplicaciones que corren sobre él.
  El Application-ID es asignado por la IANA.
- Ejemplos:
  - S6a/S6d de 3GPP – App-ID: 16777251 (AuC & Auth en LTE)
- Diameter Credit Control – App-ID: 4 (Accounting en LTE, por ejemplo)

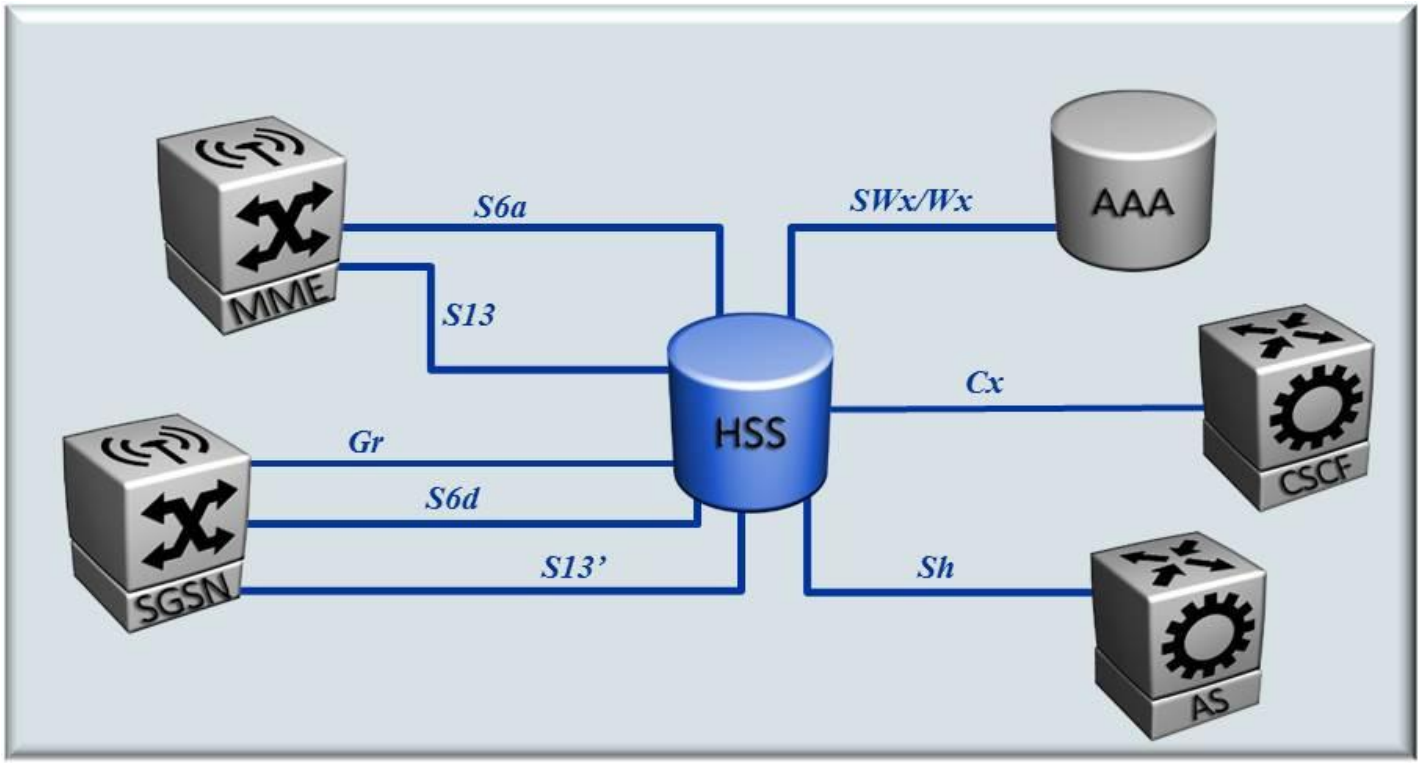
Aplicaciones

- Cada aplicación introducida debe definir sus propios Command Codes y AVPs
- Ej: Interfaz S6a (appliction-id : 16777251)

