NUMERO: 1607A - ANNO: 2015

A P P U N T I

STUDENTE: Gazzè

MATERIA: Tecnologie e Servizi di Rete. Prof.Baldi

1) Sia $V \subseteq \mathbb{R}^4$ lo spazio vettoriale delle soluzioni del seguente sistema:

$$\begin{cases} x + 2y + 2z + t = 0 \\ x + y + z = 0 \\ y + z + t = 0 \end{cases}$$

a) Trovare $\dim V$ e una sua base

b) Sia W lo spazio generato da $(-1, 1, 0, 0)$ e $(0, 0, 0, 1)$. Trovare, se esistono, tutti gli elementi di W che sono soluzioni del sistema

Svolgimento

a) la 1^a riga è pari alla somma della 2^a e della 3^a $\Rightarrow$ 1^a riga $= L(2^a, 3^a$ riga$)$

Il sistema si riduce quindi:

$$\begin{cases} x + y + z = 0 \\ y + z + t = 0 \end{cases}$$

da cui $\begin{cases} x = t \\ y = -z - t \end{cases}$

ovvero se $\exists\ C, D$ tali che

$$C\begin{pmatrix}1\\-1\\0\\1\end{pmatrix} + D\begin{pmatrix}0\\-1\\1\\0\end{pmatrix} = \underline{w} = \begin{pmatrix}-a\\a\\0\\b\end{pmatrix}$$

ovvero

$$\begin{pmatrix}1 & 0\\-1 & -1\\0 & 1\\1 & 0\end{pmatrix}\begin{pmatrix}C\\D\end{pmatrix} = \begin{pmatrix}-a\\a\\0\\b\end{pmatrix}$$

$$\rightarrow \left(\begin{array}{cc|c}1 & 0 & -a\\-1 & -1 & a\\0 & 1 & 0\\1 & 0 & b\end{array}\right)$$

dalla condizione di risolubilità del sistema deduco i valori di a e b

$R_2 \leftarrow R_2 + R_1$

$R_4 \leftarrow R_4 - R_1$

$$\left(\begin{array}{cc|c}1 & 0 & -a\\0 & -1 & 0\\0 & 1 & 0\\0 & 0 & b+a\end{array}\right)$$

$R_3 \leftarrow R_3 + R_2$

$$\left(\begin{array}{cc|c}1 & 0 & -a\\0 & -1 & 0\\0 & 0 & 0\\0 & 0 & b+a\end{array}\right)$$

2) Sia $f: \mathbb{R}^4 \to \mathbb{R}^4$ applicazione lineare definita $f(x,y,z,t) = (x+y, x+y, z-t, z-t)$

a) Trovare autovalori e autovettori di $\underline{\underline{M}}_f$

b) Dire se f è un endomorfismo semplice

Svolgimento

a) $\underline{\underline{M}}_f = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -1 & -1 \end{pmatrix}$

$$\det\left(\underline{\underline{M}}_f - \lambda \underline{\underline{I}}\right) = 0 \to \det\begin{pmatrix} 1-\lambda & 1 & 0 & 0 \\ 1 & 1-\lambda & 0 & 0 \\ 0 & 0 & 1-\lambda & 1 \\ 0 & 0 & -1 & -1-\lambda \end{pmatrix}$$

$$= \left[(1-\lambda)^2 - 1\right]\left[(1-\lambda)(-1-\lambda) + 1\right] = 0$$

$$= \left[\lambda^2 - 2\lambda\right]\left[-(1-\lambda^2) + 1\right] =$$

$$= \lambda^2\left[\lambda^2 - 2\lambda\right] = 0 \qquad \lambda^3(\lambda - 2) = 0$$

$\lambda_1 = 0$ 3 volte

$\lambda_2 = 2$ 1 volta

3) Sia $f: \mathbb{R}^2 \to \mathbb{R}$ tale che

$f(1,2) = 1$ $\quad \ker f = L(1,0)$

a) Si trovi la matrice di f rispetto alle basi canoniche

b) Si dica se f è suriettiva

c) $f^{-1}(2) = ?$

d) Si calcoli $f(2,2)$

Svolgimento

a) Base canonica di $\mathbb{R}^2 \Rightarrow \mathcal{C} = ((1,0),(0,1))$

$f(1,0) = 0$ perché $(1,0) \in \ker f$

$(0,1) = \frac{1}{2}(1,2) - \frac{1}{2}(1,0)$

$\Rightarrow f(0,1) = \frac{1}{2} f(1,2) - \frac{1}{2} f(1,0) = \frac{1}{2} - 0 = \frac{1}{2}$

$\Rightarrow \underline{\underline{M}}_f^{\mathcal{C},\mathcal{C}} = \begin{pmatrix} 0 & \frac{1}{2} \end{pmatrix} \in \mathbb{R}^{1,2}$

Sia $f : \mathbb{R}^3 \to \mathbb{R}^3$ definita da

$$f(a,b,c) = (a+b,\ 2b,\ -a+b+2c)$$

a) Determinare $\underline{\underline{A}} \in \mathbb{R}^{3,3}$ di f

b) Verificare che A è invertibile

c) determinare autovalori e autospazi di A

d) Il $\underline{\underline{P}}$ e $\underline{\underline{D}} \in \mathbb{R}^3$ con $\underline{\underline{P}}$ invertibile e $\underline{\underline{D}}$ diagonale tale che $\underline{\underline{D}} = \underline{\underline{P}}^{-1}\,\underline{\underline{A}}\,\underline{\underline{P}}$

Svolgimento

a) $$\underline{\underline{A}} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 2 & 0 \\ -1 & 1 & 2 \end{pmatrix}$$

b) $\det \underline{\underline{A}} = 4 \neq 0 \to \underline{\underline{A}}$ è invertibile

c) $$\det \begin{pmatrix} 1-\lambda & 1 & 0 \\ 0 & 2-\lambda & 0 \\ -1 & 1 & 2-\lambda \end{pmatrix} = 0$$

$$(1-\lambda)^1 (2-\lambda)^2 = 0 \qquad \lambda_1 = 1 \ \text{1 volta} \qquad \lambda_2 = 2 \ \text{2 volte}$$

$\lambda_1 = 1$ $$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1 & 1 & 2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \to \begin{cases} b = 0 \\ c - a + b = 0 \end{cases} \to a = c$$

29/09/14

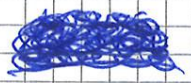
Tecnologie e servizi di rete

(netgroup.polito.it) 4° piano sopra COLIBÌ

Esame da 1 a 4 domande (Solo scritto)

Anche sotto forma di esercizi

Non sono disponibili esami di anni precedenti (Anzi sì da quest'anno)



IPV6 Nuova versione di IP pensata da 15 anni fatta un po' a soffietto IPV4

MPLS è un'evoluzione di IP migliore per le reti pubbliche

parleremo di VOIP e anche di VPN, cose che si usa per collegare per esempio reti diverse di un'azienda senza che i pacchetti si mischino con internet.

Poi parleremo di tecnologie per reti geografiche. Le reti stanno mangiando sempre di più e sono relativamente nuove.

Se avanza tempo parleremo di virtualizzazione delle reti.

IPV6

Serve perché usa spazio di indirizzamento più grosso. Gli indirizzi utilizzabili sono di più

In IPV4 sono 2^{32} meno le classi riservate perché gli indirizzi sono da 32 bit. Se aumentiamo i bit in IPV4 l'indirizzo non sta più nel pacchetto. IPV6 definisce come pacchetto fare.

stato lo spazio di indirizzamento. Questo quindi è la unica vera ragione per cui si è passati a IPV6.

02/10/14 Il processo di standardizzazione di IPV6 è iniziato nel 92 (documenti che definiscono formato pacchetto) ma molte cose sono state definite solo 5-6 anni dopo. In IPV6 rispetto alla V4 cambiano gli indirizzi, il formato dei pacchetti ma i concetti sono uguali e le soluzioni proposte per IPV6 sono state usate in IPV4. Arrivato a un "production" stage (pronto e poco testato) IPV6 era ormai accettabilmente evoluto.

Nello stack protocollare
Il livello 1 è hardware, il 2 dipende, anche misto, mentre da 3 in poi è software, quindi potenzialmente soggetto a bachi e c'è molto tempo per scrivere e rendere IPV6 accettabilmente affidabile.
Nel frattempo IPV4 era stato reso più sicuro ha ancora indirizzi a 32 bit (4 miliardi ma non si usano tutti)
Gli indirizzi IP assegnati alle stazioni sono solo quelli di classe A, B, C
Divisi in parte rete e host
L'indirizzo IP è diviso in due parti

Network	Host

Classe A => 8 bit rete, 24 host (0...) 128 reti 16777216 (-2) host
Classe B => 16 bit rete, 16 host (10...) 16384 reti 65536 (-2) host
Classe C => 24 rete, 8 host (110...) 2097152 reti 256 (-2) host
Per capire di che classe è un indirizzo si guardano i primi bit. Se il primo bit è 0 è di classe A quindi il primo byte è un prefisso e il resto è host. Gli indir. che iniziano per 111 non si usano per dare indirizzi agli host. Erano riservati, poi li hanno divisi e assegnati
Gli indirizzi che iniziano per 1110 sono per il multicast (classe D). Una staz. invia un pacchetto a più stazioni.
È un indirizzo di gruppo, non della stazione, quindi togliendo quelle che init. per 111 gli indir. per le staz scendono a circa 3,5 miliardi.
Gli ind. sono dati gerarchicamente. L'idea è che staz. collegate alla stessa rete ETHERNET abbiano la stessa parte di rete nell'indirizzo (Politecnico 193.13) e che non ci siano altre staz. collegate ad altre reti locali con quell'ind. di rete,

<u>Netmask</u>: si fanno dei prefissi la cui lunghezza non è 8, 16, 24 bit ma qualsiasi. Nella rete di prima con 3 soli host (a parte che era un prefisso di classe B, ma anche se fosse stato C) c'era un enorme spreco di indirizzi. Facendo un prefisso di 29 bit ci sono 8 possibili indir., 2 non si possono usare, ne uso 3 e ne spreco solo 3. Molto più ragionevole e si diminuisce lo spreco.

<u>Indirizzi privati</u>: ad un'azienda con 5000 calcolatori non do 5000 indirizzi. Sono indirizzi che non permettono di comunicare con l'esterno di una realtà privata. Posso scambiare pacchetti all'interno. Se tento di contattare GOOGLE non torna la risposta. Sono <u>specificati</u>, ad esempio per es. quelli che iniziano per 10 sono privati al POLI ma anche nella rete di GOOGLE quindi GOOGLE risponderebbe al mio o a uno dei suoi calcol. Cmq non tutte le staz. devono ricevere traffico esternamente. Per quelle "sì" invece si usa il concetto di NAT.

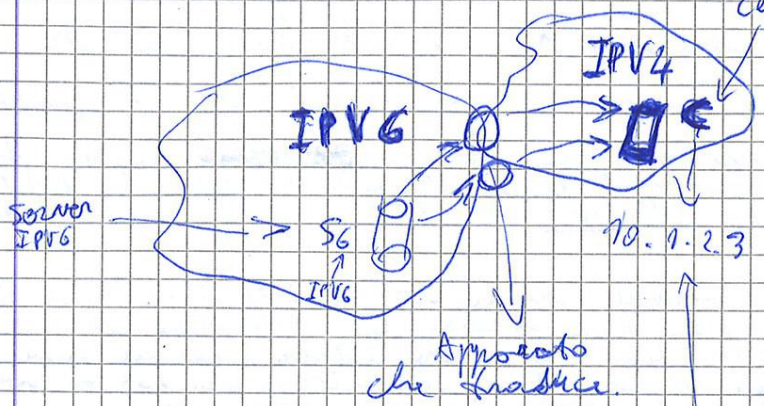
Con il NAT si è potuto rallentare l'esaurimento degli IP ma esso ha delle pecche se un server remoto deve contattare un computer locale che non ha fatto richieste al momento. Va bene quindi se i client contattano dei server. Se un server remoto deve contattare un calcol. locale anche se sapesse che deve contattare il NAT, poi come si fa? Un telef. deve poter fare e ricevere chiamate quindi non può avere un indirizzo privato. Si possono usare gli ALG (Application Layer Gateway) detti anche PROXY non è una soluz. pulita. C'è anche il <u>probl. della scalabilità del routing</u>. Facendo gli indirizzi gerarchici si ha una scalabilità ma fino ad un certo punto. Poi di prefissi che identificano una rete finora ce ne sono molti. Il POLI ha indir. di classe B e poi usa indir. più lunghi per le varie reti interne ma dall'esterno il pref. di 2 byte identifica tutte le destinaz. del POLI. Più la rete Internet cresce più crescono le tab. di routing (probl. di mem., capac. elaborativa). I router costruiscono le tab. scambiando info coi altri router. Una cosa è scambiare info su 100 dest. un'altra su 100K. Molta capacità trasmissiva della rete verrebbe occupata così a scapito dei pacchetti veri e propri. Ogni destinazione nuova fa sì che cambino i percorsi e quindi altro scambio di info. Se ci sono tante destinaz. la prob. che cambi qualcosa aumenta. E' un probl. non tanto per i router di periferia (quello del POLI deve sapere tutto del POLI, poi gli basta buttare i pacchetti all'esterno) ma per quelli in mezzo alla Internet. Un grafico mostra quanto sono cresciute le tab. di routing di questi router negli ultimi anni. RIB (sta per Routing Information Base) è un altro nome per <u>routing table</u>. BGP è un protocollo di routing. La crescita è stata più che lineare. Si è passati da tabelle di 15000 righe a 400000 e nonostante le migliorie ai router è un problema. Per ridurre ulteriormente il num. di righe si può <u>raggruppare ulteriormente</u>. Posso raggruppare ad esempio 1.2.1.0/24 e 1.2.2.0/24 in cui il prefisso sono i primi 3 byte in 1.2.0.0/16. Tutto ciò che inizia per 1.2 viene mandato in una certa direzione. E' fattibile con il CIDR (Classless Inter-Domain Routing), classless perché ormai non interessa il concetto di classe. Il probl. è se tutto ciò che è 1.2 <u>non</u> è nella stessa direzione, perché i prefissi sono stati dati con differenti criteri. <u>Non si poteva fare in modo razionale</u> perché con le gerarchie paghi sprecando. Tutto ciò che è 1.2 lo facciamo stare in una certa zona della rete al massimo in quella zona non c'è bisogno di 2^{16} indirizzi, quindi si spreca perché non si può assegnare i restanti da un'altra parte. L'unico modo per uscire da sta situazione è ripartire da zero con IPV6. Ci sarà un periodo di transiz. tra IPV4 e IPV6. A parte dello sforzo per standardizzare IPV6 è stata profusa in ciò. La transiz. durerà anni. Furono fatte diverse proposte all'inizio: TUBA che usava CLNP come nuovo IP, CATNIP, SIPP, ma vinse quella meno innovativa: SIPP. Le altre creavano troppa discontinuità nel mondo di Internet.

H è l'<u>efficienza di indirizzamento</u> in diversi tipi di rete che fanno l'indirizzamento gerarchico.

$$H = \frac{\log_{10}(\text{num. ind in uso effettivamente})}{\text{num. bits indirizzo}} \Rightarrow 0{,}22 \le H \le 0{,}26$$

Pensando di fare una rete che riuscisse a dare indirizzi a 1 milione di miliardi di stazioni 10^{15} e considerando l'H più piccola uscivano fuori 68 bits e hanno arrotondato a 128 bits.

Una parte dello spazio di indirizzamento è stata riservata per la transizione tra IPV4 e IPV6



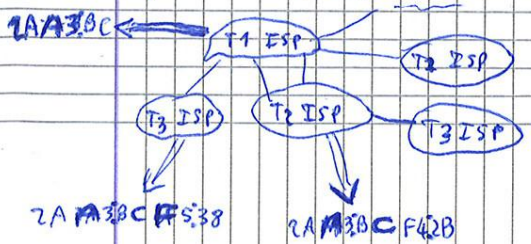
Sistema deve usare una corrispondenza in modo tale che S6 con IPV6 riesca ad indirizzare C, e viceversa C con IPV4 S6

La rete IPV6 che deve mandare il server S6 fa con IPV4 e la rete IPV6 con IPV6. Deve esserci un indirizzo IPV6 equivalente a 10.1.2.3. Basta mettere i 32 bit dentro i 128 e ignorare i restanti 96 o un valore fisso. Sono gli ind. IPV6 che hanno i primi 80 bit a 0. Con il metodo IPV4 mapped addresses gli altri 96 sono a 1, mentre con IPV4-compatible sono anch'essi a 0. Siccome gli ultimi 4 byte rappresentano l'ip IPV4, si possono scrivere in decimale. Ci sono 2 modi diversi perché probabilmente ciò permette scelte diverse per la traduzione. Anche lo spazio per la traduz. non è tutto dedicato, ma ce ne sono molti per traduzione. Il resto dello spazio di ind. non viene anch'esso usato tutto ma si apre la parte che inizia per 0x001

Gli indirizzi che iniziano per 0x001 sono unicast globali e le assegna lo IANA. Devono essere assegnati (congiunti in modo coordinato dalla IANA e enti dell'economia) in modo fisso per favorire l'aggregazione per ridurre le dim. delle tab. di routing e seguire la topologia della rete come non più per i numeri di telefono (assegnati in modo geografico. Invece fisso perché c'era l'operatore tel. per nazione). Per internet si segue la gerarchia dei service provider che definisce la topologia.

PEERING → Collegamento tra 2 ISP di pari livello (ex. due TIER 1) per mutuo interesse

CUSTOMER/PROVIDER → E' intorno da un ISP di livello più basso connettersi con un ISP di liv. superiore per smistare il suo traffico a livello maggiore. Quello più basso paga il servizio di interconnettività a quello superiore. Alcuni TIER 2 possono essere collegati a più TIER 1 quindi è più mesh che gerarchica la rete.



Agli ISP che vogliono connettersi al T1 ISP, questi assegna un prefisso, p.e. 2AF3BC e poi ognuno avrà la restante parte diversa quindi si monta un filtro per il mondo basterà memorizzare in tabella l'indirizzo 2AF3BC. I prefissi diventano più grandi scendendo di livello.

6x001 unicast

Per i global unicast ← il prefisso è 64 bit e anche l'interface identifier è lungo 64 bit.

Qui praticamente c'è una sola classe (che non viene chiamata classe) e non c'è il concetto di Netmask. Il sostituto della NM è proprio la lungh. del prefisso. Quando si dice che un router ha come destinazione un prefisso in IPV4 vuol dire che deve conoscere un ind. e la Netmask che gli dice di quell'ind. quale è il prefisso. In IPV6 basta il num. di bit del prefisso che deve sapere il router.

Gli hosts dello stesso link si chiamano On-link

Di diversi link si dicono OFF-link.

FC/7 è un numero a 8 bit ma il 7 indica di guardare solo i primi 7. Vediamo come è diviso gerarchicamente il prefisso di un G.U.

3 bit	13 bit	32 bit	16 bit	64 bit
0 0 1	TLA ID	NLA ID	SLA ID	Interface ID

ETHERNET ADDR (EUI 64)

L'host quando manda un pacchetto guarda il pref. (i primi 64 bit) se è uguale a quello dest. c'è un link e lo manda direttamente, altrimenti lo inoltra al default gateway

TOP LEVEL authority, che non è detto sia un'ISP di livello 1 per forza, può anche essere un'azienda con una rete enorme. Lo IANA sceglie questi 13 bit.

Questo formato supporta solo una gerarchia a 2 livelli. Il SLA sarebbe la rete privata di un cliente. Per più liv. maggiori il TLA al NLA dà meno di 32 bit e quelli che avanzano NLA li può usare per i suoi clienti.

L'operatore deve usare i 32 bit di NLA ID per dare prefissi ai suoi clienti che si chiamano Next level authority. Se TISCALI acquista un servizio di connettività IPV6 da Telecom Italia, e il pref. di quest'ultima è 2003: essa deve dare a TISCALI un prefisso di 4 byte e il resto è a scelta di TISCALI che userà i restanti 16 bit per identificare i suoi link.

Gli SLA ID vengono gestiti da un'autorità che opera a livello di subnet e li usa per identificare le sue sottoreti (TISCALI in quest'esempio). Per l'interface ID ci sono diverse opzioni: può sceglierlo l'amministr. di rete oppure se lo può creare la stazione IPV6 automaticamente.

L'interface Id è di 64 bit per metterci il MAC e assicurarci unicità. Non tutti sono MAC (che cmq è di 48 non 64 bit) → ciò è EUI64 perché stanno sconvolgendo anche essi e cmq c'è un meccanismo di DAD che ogni stazione deve fare prima di essere attiva.

Mi rendo tracciabile nel mondo inserendo il mio MAC nell'IP, ma non è obbligatorio. Si può usare il MAC come seme per generare numeri random.

Duplicate Address Detection

ed esso viene scoperto tramite ARP

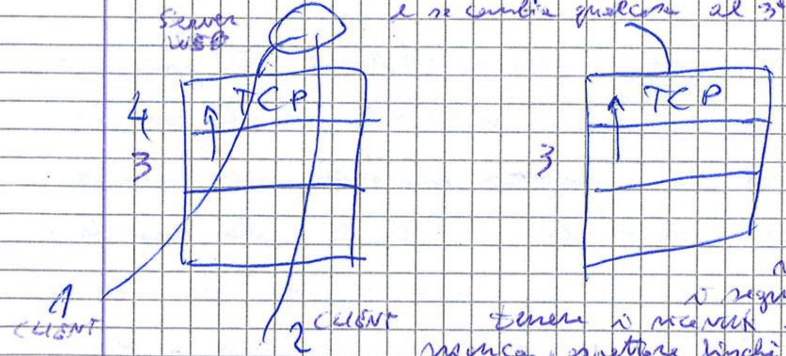
Il protocollo ARP (Address Resolution Protocol) prende l'IP e dà fuori il MAC. Serve a comunicare in locale con una trama di livello 2 (ci vuole l'ind. ethernet dest. se la rete è ethernet). In IPV6 è stato integrato in ICMP perché non è fatto tanto bene. Il meccanismo aggiunto a ICMP si chiama Neighbour discovery.