Command-Name	Abbr.	Code	Application
Update-Location-Request	ULR	316	S6a/S6d (16777251)
Update-Location-Answer	ULA	316	S6a/S6d (16777251)
Cancel-Location-Reqeust	CLR	317	S6a/S6d (16777251)
Cancel-Location-Response	CLA	317	S6a/S6d (16777251)
Authentication-Information-Request	AIR	318	S6a/S6d (16777251)
Authentication-Information-Answer	AIA	318	S6a/S6d (16777251)
Insert-Subscriber-Data-Request	ISD	319	S6a/S6d (16777251)
Insert-Subscriber-Data-Response	ISD	319	S6a/S6d (16777251)
Delete-Subscriber-Data-Request	DSD	320	S6a/S6d (16777251)
Delete-Subscriber-Data-Response	DSD	320	S6a/S6d (16777251)
Purge-UE-Request	PER	321	S6a/S6d (16777251)
Purge-UE-Response	PEA	321	S6a/S6d (16777251)
Notify-Request	NR	323	S6a/S6d (16777251)
Notify-Answer	NA	323	S6a/S6d (16777251)

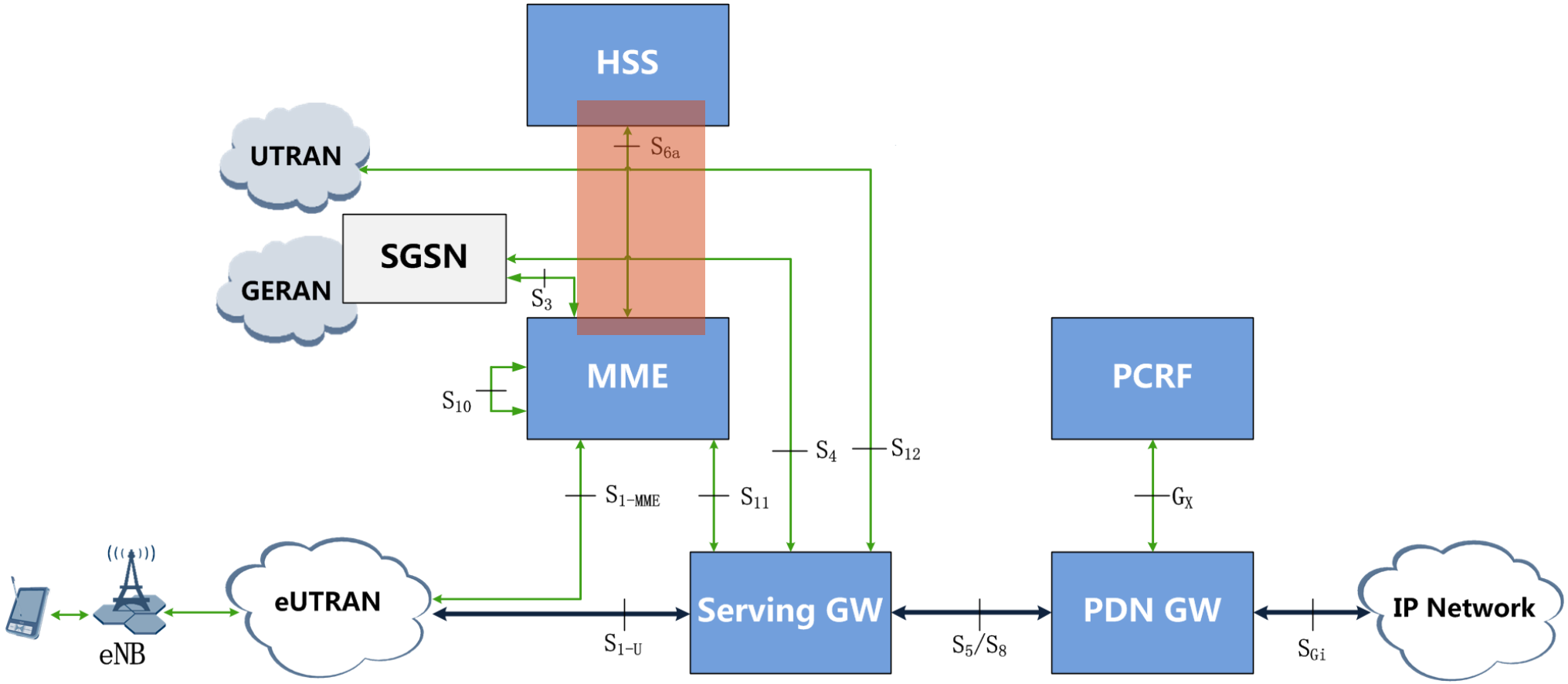
Ejemplos

- Hay varios ejemplos en redes móviles de 4a generación.