Anche IGMP (Internet Group Management Protocol), che serve a fare comunicazioni in multicast, è stato integrato in ICMP. Se una staz. vuole ricevere pacchetti, inviati in multicast, di un certo gruppo di interesse, deve dirlo con IGMP. Ma come detto è integrato in ICMP.

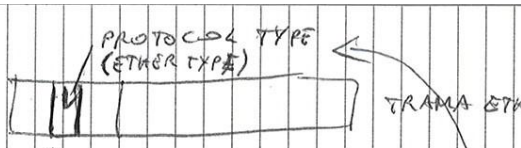
Altri protocolli sono stati solo modificati. Per DNS c'era da cambiare il tipo di record nel DB distribuito, perché l'IP è più grande. I record (in IPV4) che tengono la corrispondenza tra nome e IP si chiamano A, e siccome in IPV6 l'address è il quadruplo, li chiamano AAAA. Il softw. IP che si deve cambiare, sta nel S.O. ma nel frattempo sono stati tutti aggiornati.

RIP e OSPF (OSPF di per sé è gerarchico)
C'era anche da aggiornare i protocolli di routing. I router in IPV4 identificano una stazione con un prefisso che è un indirizzo e una netmask.
Quindi i router ragionano in termini di subnet. In IPV6 è sempre un prefisso, ma si scrive in modo diverso, perché si scrive come una seq. di bit seguita da un byte che dice quanto è lungo il prefisso di quella seq. di bit.
Ci sarebbe anche da fare qualcosa sui router, perché se fanno qualcosa in HW non basta aggiornare il SW.

Bisogna anche modificare i protocolli TCP e UDP, cosa che in teoria è strana perché sono di livello 4 e i liv. sono indipendenti e se cambia qualcosa al 3°, il 4° non si deve toccare.



Il TCP deve assicurarsi che se il trasm. manda una seq. di byte, quella seq. arrive a dest. all'utilizzatore di TCP, esattamente identica, quindi con tutti i byte e nell'ordine giusto. La seq. viene spezzettata dentro dei segmenti TCP che vengono messi dentro dei pacchetti IP e vengono mandati nella rete. Quando arrivano i segmenti, se ne manca qualcuno, TCP si deve tenere i ricevuti e chiedere al TX di rimandargli ciò che manca, aspettare finché ha tutto, ricomporre e mandare al liv 5.
Il prob. è che ci sono diverse connessioni TCP alla volta. Per distinguere i pacchetti si guarda l'ind. del mittente (anche le porte, sebbene per 2 client diversi potrebbero essere identiche) che è stato cambiato in IPV6, e quindi c'è da mettere mano al codice di TCP e UDP.



Il campo della versione non serve perché la si può dedurre dall'intestazione di livello 2, cioè il pacch. IP andrà a finire per esempio in una trama Ethernet, e il campo dice che dentro c'è un pacchetto IPV4 (c'è scritto IP e la versione la si ricaverà dal campo dell'header) (in IPV4)

In IPV6 invece in quel campo c'è scritto IPV6 quindi il campo VER non serve più. Hanno lasciato perché magari tornerà utile nelle succ. vers. di IP.

Il campo Priorità è il TOS (Type of Service) di IPV4. Serve a gestire nei router i pacchetti in modo diverso in base alla politica. I pacch. di Tipo 1 li faccio passare in fretta, quelli di Tipo 2 cerco di scartarne il meno possibile etc etc...

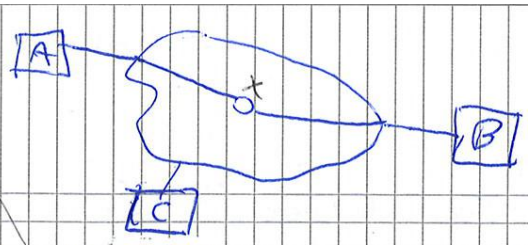
Ho payload length (lunghezza del contenuto del pacch. IP => Total Lenght in IPV4)

L'Hop Limit è il TTL di IPV4. È decrementato ad ogni hop e se arriva a zero il pacchetto viene scartato.

Il Flow label non c'era in IPV4, dove una router capiva se dei pacchetti sono impacchettati (se sono di una telefonata o di un trasferimento file) in base alla porta, il protocollo di liv. 4, gli indirizzi (no mult. che dest. come per le porte). Tutte queste info deve trovarle nel pacch. e magari tenerne traccia. Dove fare il parsing dell'intestazione che non sarebbe di suo mestiere. Un router potrebbe voler sapere se dei pacch. sono impacchettati, per ragioni di QoS.

Con le Flow label, il S.O. sceglie un valore per la telefonata, uno per il file transfer ecc... I router così devono fare meno calcolo per stimare cosa inviare, e nello specifico, per capire che dei pacchetti fanno parte dello stesso flusso, gli basta guardare il campo Flow Label e l'indirizzo mittente. Probabilmente non si usa comunque, perché si è rinunciato a fare la QoS. Tale cosa si fa grossolanamente con la differenziazione del traffico (soluzioni basate su tale concetto). Ad ogni modo FL non serve solo alla QoS ma in tutti quei casi in cui c'è bisogno di capire che dei pacchetti fanno parte dello stesso flusso.

In IPV4 c'era il campo Protocol (Protocollo di liv. superiore) che serviva a dire cosa è contenuto nel pacchetto IP. In IPV6 non ci piace dire "Prot. di liv. sup." perché dopo l'intestazione base potremmo avere di aggiuntive => lo chiamiamo Next Header che può essere un messaggio di liv. 4 (TCP, UDP), oppure può essere un'altra intestazione IP (Extension Headers aggiunti se servono)



L'authentication header

Il primo header non guardato dal router. Serve per ragioni di sicurezza. come fa B a sapere che un pacch. le ha mandato A? Guarda l'ind mittente ma C può mandargli un pacch. spacciandosi per A agendo su quel campo e quindi B (almeno a livello IP) non può verificare nulla. Oppure A può mandare un pacch. e X può cambiare il contenuto e B non può saperlo. Con quest'header si è messa una pezza a questi problemi. Poi hanno estero la soluz. a IPV4 con il protocollo IPSEC (non prende aggiungere un header) che ha un AH che si mette dentro il pacch. IP per verificare di ricevere eventuali manipolazioni.

Encrypted Security Payload

Alle volte non voglio non solo che qualcuno possa modificare i dati, ma nemmeno guardarli. Se cifro il contenuto, devo dare al ricevente modo di decifrarlo. La chiave non può essere ovviamente messa nel pacch, ma TX e RX si possono mettere d'accordo su che chiavi usare prima. Poi nell'ESP si indica di usare la chiave numero x che si è decisa prima di usare. Con un altro pacch. uso la chiave y e l'algoritmo z. Chi guarda il pacch. in mezzo alla rete, non riesce a capire cosa contiene. Questo header supporta un certo num. di meccanismi di cifratura tra i quali RSA. Nelle VPN c'è bisogno di cifrare i dati e di una soluz. in IPV4 sono basate su IPSEC (basato su AH e ESP: sono cioè gli header di questo protocollo).

Destination Options

Analoga all'Hop by Hop Option. Il liv. IP sorg. può mettere delle info opzionali anche non standard, basta che al liv. IP dest. le capisca, che poi questo può usare.

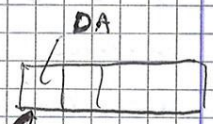
Extension Header Format

Nell'header base scrivono quale è il primo Extension Header. Per sapere il successivo, c'è il primo byte dell'EH che serve a questo. Il secondo byte non si usa sempre perché alcuni EH hanno lunghezza fissa, ma se è usato indica la lunghezza dell'EH. Poi ci sono i veri dati.

Incapsulamento

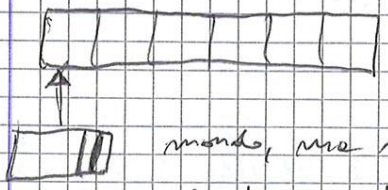
Header IP che contiene come Next un numero che sta per TCP (il pacch. IP contiene un segmento TCP). Se nel pacch IP il Next indica un Routing Header, il successivo header sarà un RH ecc ecc...

Gli Header con opzioni (HBHO e DO) hanno un formato molto flessibile. Ogni blocchetto al suo interno ha Tipo, Lunghezza, Valore (TLV). Se arriva un Header con a un router che non capisce tutte le opzioni guarda la prima opzione, capisce e salta di tanti byte quanto la lunghezza dell'opzione 2 ecc ecc. Il router elabora le opzioni che riesce, e si possono mischiare info standard e non standard.



Multicast address (IPV6 non è lo stesso in IPV4)

Dato un address multicast (che inizia con 8 bit a 1) a cui mandare un pacchetto, nella trama di livello 2 (nel DA) vi è un multicast di tipo ethernet. Gli ind. ethernet multicast sono caratterizzati dal fatto che uno dei primi bit dell'indirizzo è uguale a **1**. Gli ind. ethernet sono fatti di 6 byte (MAC Address a 48 bit) nei primi byte, i 2 bit meno significativi hanno un significato particolare:



(0) (1)

uno mi dice se l'indirizzo è locale o globale (gli ind. che le schede hanno sono globali, unici in tutto il mondo, ma si possono anche usare ind. non unici). Sono i bit meno significativi perché in ETHERNET sono i primi a venire trasmessi in ogni byte. Ecco perché.

L'altro bit serve a dire se l'indirizzo è unicast o multicast (si usa per comunicare con un gruppo di schede ethernet). In una rete ethernet tutte le schede vedono tutti i pacchetti, ma ricevono solo quelli col loro indirizzo o col broadcast. Gli altri non vengono sollevati al S (la scheda non genera un interrupt alla CPU per dirle che c'è un pacch. da elaborare)
Una scheda spesso i pacch. multicast se non istruita diversamente attraverso i driver dal S.O. Quindi il multicast in ethernet funziona in modo gestito dal ricevitore. Se una scheda è impostata per ricevere pacch. inviati all'indirizzo X, allora li riceverà. Quindi il gruppo che riceve all'ind X si forma quando le schede vengono configurate a far parte del gruppo X.
Quindi nella trama ethernet con DA ci mette X e le schede istruite a ricevere quel gruppo X ricevono il pacchetto altrimenti no. Ora se io voglio inviare un pacchetto IP a un ind. IP multicast, devo mettere una trama MAC che va a un indirizzo MAC multicast. Ma dato un ind IP multicast, che ind. MAC multicast ci va?
Si fa una corrispondenza algoritmica tra un ind. IP multicast e un ind. MAC multicast. Vediamo come si fa in IPV6. In IPV4 è diverso perché gli ind. hanno un formato diverso.

Si prendono gli ultimi 32 bit dell'IP address e ci si mette davanti 3333 (in HEX) per arrivare ai 48 bit di un MAC address. Si vuole che 33 dica che il primo byte ha gli ultimi 2 bit a 1. Quindi se l'IP address a cui devo inviare pacch. è FF::1, il MAC multicast è 333300000001.

Di default non viene ricevuto da nessuno.

ICMP V6

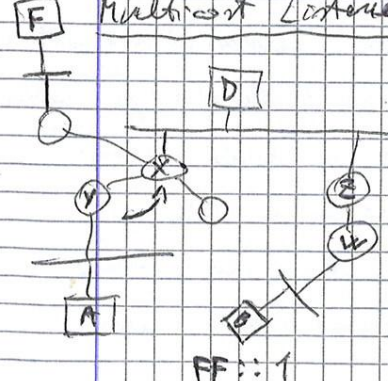
I pacch. (messaggi) ICMP vengono incapsulati dentro i pacch IP come in IPV4.
Si usa come in IPV4 per diagnostica (PING, echo request, echo reply per implementare PING) che comprende molteplici ha parte della rete come quando un router decide di scartare un pacch. e avvisa il mittente proprio con ICMP

(4) Parameter Problem → Il router ha scartato un pacchetto perché ha trovato nell'header un valore che non si aspettava oppure un campo che non riconosce. In realtà i campi opzionali sono codificati in modo tale che anche se un router non li riconosce può andare avanti lo stesso.

(3) Time exceeded → Quando il TTL del pacch. va a zero o nel caso di IPV6 l'hop limit, il router non può inoltrare il pacch., lo scarta e avvisa il mittente con l'address del pacchetto.

(2) Packet too big → Normalmente i router frammentano i pacch., tranne se sono (in IPV4) identificati come Don't Fragment, e in questo caso può servire questa notifica. In IPV6 il bit ↗ non c'è ma se un router non frammenta e riceve un pacch. troppo grande notifica così.

Il tipo di messaggio specifico viene specificato nel campo type. I messaggi 135 (Neighbor Solicitation) e 136 (Neighbor Advertisement) servono per il Neighbor Discovery. Il 135 è la richiesta e il 136 la risposta. Il campo code serve perché alcuni messaggi hanno dei sottotipi. La checksum protegge da errori il messaggio ICMP, e il body dipende dal tipo del messaggio. Per un (2) potrebbe esserci l'header del pacchetto che ha generato il problema più una parte del payload. Non è obbligatorio rispondere a un pacch. ICMP.
sia in TX che in RX.
L'ICMP è tutto opzionale tranne il Neighbor discovery. Ci sono anche nuovi messaggi come 133 → Router Solicitation e 134 → Router advertisement, che servono a una stat. per scoprire l'esistenza di router vicini (serve nell'autoconfigurazione). In IPV4 una stat. per funzionare ha bisogno di 3 info: Indirizzo IP, Netmask, e default Gateway. Non può mandare pacch. fuori dalla LAN senza quest'ultimo e in IPV6 non c'è modo di configurarlo automaticamente se non col DHCP. In IPV6 la stat. possono scoprire i router vicini e usarli come D.G.



Multicast Listener Discovery (messaggi 130, 131, 132)

In IPV4 per il multicast si usa un protocollo a parte che si chiama IGMP. In IPV6 è integrato in ICMP.

Vediamo subito come funziona il multicast a livello di link tramite trame di liv. 2 multicast, poi esso deve funzionare anche su Internet. Come fanno A e B a ricevere un multicast da D? A e B sono interessati a ricevere pacchetti multicast p.e. da FF::1 e configurano le loro schede di rete a ricevere trame di liv. 2 multicast da 3333::1. I pacch. IP inviati da D sono messi quindi dentro trame con tale address, ma queste trame non arrivano al link di D, per cui non

⊗ Messaggio ICMP che comunica interamente verso il gruppo. In questo caso vanno persone a molte perché sono messaggi per i router che così possono fare arrivare sulla rete i pacch. multicast. Ma i pacch multicast per il SNMA non devono arrivare da nessuna parte perché tale ind. si usa solo localmente. Funziona così perché viene trattato come un qualunque ind. multicast. Il LLA può essere usato per fare un'operazione di Router Discovery (messaggi ICMP di Router Solicitation e Router Advertisement)

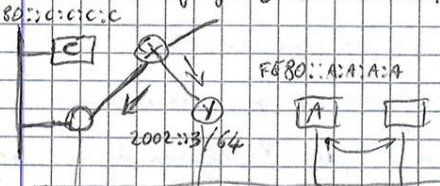
messaggi che viaggiano su un singolo link. Servono solo a X (Y) per scoprire quali sono i gruppi di almeno una stat. interessata. L'info viene propagata con i prot. di routing multicast. La query può essere mirata ad un gruppo (FF::1) specifico, o qualsiasi (campo Multicast Address vuoto).

Anche a livello di switch esistono protocolli per far loro capire dove ci sono stazioni iscritte a un gruppo e dove no, ed evitare così che facciano broadcast, [illegible] quando si riferiscono al broadcast.

16/10/14

Se le stat. non hanno ancora info si creano un ind. link local usando il pref. LL che è fisso FE80::/64. Appena viene accesa, la stat. si crea tale indirizzo. Gli indirizzi IPV6 sono fatti da 64 bit di pref. e 64 di ID Interfaccia quindi i prefissi privati sono tanti (FD/8 → devo arrivare a 64 bit). Il pref. LL è 1 solo, quindi la stat. può crearsi da se l'indirizzo. Nell ID Interfaccia ci deve mettere qualcosa di univoco nel link. Si potrebbe usare il MAC, ma questo alla fine si può cambiare. Qualunque sia il modo per ricavarselo, con la procedura di Duplicate Address Detection la stat. può accertarsi che nessuno nel link sta usando quell'ID. Ottenuto il LL address, la stat. può operare nel link, quindi scambiare pacch. con un'altra stat. del link. Dopo aver scelto l'ind. (p.e. FE80::A:A:A:A) per far sì che il Neighbor Disc. funzioni si deve registrare al gruppo multicast Solicited Node Multicast address. Significa mandare alla scheda di rete come multicast e generare un Multicast Listener Report. ⊗ Per ritrovare il modulo nodo. Fuori dal link serve un prefisso unico nella rete quindi serve procurarsi un pref. site local o globale privato. La stat. non può usare il pref. LL perché è sempre lo stesso e farebbe pensare ad A che C sia nello stesso link (cosa non vera in questo caso) facendo partire un Neighbor Solicitation a cui nessuno risponderà. La stat. non può inventarsi un pref. privato quindi genera un mess. di Router Solicitation (ICMP) che viene mandato a un ind. particolare che è quello di tutti i router (multicast a cui si sono iscritti tutti i router). Se ci sono router sulla rete locale, risponderanno comunicando il prefisso con cui sono stati configg. per quella rete locale. Anche via IPV4 per ogni rete fisica c'è un prefisso unico per quella LAN e non si possono avere 2 stat. sulle due LAN con lo stesso prefisso. I pacch. non uscirebbero dalla LAN e i pacch. esterni i router vedrebbero 2 strade e non saprebbero dove mandarli. Sceglierebbero una strada e sull'altra i pacch. non arriverebbero mai.

Nello scegliere gli ind. l'amm. di rete deve ricordarsi che link diversi abbiano pref. diversi. A, attivato il LLA, manda un RS e Y risponde col suo indirizzo, nel quale A vede il prefisso 2002::3/64. Usando anche lui quel prefisso, si crea un indirizzo allo stesso modo generando un Interfaccia ID. Potrebbe essere 2002::3:A:A:A:A ma anche 2002::3:A:B:C:D. Questo si chiama autoconfigurazione stateless e non si usa il servizio DHCP. Ad ogni modo ha stat adesso conosce un router, quindi se ha un pacch. con un prefisso diverso dal suo, lo manda ad esso, proprio come in IPV4, solo che lì il default gateway bisogna impostarlo a mano o col DHCP.



... privata e l'utente medio non può comprendere il DHCP all'interno del router etc... con quanto detto prima al router, oltre al suo ind., viene anche dato il pref. da usare nella rete locale, e lo distribuisce alle stat col DHCP (o a eventuali router di BR)

Riepilogando: ind. LL, verif. unicità, Router Advert., config. di più indir. usando prefissi annunciati dai router, e che possono essere globali, privati, site local.

Ogni stat ha più indirizzi sull'interfaccia e li usa in base al caso. Vale anche in IPV4 anche se non nasce per avere più indirizzi. Nella maggior parte dei S.O. è possibile configurare più di uno e specificare da quale inviare i pacchetti. Con la stateless si risolve il problema del Dentist Office. Il DHCP V6 è analogo a quello V4 con qualche funzionalità in più (i messaggi hanno nomi diversi) La stat. chiede la config., più server DHCP nella rete fanno la loro offerta e la stat. richiede una delle config. proposte e poi c'è il reply (acknolagment di IPV4). La config. ha una durata limitata e la stat. deve rinfrescarla, o la può rilasciare etc... Nella config. con DHCP, la stat. riceve il proprio ind., non un prefisso. Nella stateless il pref. è poi fissato dall'interfaccia ID.

C'è una procedura per capire se il LL scelto è unico. Si chiama come detto Duplicate Address Detection

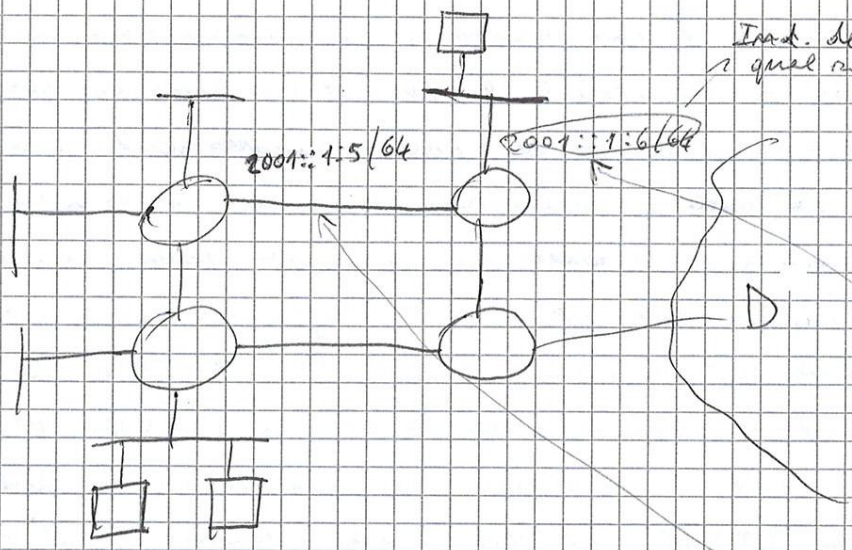
Si fa una Neighbor Solicitation relativa all'ind. che la stazione ha scelto. Mettiamo che inizia con l'ind LL. FE80::A:A:A:A (non vale anche per 2002:3:A:B:C:D) ma lo fa con entrambi. Si crea il SNMA, manda un messaggio ICMP Neigh. Sol. a tale indir.. Se c'è un'altra stat. con quell'indirizzo, darà config. e interviene e risponderà, e se risponde, la stat. capisce che non è unico e si genera un Interface ID diverso. Ovviamente la stat. farà più richieste perché una potrebbe andare persa e poi se non c'è risposta assume che non la usa nessuno.