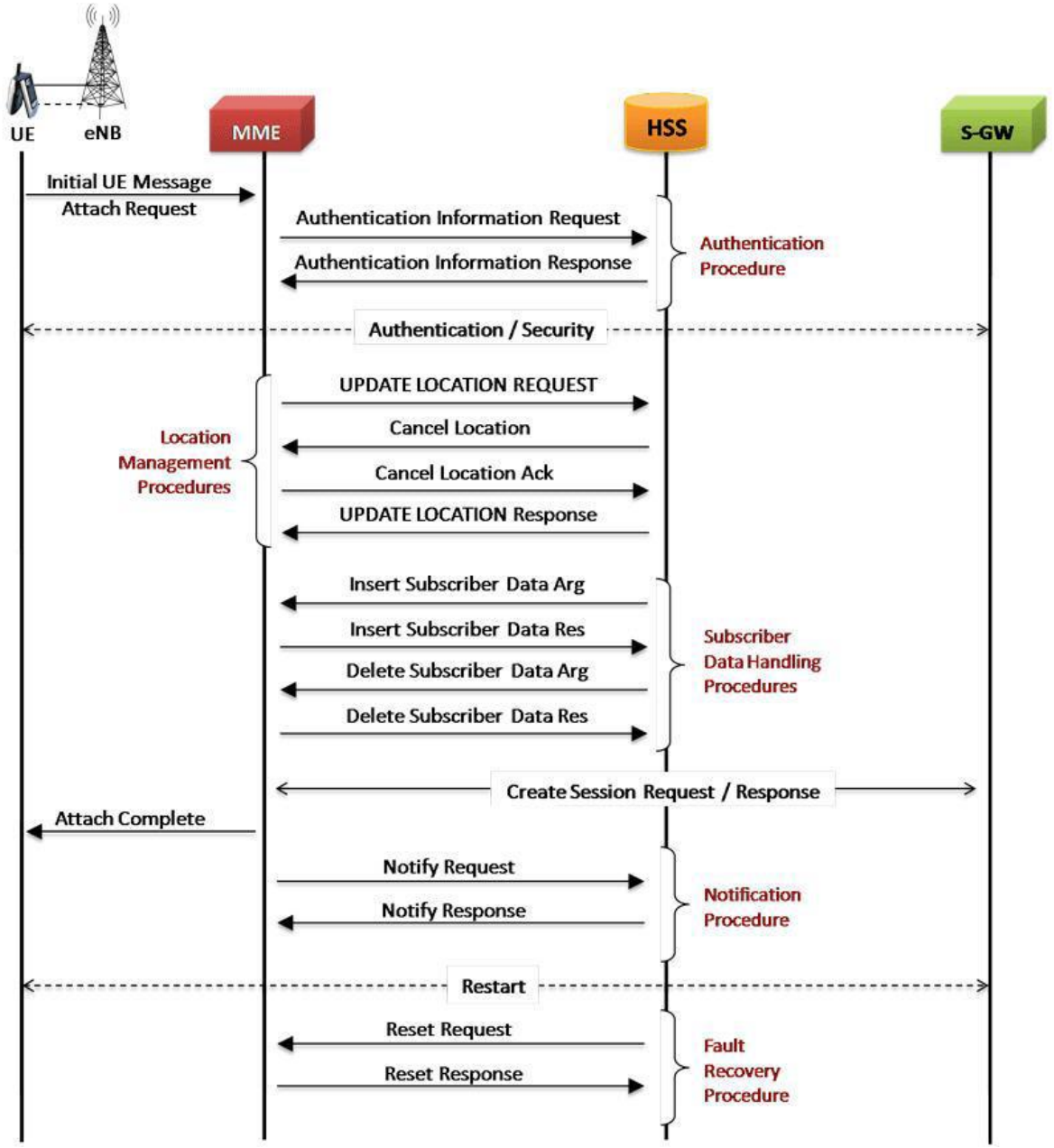
Ejemplos: S6a

- Hay varios ejemplos en redes móviles de 4a generación.