Per fare in modo che l'Interface ID è probabile sia univoco ci sono diversi modi una delle opzioni è il MAC (48 bit ma c'è già estensione a 64) => prendi 3 byte del MAC, poi una certa configurazione FF:FE e poi gli ultimi 3 byte del MAC. Siccome i MAC sono univoci, facendo così è improbabile ci siano due Interface ID uguali. Tutte le volte che poi la stat dovrà rigenerarsi un Int ID, uscirà sempre lo stesso. Questo vale se mi collego a casa, al pub o all'aeroporto, quindi sono tracciabile e si viola la privacy dell'utente. In IPV4 c'è il NAT, ma anche gli ind. pubblici cambiano quando DHC.

Per la privacy si è trovato un modo per far sì che l'interface ID sia sempre diverso. Si prendono i 64 bit generati a partire dal MAC, gli si affiancano 64 bit random. Dai 128 bit con MD5 si generano altri 128 bit (il risultato non vale quindi non posso risalire ai 128 che l'hanno generati. I 64 bit più significativi se superano il processo di Duplicate Address Detection vengono usati come Int ID. La prossima volta che viene un Int ID esso viene generato sempre con una funzione dall'ultimo Int ID usato

17/10/14

Esercizio sull'indirizzamento IPV6



Sono l'amministratore di questa rete e voglio collegarla alla rete IP. Mi viene dato il prefisso 2001::1/48

Dobbiamo dare degli indirizzi alle interfacce delle stazioni in modo che possano poi funzionare sulla rete IP. Ogni router avrà almeno un indirizzo per interfaccia e saranno tutti diversi. Se metto quel valore, su quel link tutte le interfacce avranno quel prefisso. L'indirizzo completo dell'interfaccia di quel router sarà per esempio 2001::1:6:1:1:1:1/128. La stazione sul link sa che deve usare quel prefisso grazie ai messaggi ICMP dei Router Solicitation e Router Advertisement, oppure DHCP oppure la configuriamo a mano. Qui possiamo mettere un altro prefisso al 5 al posto del 6 che possiamo usare per tutte le interfacce di quel link, le 2 dei router in questo caso. Noi non sappiamo quante interfacce ci sono su un link ma non ci interessiamo a differenza di IPV4 dove si cercava di fare il prefisso più lungo possibile che ci desse abbastanza host ID. Qui abbiamo prefissi di 64 bit e l'Interface ID è anche di 64 bit, quindi abbiamo un numero di Interface ID enorme. Chiaramente su un link punto-punto dove ci mi servono solo 2 sprechiamo un sacco di indirizzi ma sono talmente tanti che non ci interessiamo. Possiamo anche aggregare perché se abbiamo fino 48 bit di prefisso, ne restano 16 (= 65000 reti)

(Se serve riscaldare anche per inverno)

o farne di nuovi. Ci sono alcuni protocolli che supportano il routing integrato, ma l'approccio comune è quello di Ships in the night.

Ships in the night

Due mondi che mangiano da molte ognuno che va per la sua strada ignorando l'altro. Se in una rete, D può essere sia una dest. IPV4 che IPV6, i router usano 2 protocolli diversi, 1 per capire come raggiungere D(IPV4) e uno per D(IPV6). Il modo in cui decidono il percorso per mandare pacch. IPV4 e una stazione può essere completamente diverso da quello con cui scelgono i percorsi per mandare pacchetti IPV6, quindi i router fanno lavoro doppio. Con questo approccio si può continuare a usare gli stessi protocolli IPV4 senza modificarli. Fare l'approccio integrato vuol dire passare a un nuovo protocollo, che potrebbe essere bacato, non funzionare bene etc. Per questo è preferibile l'approccio Ships in the night. Nella tabella della slide 21 si nota che la maggior parte dei protocolli sono usati in questo approccio. Static significa scrivere le tabelle a mano. Il RIP è un protocollo IPV4 e la versione IPV6 si chiama RIPng (Next Generation) La versione IPV6 non si può usare in IPV4 (stesso discorso per EIGRP for IPV6) L'IS-IS è molto vecchio e pensato per le reti OSI, non IP, ma è fatto bene ed è stata fatta una versione IPV6 compatibile con IPV4. Già la versione IPV4 faceva una sorta di routing integrato per destinazioni OSI (dest. identificate con un prefisso OSI) e IPV4. La pila OSI ha una sua implementazione ed evoluzione dei protocolli fatti seguendo tale architettura, uno di questi è IS-IS. Esso è stato esteso per portare info anche su dest. IPV4 e con un'altra estensione anche su IPV6. Ragionamento simile per MP-BGP4.

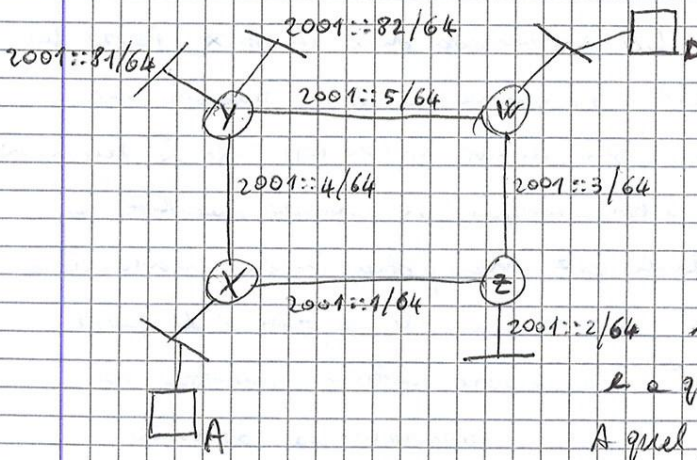
Showing IPV6 routing table

Ci sono una serie di destinazioni, dei prefissi, seguiti dalla lunghezza del prefisso. Il numero tra [] è una metrica, una misura di quanto lontano è la destinazione (ossia che può essere configurata o imparata con qualche protocollo di routing). Dopo "via" c'è l'ind. del NEXT HOP, cioè di un router direttamente collegato a quello su esame, a cui mandare il pacchetto se il prefisso dest. è quello sopra. Quindi tale indirizzo è

non fanno parte di per se delle tabelle di routing. Ci sono in quelle dei router CISCO e specificano dove il router ha imparato quella particolare informazione. Per esempio la prima riga ha una "O" che significa che l'info è stata ottenuta con OSPF. Tornando alla figura di prima, l'amministratore potrebbe dire a mano (statica) ad X che qualsiasi destinazione la può raggiungere tramite Y e potrebbe dire a X di usare un protocollo di routing (per esempio OSPF per dire agli altri router in generale quali destinazioni si raggiungere, e in particolare che conosce la default route. Così quando non sanno dove mandare, gli adiacenti mandano a X, mentre W nella default route avrà Z che a sua volta avrà X appunto. Gli stessi concetti valgono per IPV6.

Esercizio

Dato la rete sotto e il prefisso 2001::/48 scegliere dei prefissi per i link e degli indirizzi per i router e scrivere la tabella di routing dei router in modo che ogni router sappia raggiungere qualsiasi destinazione che c'è lì. Quindi se A deve mandare un pacchetto a D, si rende conto che D non è sullo stesso link e consegna il pacch. al router X che conosce grazie ad un router advertisement, ha scoperto il suo indirizzo IP, e da esso ricava il MAC address con il Neighbour Discovery e a quel punto gli può consegnare il pacchetto per D.



A quel punto X per raggiungere D deve avere nella sua tabella di routing il prefisso di D. Lì ci deve essere scritto se passare il pacchetto a Y o a Z.

Se aggiungo sul link X-Z il prefisso 2001::1/64 devo mettere nella tabella di X che per raggiungere 2001::1/64 bisogna uscire dall'interf. che collega X a Z? NO, perché se dico a X che lì il suo indirizzo è 2001::1:A:A:A:A so già ho detto implicitamente che tutto ciò che inizia per 2001::1/64 si raggiunge tramite quell'interfaccia. A X dovremmo dire invece come raggiungere

la tabella per conto loro, non si risparmia solo nella tabella che è più piccola e quindi i router devono lavorare meno, usare meno memoria, ma anche nelle linee che si condividono. Invece che dirsi l'un l'altro che sono raggiungere 150 dest. diverse, se ne dicono 1 con un prefisso di 60 o magari anche 58, dato che sono 150 destinazioni.

03/11/14 Voice over IP significa trasferimento della voce tramite il protocollo IP, anche se può lo si è usato per trasmettere qualsiasi tipo di dato multimediale.

I telefoni funzionano a commutazione di circuito, quindi prima di inviare la voce devo instaurare un circuito nella rete quindi devo allocare delle risorse sufficienti a trasportare la voce

Alloco dei circuiti full-duplex (andata e ritorno) di 64 Kbps, valore che deriva dalla modalità di campionamento della voce stabilita in base al teorema di Nyquist. Campiono a 8 KHz, per uno 8 bit (in base a che dice col teorema) per ogni camp vocale che esce dal segnale, e quindi 8 KHz, 8 bit/camp. => flusso di 64 Kbps. Dato che è FD devo allocare 64 Kbps in una direzione e 64 Kbps nell'altra.
In questa modalità non c'è compressione, è un segnale digitale che contiene tutta l'informazione del segnale analogico. Se c'è una pausa e il segnale è piatto non posso interrompere il campionamento, devo cmq creare i campioni da 8 bit con una freq. da 8 KHz. Non posso inoltre garantire caratteristiche di altra qualità, ho sempre 64 Kbps. Ciò che posso fare è allocare alla stessa telefonata più circuiti da 64 Kbps, modificando in questo modo la qualità, ma in modo molto arbitrario, perché ogni circuito deve essere da 64 Kbps.
Non posso avere moltiplicazione statistica: quando 2 utenti stanno usando una rete a pacchetto possono usarla insieme, mentre uno sta zitto l'altro può usare tranquillamente le risorse disponibili. Tutti e due trasmettono alla velocità massima finché non vanno a parlare insieme. Se devono mandare insieme dei pacchetti uno dei 2 sarà buffferizzato e trasmesso leggermente dopo ma in generale entrambi hanno un grado di condividere le stesse risorse (definizione non formale). Nella rete a circuito non posso ottenerlo perché ogni volta che alloco un circuito esso è staticamente allocato alla persona che l'ha creato quindi alla chiamata entrata. Se uno sta zitto, le altre sorgenti non possono usare quelle risorse perché sono allocate staticamente all'altra sorgente. In pratica nella commutazione di circuito c'è un uso inefficiente delle risorse.
Nella telefonia tradizionale ho bisogno di una procedura di segnalazione, il CALL SETUP. Consiste nell'invio di opportuni messaggi per fare in modo che il circuito venga creato. Il segnale di libero fa parte della segnalazione per la rete a circuito.

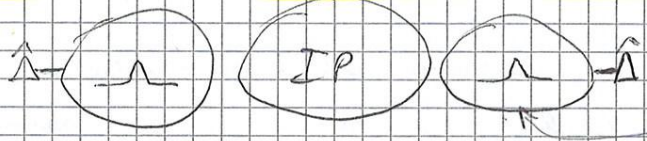
centrale, ci sarebbe di nuovo lo stesso problema. Non posso garantire in modo formale che quel traffico possa avere qualità. Un altro approccio è riservare delle risorse nella rete quando vado su ogni nodo in modo simile alla commutaz. di circ. e riservare una quota delle risorse (per esempio una quota della banda di uscita). Così i pacchetti multimediali subiranno un ritardo che posso minimizzare in qualche modo e ci sono formule per scrivere formalmente l'Upper Bound del ritardo. Il problema è la scalabilità perché se per ogni flusso devemi andare sui router a riservare banda, spazio nei buffer, il nodo passerebbe il suo tempo a fare questo, i suoi cicli di CPU a gestire il protocollo e non avrebbe più capacità computazionali per smistare il traffico che è il suo vero obiettivo.

Nessuna delle 2 soluzioni è ottimale, la prima perché non permette formalmente che il flusso audio/video possa avere qualità, la seconda perché non è applicabile su reti di larga scala.
Le cose funzionano perché la rete Internet attuale è abbondante rispetto a quanto dati ci passano ma non sono un problema significativo.
Questo non significa che non lo saranno in futuro. Se i link cominciassero ad essere congestionati, Skype inizierebbe ad avere problemi perché non ci sono soluzioni formali per gestire la qualità del servizio.

Nella rete ci sono diversi attori con differenti obiettivi e ognuno vede la stessa tecnologia da un diverso punto di vista. L'obiettivo fondamentale dell'utente domestico è risparmiare. La telefonia tradizionale mi piace perché vedo subito alcune risorse, cioè il circuito e ciò non può essere gratis. Nella telefonia su IP vedo ad aggiungere un Σ (la telefonata) a tutto il traffico che sta transitando con la multiplazione statistica. Per l'operatore i costi sono differenti perché non c'è differenza tra telefonate e traffico WEB. Anche la GUI e i piccoli servizi di Skype possono allettare l'utente ma fondamentalmente il fatto che telefono gratis è l'obiettivo principale

Parliamo di Skype perché è conosciuto da tutti, ma poi ci concentreremo su protocolli standard come SIP. Se devo progettare una rete VoIP la faccio con SIP, non con Skype, o comq la faccio con protocolli standard.

Ora l'obiettivo è quello di mettere le telefonate nei pacchetti anziché nei circuiti e usare la sola infrastruttura a pacchetto. Si fa esattamente come fanno se voglio fare una telefonata VOIP da due dispositivi END-TO-END. I protocolli VOIP poco cambia se girano tra 2 telefoni o tra 2 END-TO-END ci vuole sempre un modo per prendere campioni vocali e metterli nei pacchetti.
Si parla di TOIP (Telephone over IP) ma il concetto è il medesimo La distinzione è formale tra ciò che si fa nel VOIP classico in cui prendo la voce e la impacchetto mandandola nella rete a pacchetto. Nel caso di TOIP si cerca solo di usare meglio il Backbone. La telefonata sull'edge sarà sempre a circuito, si cerca di portare in un backbone a pacchetto la mia telefonata a circuito.
L'utente usa il classico telefono a circuito, quindi dalla casa fino alla centrale dell'operatore si usa il circuito, non l'ADSL. Poi i campioni voce della telefonata sono imbustati in pacchetti e inviati nella rete a pacchetto. Ecco perché il nome TOIP



Nel TOIP c'è la complicazione aggiuntiva di dover trasferire anche i messaggi di segnalazione, non basta far squillare il telefono ma serve trasferire anche la segnalazione in network che va a creare i circuiti perché qui ci possono magari dei commutatori a circuito che vanno settati.

Vantaggi / svantaggi

Col TOIP posso continuare a usare il vecchio telefono. Conviene all'operatore perché deve mantenere una sola infrastruttura, e al cliente affezionato alla vecchia telefonia. Una mancanza di update è mantenuta una sola nel backbone, perché ci vogliono dei gateway che trasformano la telefonata a circuito in una a pacchetto. Sono i dispositivi che prendono i campioni e li mettono nei pacchetti. Il TOIP a differenza del VOIP non va a migliorare i servizi per l'utente. Il provider potrebbe pensare di avere sia il TOIP che il VOIP in modo tale che l'utente che vuole servizi innovativi possa usare il VOIP END-TO-END, cioè direttamente partendo dal proprio dispositivo, ora penalizzo fatto solo in parte. I provider tendono a fare offerte ADSL + telefonate in VOIP, ma non spingono verso dei servizi innovativi come l'e-presence.

Per un'azienda in teoria l'obiettivo non deve essere risparmiare ma avere servizi innovativi per migliorare il processo produttivo e quindi aumentare gli utili, le vendite e l'efficienza aziendale

Il VOIP può essere utile perché può fornire il servizio di E-PRESENCE, e quindi utente di diverse reti possono sapere quando le altre persone sono disponibili a parlare. Può sembrare una ridondanza ma non lo è

Alle fine anche le aziende sono interessate a risparmiare. Per cui penso lo si può vedere come l'utente domestico che vuole il VoIP per comunicare dal resto del mondo a basso costo

Creating a VOIP flow

Ma come si instradano i pacchetti? Come li tratto in rete? Quando come realizzo il data plane della rete in modo tale che poss supportare bene il VOIP? Abbiamo detto che è vero che non dobbiamo prenotare risorse in rete, ma dobbiamo garantire una regolazione END-TO-END (tono libero, occupato etc...). Questo non ha a che vedere col piano dati ma col piano di controllo. Ora parliamo brevemente del piano dati poi vedremo i capitoli più importanti dal punto di vista del networking legati al piano di controllo. La regolazione è a tutti gli effetti una problematica di rete. Si deve trovare il modo di instradare la chiamata in rete. Come trovo il destinatario? Quale routing uso?

Trattiamo le cose una alla volta: Col sampling scelto la f_c, si campiona il segn. analogico e poi si devono trasformare i campioni reali, che sono dei livelli di segnale in bit. Qui si entra nella seconda fase che è quella di Encoding che è il passaggio da a comp. reali alla seq. di bit. Il codec più famoso è il PCM66 (camp da 8 bit, $f_c = 8$kHz) quindi flusso a 64 KBPS. È un encoding senza perdite e poss migliorare l'efficienza di trasmissione in diversi modi
Posso introdurre un fattore di compressione nell'encoding per ridurre il bitrate senza perdere eccessivamente qualità. Con la codifica per differenze (funziona meglio col video) se ho 2 immagini, invece di codificarle allo

tale tipo di rete ci vuole il PCM 64. Con una codifica aggressiva che riduce i bit per la codifica all'interno della rete VOIP, quando il FAX arriva il gateway non è più in grado di riconoscere il segnale con gli 8 bit perché è stato perso al gateway d'entrata. Per la voce vale è un problema perché non ne perde qualità (al max voce più metallica leggermente), invece per il FAX è un problema. Quindi si usa il PCM 64 per retrocompatibilità con i servizi offerti dalla telefonia tradizionale. L'uso delle risorse è meno efficiente ma per ora banda ce n'è.

06/11/14

Con la soppressione delle pause è nell'encoder il problema di risincronizzarsi, non sul ricevitore (ERRORE lezione prima). Si può risolvere ma richiede potenza elaboratore.

Dobbiamo cancellare l'eco, questo perché ciò che dice A esce e va a finire sul microfono e gli torna indietro. Se arriva con un ritardo massimo di 40 ms il cervello umano non lo percepisce, altrimenti è un problema. 200 ms è un esempio tipico di valore tipo che si può avere su una rete IP. Per eliminarlo ci sono sia soluzioni SW che HW a seconda che il dispositivo sia uno SW Phone o un telefono HW. La soppressione delle pause permette di limitare la banda e se ne può anche fare a meno, questo invece è fortemente e ne tolto.

L'eco c'è anche nella telefonia classica, è un problema di ritardi. Negli USA lo avevano messo sin costo subito e cancellavano l'eco con degli algoritmi e dispositivi. In Europa la cancellazione dell'eco non c'era perché le distanze erano per lo più nei limiti nazionali; poi lo è diventato quando si è cominciato a fare chiamate internazionali, perché in esse il ritardo tende a salire sopra i 40 ms e si è iniziato a sentire l'eco.

È chiaro che dobbiamo considerare questo problema, perché nella rete a pacchetto, oltre al ritardo di propagazione c'è quello dovuto al buffering, alla packetizzazione, alla trasmissione ecc... Tuttavia ci sono fonti di ritardo che nella rete a circuito non c'erano.

Packetizzazione

I campioni codificati devono poter essere trasferiti all'interno di pacchetti IP. Essi saranno equidistanziati perché inviati con una certa fc. Potrei pensare di mettere ogni campione in un pacchetto, e dal punto di vista del ritardo è una soluzione ottimale (ritardo della creazione del pacchetto), e spedirlo subito. Il problema è l'overhead che si genera.

In realtà nelle slide c'è il priority

la tecnica funziona perché il traffico di questo tipo è una percentuale non alta, e inoltre la capacità della rete dinamicamente non è sfruttata al max. Per accodare i pacchetti nella giusta coda si usa il marking (per far capire al router quali sono i pacch. di traffico prioritario). Nel caso di IP si scrive nel campo Type of Service (IPV4). In IPV6 è il Differentiated Shaping Service. Valori diversi a router impostati in un certo modo. Col VOIP il posto migliore per fare il marking (anche perché prima non ho il pacchetto) è sul gateway. Prima sono via un circuito e la qualità è garantita dalle risorse allocate per il circuito. Per VOIP invece abbiamo 2 sotto casistiche. Nel primo caso il servizio VOIP è incluso nell'ADSL. In questi casi c'è una scatoletta tipo Telefono Station a cui collegare il telefono. Il provider può benissimo fare il marking dentro la scatoletta, ma sono scelte dell'operatore perché può decidere di farlo in centrale sul pop, e non è un problema perché non vado a usare link condivisi. Tra l'altro in genere ci vengono forniti 2 indirizzi IP, uno per i dati e uno per il VOIP, quindi anche nella base dell'IP address il provider può fare il marking. L'altro sottocaso è VOIP e basta con un client VOIP che gira sul PC tipo Skype, o un client SIP. La cosa migliore sarebbe lasciare all'utente la responsabilità di marcare il traffico VOIP ma quasi tutti ne approfitterebbero per velocizzare tutto il proprio traffico. L'altra soluzione è che il provider cerca di classificare il traffico sul gateway ma non è semplice. Si potrebbe affidare al numero di porta: 80 → TCP quindi non è VOIP, almeno in teoria. Non è detto che sulla 80 ci debba per forza girare il WEB e sulla X il VOIP, anzi non possono cambiare.