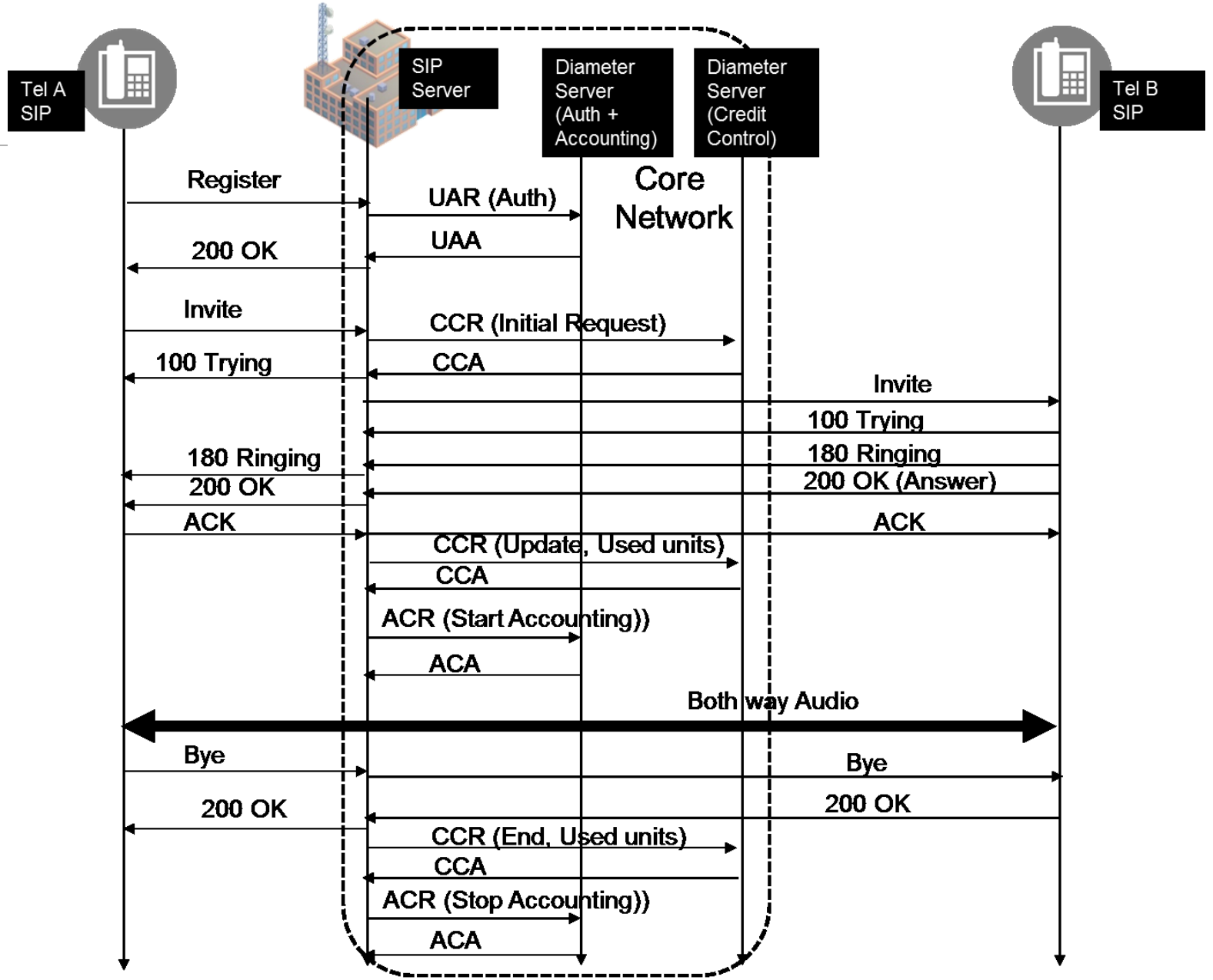
Ejemplos: S6a

- Call Flow: EPS Attachment



Ejemplos: SIP

- Call Flow: llamada SIP



Referencias

- https://tools.ietf.org/html/rfc6733
- https://tools.ietf.org/html/rfc6408
- https://tools.ietf.org/html/rfc3358
- http://www.3gpp.org/technologies/keywords-acronyms/109-ims
- http://www.ibm.com/developerworks/library/wi-diameter/
- http://diameter-protocol.blogspot.com/2011/03/introduction-to-diameter.html