Skype per superare alcuni NAT e firewall fa marchiare da WEB usando la porta 80 per esempio. Un'altra soluzione sarebbe quella di fare il Deep Packet Inspection, ovvero leggo il payload alla ricerca di keyword per capire che traffico trasporta il pacchetto. La keyword GET mi fa capire che è traffico HTTP. È abbastanza onerosa a livello computazionale e non riesce a classificare il traffico criptografato, e Skype è criptografato essendo un protocollo proprietario segreto. Se il traffico è criptografato non vedo il payload e

con jitter nullo. È vero che con la priority queueing si cerca di eliminarlo, ma solo idealmente. I campioni arrivano quindi con delay diversi, e devono essere rimessi alla stessa distanza. Il modulo fa quindi da buffer e rilascia i campioni (ognuno è a pacchetto) ogni 125 μs (PCM 64)

Il modulo detto fa anche una operazione di rendering. Il routing IP è best effort, quindi magari un secondo pacchetto fa un percorso più breve e arriva prima del primo. Se uso UDP, fare il rendering mobile perché UDP non ha numeri di sequenza (TCP sì) e quindi l'applicazione deve essere sviluppata in modo (aggiungendo l'info tra i dati) che essa stessa possa ordinare e ricostruire l'ordine. Il modulo di rendering deve essere parte integrante dell'applicazione e basarsi sulle info che l'applicazione mette nel pacchetto dati.

con RTP ho il numero di sequenza e il re-ordering risulta quindi più semplice. RTP può fornire un supporto migliore al traffico multimediale rispetto al semplice uso di UDP. Dopo il rendering entra in gioco il decodificatore, deve riconoscibile i campioni in analogico in modo tale che lo speaker li faccia sentire all'orecchio. Il Decoder è più semplice dell'encoder perché deve fare solo questo, non deve usare algoritmi per la soppressione della parola, non deve usare algoritmi per calcolare la parte più importante di un'immagine/suono/video per poter usare più bit per la codifica. Prima del decoder può esserci un'altra funzionalità, una tecnica di compressione degli errori, anche se poco usata. L'idea è di inserire in un certo pacchetto una campione N usando una codifica high quality, invece il campione codificato alla qualità che desidero dove e quel flusso. Nel pacchetto successivo, oltre al campione +1 ad alta qualità, metto una copia del precedente a bassa qualità che occupa meno banda. L'idea è che se perdo N high quality dopo ho campione quello low quality, il tutto senza duplicare la banda ma aumentandola di poco. Non è molto usata perché di solito le perdite nelle reti non sono random, ma avvengono a burst. La tecnica funziona bene se il prob. è di avere sui pacch. Per esempio se trasmetto un pacch. su un canale wireless e un ricevitore ho un errore sporadico che la tecnica risolve perché ci vuole accorgo a short che il pacch - è corretto e viene scartato. Ho il pacch. successivo e l'uccello è difficile possa di nuovo. Al giorno d'oggi gli errori non si fanno per il checksum corretto ma per congestione e la congestione la si ha per qualche ms e si perdono diversi pacch. consecutivi.

alla volta. Tra cose di probl. di rete che fanno scattare il timeout e TCP si blocca, Skype non invia i pacch. neanche ma tiene conto del numero di pacch. che in quell'intervallo gli sarebbe dovuto pervenire. Se quelli dopo rimarrebbero bloccati. Quando TCP riparte e si libera spazio nel buffer, i campioni vecchi non riconosciuti vengono scartati e a TCP arrivano quelli attuali. Non è una soluz. ottimale ma su UDP con rete congestionata è molto probabile che andrei a perdere tutti i campioni scartati con l'approccio TCP. Il destinatario sente un attimo di silenzio ma poi la conversazione riparte. Il servizio è sfruttato decisamente bene.

Per quanto riguarda la banda, i flussi di questo tipo (Voice) sono scalabili, quindi non vincono ad adattarsi alla banda. Un flusso da 64 Kbps è il tasso del campionatore + encoder che lo ha generato e deve garantire che la rete sia in grado di trasportarlo tutto. Se ho 40 Kbps non riesco a trasportare il flusso e accumulo ritardo. I flussi di tipo dati sono invece elastici. Se ho 1 Gbps trasferisco a tanto, se ho 1 Mbps a tanto ecc. Varia il tempo di download ma non è un grosso problema.

Perdute

Skype abbiamo visto che butta via dei campioni, eppure se ne potrebbero perdere con UDP per le congestioni. Misure sperimentali hanno stabilito che se perde fino al 5% dei pacchetti l'utente è comunque soddisfatto del servizio perché la fase è comprensibile. Ciò non è vero nel traffico dati con cui per ricevere un file devo ricevere tutti i pacchetti e infatti usiamo TCP.

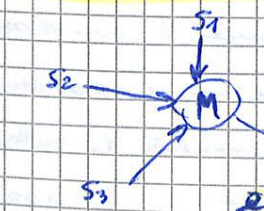
07/11/11

RTP è stato definito per aiutare UDP nel trasporto dei campioni multimediali. Con UDP posso fare il multiplexing/demultiplexing grazie al numero di porta. Quando la macchina ric, grazie al DM di UDP a quale porta consegnare il pacchetto arrivato a destinazione. Così arriva all'applicativo giusto. Per fare il rendering UDP solo non è sufficiente perché non ho il num di sequenza. Potrei far sì che l'applicazione metta tale numero dentro il messaggio. La soluzione migliore è un protocollo tra UDP e l'applicazione, RTP appunto, così l'applicazione si occupa solo di encoding/decoding che sono cose proprie del livello applicativo. RTP fornisce una trasmissione multicast in modo nativo. Il può offrire una trasmissione

comunicazione non va come speravo e può chiudere su un codec più aggressivo cambiando il bitrate. RTP è difficile da classificare nei firewall ma anche per scopi di QoS e quest'ultima cosa è legata al marking, infatti abbiamo detto che è difficile marcare perché non si è certi che ad una porta corrisponda il servizio atteso. Qui è ancora peggio perché RTP non usa porte standard (le well known ports). Quanto perché per ogni flusso ho bisogno di una porta e non posso sapere quante una me ne servono a priori. Per il firewall potrei voler permettere la comunicazione multimediale e come lo configuro il firewall? In genere si lasciano aperte le porte legate ai servizi che voglio offrire, e per il WEB è semplice perché uno guarda sempre la porta 80 e la apre, stessa cosa per la porta con la 25 etc. Ma RTP non usa well known ports, e se apro tutte le porte per farlo passare, tanto vale eliminare il firewall. Si risolve col buon senso: molte implementazioni di RTP usano uno specifico range di porte, per esempio 4000-4030 ma non è una soluzione standard.

RTP Packet format

V è la versione. CC è il numero di SSRC e CSRC presenti nel pacchetto. M sta per marking, quindi anche RTP fornisce un bit per marcare, perché non è detto che RTP debba viaggiare per forza su IP che ha il TOS per questo. Se uso RTP su livello 3 che non prevede un campo del genere, allora ho la possibilità di marcare con M. Il PT serve a specificare il codec e il SN e timestamp li abbiamo visti. SSRC, CSRC sono legati al multicast in un certo senso, in particolare all'RTP mixer. SSRC è l'identificatore (di solito un indirizzo IP) dell'originatore del flusso. Di CSRC ne posso avere più di uno e sono gli identificatori di tutti i dispositivi che hanno contribuito a creare il flusso. L'RTP Mixer è un dispositivo capace di manipolare flussi RTP, per esempio mixando diversi flussi. S_1, S_2, S_3 sono diverse sorgenti di flussi multimediali.



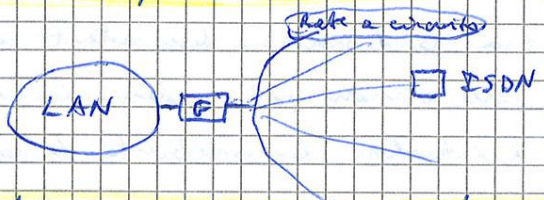
Se le concentro sul mixer, questo è in grado di mixarli e di produrre un 4° flusso verso una certa destinazione D. Se le sorgenti sono audio, ognuna di esse avrà una certa forma e verrà campionata a 8 kHz su 8 bit generando un flusso a

supportare una conn da 4 B e anche una VC sono 4 B. Quindi l'RTP Mixer lo mette dove ho disponibilità di banda, tipicamente in una Data Center dove ho 1 Gbps di banda d'accesso. Vediamo che nè in SIP che H323, l'RTP Mixer è previsto dallo standard ma sta a noi decidere dove metterlo. Tipicamente è un server e non ha quindi le restrittezze di banda dei client. Skype non, ma la certezza non c'è perché il protocollo è segreto, che implementi l'RTP Mixer (o quello che è dato che probabilmente non usa RTP) sul client che avvia l'audio conferenza, dato che usando una applicazione peer-to-peer, non c'è il concetto di server. Quindi è bene che gestiamo la chiamata chi ha più banda. (Se serve vedi collegamento parallelismo RTP Mixer - Relay Server 28:30)

Se vogliamo modellare una rete VOIP dobbiamo considerare 2 situazioni. La prima è tipicamente legata al VOIP, in cui ho una parte di rete a circuito (POTS) e una IP (tipicamente il backbone). Qui serve il Media Gateway già citato, che prende i campioni vocali e li mette nei pacch. (poi dall'altra parte ricevono numeri su circuito). Poi serve il Signaling Gateway perché deve trasportare la segnalazione. Su rete telef. tradizionale, la composizione del numero serve ad instaurare il circuito, ma dovendo attraversare la rete IP, in realtà qui serve per costruire l'ind. IP corretto che è quello del router finale del backbone che mi permette di uscire nel posto corretto della rete a circuito. Se scrivo 011, il backbone deve far sì che i pacchetti arrivino a Torino quando arriva a Torino, serve tutto il numero per raggiungere la dest. specifica quando è chiaro che si deve instradare anche la segnalazione. Serve qualcosa che metta nei pacch. i messaggi della segnalazione a circuito. Poi c'è il Gateway Controller che è l'entità aggiuntiva che controlla gli altri 2 gateway e aggiunge funzionalità minore. Per esempio, posso fare fatturare una segnalazione che arriva ai bordi del backbone IP, e potrebbe essere necessario un meccanismo di autenticazione per entrare nella rete IP. Nella rete a circuito il problema non si pone perché abbiamo l'abbonamento con Telecom. Questo gateway può fare anche operazioni di controllo sul credito per esempio. La seconda situazione è quella di rete omogenea, ovvero completamente a circuito o a pacchetto. In una rete VOIP tendenzialmente non servono dei gateway perché non devo tradurre nè flussi

10/11/14

Protocollo H.323. È stato pensato non per le comunicazioni VOIP come le abbiamo intese finora in cui un client VOIP parla con un altro client VOIP e se proprio serve si usa anche la rete PSTN. Si è partiti dal problema di realizzare audio/video conferenza sulla rete a circuito usando ISDN. Degli utenti di una LAN dovevano poter comunicare con questi dispositivi ISDN. Quindi era un tentativo di abbandonare ISDN dove la tecnologia a pacchetto permetteva di farlo. All'interno di una LAN, basta aggiungere un client VOIP per poter effettuare comunicazioni su IP. Ciò che non si può fare è connettere direttamente al mondo VOIP il client che sta ancora a casa con la connessione ISDN. Quindi lo scopo di H.323 era permettere l'audioconferenza tra disp. connessi alla rete a circuito, tipicamente remoti e ISDN, e disp. connessi alla rete IP via cablato locale. Nasce soprattutto come standard per la comunicazione su LAN perché è una la rete IP e H.323 agisce su essa e interfaccia la LAN col gateway. Poi è stato esteso alla rete WAN, quindi un client connesso a una LAN IP, può parlare con un altro client ma sulla diversa LAN connessa da router, quindi attraverso una rete IP. H.323 supporta le comunicazioni audio, video ma anche dati e per scambio dati non si intende file transfer, ma cose tipo lavagna interattiva condivisa con un utente remoto. Il gateway è un componente fondamentale di H.323 perché interfaccia il mondo a circuito con quello a pacchetto, anche se qui fa poco perché i messaggi sono quelli della rete a circuito incapsulati nella rete a pacchetto senza cambiarli. Il protocollo è stato pensato per semplificare la vita al gateway. Il Gatekeeper è il gestore della rete H.323. C'è anche una Proxy GK che può essere usata da client con limitate risorse. Il GK è un sostegno al server di supporto (registrazione, autenticazione, quindi i client devono registrarsi sul GK per poter usare la rete VOIP, la localizzazione degli utenti). Se non voglio che il client si sobbarchi tutto lo scambio di messaggi col GK configuro e uso il proxy. Per esempio lo scambio di messaggi col GK per la localizzazione del dest. viene fatto tra il GK e il proxy



(così che non dice mai niente!)

connection. È ottimo su Internet perché di solito TCP UDP fanno la error detection ma non la connection. Questo perché è più semplice e comunque si ritrasmette. TCP ignora i timeout e gli ack duplicati, UDP no ed è compito dell'applicazione. La connection è tipica della telefonia e circuito. Il RAS controller gestisce la registrazione, l'ammissione, la chiusura, insomma tutte le operazioni che fa il Gate Keeper all'inizio della comunicazione. Il gateway è importantissimo in H.323. Esso converte i protocolli dati da circuito a pacchetto, così come i prot. di segnalazione e gestisce il controllo del canale. C'è un discreto numero di protocolli da convertire ma per il Gateway è facile perché ha gli stessi uno prima che dopo. Completa l'insieme il GK è responsabile di una particolare zona, gestisce l'admission control e l'autorizzazione, l'autenticazione tramite i messaggi RAS. In modo opzionale gestisce il controllo della banda, cerca di realizzare una soluzione per la QoS. Se tutti gli utenti registrati su una zona, vanno dal GK per iniziare una chiamata, esso può tenere traccia del num. di chiam. Attive e cerare di prevenire la perdita delle conversazioni ponendo un limite a tale numero. Se tal numero è 10, l'11° chiamata non la fa fare. È un tentativo di un'applicazione di controllare la QoS ma non è una gran soluzione perché dipende dalla banda occupata da ogni tx e soprattutto non è detto che un utente debba fare passare dal GK per raggiungere un altro utente. Il GK gestisce la localizzazione, provvede cioè all'utente l'IP address del destinat., ma se ce l'ha già, può semplicemente dal client e bypassa il GK. Per H.323 non c'è modo di bloccare tale chiamata. NetMeeting di Microsoft usava H.323 senza GK. Gli accessori H.323 sono nella forma nome@domain.com ma può anche avere dei numeri telefonici veri e propri (E-164 è lo standard per gestire i num. di telefono). Questo però comporta di usare il GK, altrimenti si usa IP address/porta. Trovare il server che trasforma il nome in un indirizzo IP è relativamente facile perché potrei usare, come nella posta elettronica il DNS con record MX che mi manda al server responsabile per quello dominio. Può però anche

zone

SIP in particolare realizza una segnalazione END-TO-END, quindi non tocca i router. Per il data plane si appoggia a RTP/RTCP, per la QoS basata sulla riservazione delle risorse si appoggia a RSVP. SDP viene usato per la descrizione delle sessioni VoIP da gestire.

SIP supporta anche il mapping dei numeri come H.323 e gestisce la personal mobility (che non è la mobility) detta anche Nomadicità. Se ho un account SIP e sono connesso alla LAN del Politecnico il dominio SIP fa sì che venga raggiunta dall'indirizzo IP che ho in questo momento. A cosa sono raggiungibile con lo stesso account SIP non altro end IP. Questa è la nomadicità, non il fatto che durante la chiamata cambio la cella wireless, UMTS o quello che è. Quella è la mobilità (senza personal) che non è gestita da SIP, sono problemi della rete IP. SIP è un protocollo di tipo client-server quindi c'è un router che supporta da qualche parte a cui i client chiedono le opportune funzionalità, come la registrazione, la localizzazione dei client, etc... Questo in generale, ma anche nella singola chiamata END-TO-END c'è il concetto di client-server anche se abbiamo una rete PEER-TO-PEER a tutti gli effetti. Chi inizia la chiamata è il client, e su ogni dispositivo c'è un modulo server che esegue su una porta pronto a ricevere i messaggi SIP, e quindi è a tutti gli effetti un server. SIP usa messaggi HTTP like, quindi formato testo. C'è una riga di comando, una serie di header, e se serve un corpo del messaggio SIP può funzionare sia su TCP che UDP o TLS, ma non stiamo parlando del Data Plane. I messaggi SIP per gestire la chiamata possono viaggiare su tutti e 3. TLS è la nuova versione di SSL (è un TCP sicuro). Il piano dati tipicamente è gestito con UDP/RTP. Se trasferiamo i messaggi SIP su UDP evitiamo di instaurare la connessione TCP ed è più semplice, è utile su dispositivi embedded che hanno memoria e potenza di calcolo limitate. In realtà anche se il client gira su PC, in genere usa UDP perché è il modo più semplice. La sequenza di messaggi di segnalazione non è lunga quindi non c'è l'esigenza di mettere su piedi una connessione TCP, serve (come per il DNS) gestire le ritrasmissioni a livello applicativo. Ci sono

con un protocollo a parte ma senza poco diffuse le applicazioni che supportano solo sessioni.

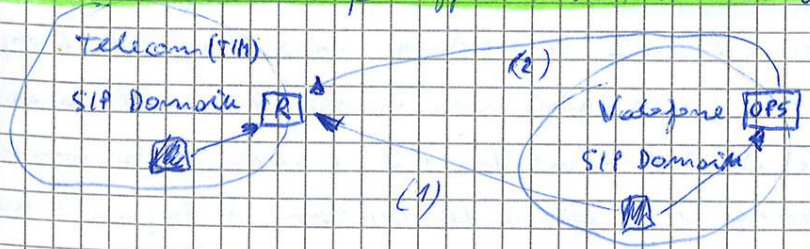
SDP (Session Description Protocol)

Protocollo per gestire le sessioni che vengono con SIP. Se effettuo una chiamata, devo comunicare quali codec usare, quale tipo di chiamata voglio effettuare, alla controparte. Devo dirle quali flussi multimediali saranno trasferiti nella rete. SDP non c'entra nulla con SIP, è stato standardizzato per descrivere delle sessioni e lo si è usato anche creare una nuova. In esso mettiamo il numero di stream multimediali, il tipo di sessione (audio, video etc.) il codec da usare, il protocollo di trasporto (UDP con RTP o senza etc...). Non tutte le funzionalità di SDP sono utili a SIP (per esempio una sessione e quasi di un film in streaming. In una chiamata non so quando finirà). Anche SDP è un semplice testo ed è incluso nel body del messaggio SIP. È composto da 2 parti: la prima riguarda la sessione tra i 2 utenti, e la seconda che riguarda i flussi specifici. Si capisce che sta iniziando una parte di sessione quando leggo "v=" (si riferisce alla versione). In questa parte c'è una serie di attributi che descrivono l'intera sessione. Potrei specificare un algoritmo di crittografia con cui cifrare tutti i flussi. Poi devo descrivere ogni singolo flusso multimediale (video-chiamata sono minimo 2 → audio + video). Nella parte di Media quando ci sono tante sottoparti quanti sono i flussi da gestire (dalla "m=" si capisce che inizia una parte Media). m= audio 3456 RTP/AVP 96 indica che la sessione contiene un flusso audio, poi c'è la porta UDP usata. RTP indica che viene usato, e si capisce anche che la porta è UDP perché RTP viaggia su UDP. Poi c'è un modo per specificare il codec. Poi ci sono un'altra sezione per il video

13/11/14

Nella parte di sessione possono avere come attributi "v" che indica la versione. "k" è l'algoritmo di encryption che verrà usato per tutta la sessione. Alcuni attributi come "a" o proprio "k" vanno bene anche nella parte di media. Abbiamo visto cosa si mette nell'attributo "m". RTP non usa porte standard,

di server. Il Registrar Server è quello a cui registrarsi per accedere a un certo dominio SIP. Devo comunicargli lo UserName così lui può fare il mapping tra esso e l'attuale mio indirizzo IP. Ricordiamo che SIP supporta la mobilità. La registrazione può essere vincolata ad una autenticazione e interviene quindi l'AAA Server. Il RS può interrogarsi per decidere se un certo utente può accedere al dominio. L'AAA Server tipicamente non fa parte della rete SIP (sarà un RADIUS), ma dell'azienda e andrà ad autenticare gli utenti per diversi servizi. Il Location Server mi permette di localizzare gli utenti da contattare. Esso sarà il RS su cui è registrato l'utente da chiamare perché lì c'è il suo indirizzo IP. Il Redirect Server permette di redirigere le chiamate su diversi devices. Può essere configurato per esempio, per fare sì che di notte arrivi le chiamate sulla segreteria telefonica, di mattina sul cell., di pomeriggio sul fisso. L'MCU è l'RTP Mixer di SIP. Il Media Server è quello con cui interagisco tramite RTSP, quando contatto la segreteria per esempio, o magari la musichetta che parte quando mi mette in attesa qualcuno (è uno streaming audio). Il Media Proxy è il famoso relay (per passare da NAT/firewall). Esso serve a gestire il relay dei flussi RTP, della chiamata vera e propria una gli stessi problemini si hanno coi pacchetti SIP. Per tali pacchetti vedremo come fare il Relay. Ovviamente non tutti questi server devono essere presenti ogni volta. Il Gateway qui, chiaramente con H.323 deve fare molte più conversioni. Il Proxy Server è simile a quello di H.323, serve ad aiutare i terminali con poche risorse di calcolo. Scambierà messaggi col Reg. Server per la registrazione/autenticazione, col LS per la localizzazione. Al giorno d'oggi può avere senso in una rete di sensori, dato che questi hanno poche risorse. E' outbound Proxy Server invece? Se abbiamo 2 operatori di rete, è ovviamente se mettono d'accordo per offrire i loro servizi di telefonia mobile basandosi su SIP. Se sono un utente di TIM, ho un account nel dominio TIM.it e quindi mi registro sul RegS



Addressing

Abbiamo visto quello simile alla posta elettronica ma c'è una seconda possibilità: telephone-number gateway anche se poco usata e la trascuriamo. A differenza dei numeri di cell, il numero è associato all'utente e non al dispositivo; per via della mobilità. (Slide 21 NO).

Ma vediamo come avviene la localizzazione nel contesto di SIP. Il problema è localizzare il SIP server, che contiene il mapping tra utenti e ind. IP. Non è una cosa banale perché ci sono tanti SIP server (1 per ogni dominio) e non si può tenere tutta la lista, non sarebbe scalabile. Si usa il DNS nel modo che si usa per la posta elettronica. Se devo inviare una mail a luigi@google.com, uso il server SMTP del Politecnico. Il Poli per consegnare la mail a luigi deve reperirla nella sua coda nel server di posta di Google. Per trovare questo lancia una query MX per Google.com. Tipicamente nel server autoritativo del dominio Google.com ci sarà un record di tipo MX che returna l'IP address del server di posta di Google.com. Qui il problema è analogo ma non si può usare MX che è già usata per la posta. Si lancia una query SRV per google.com che viaggerà tra server DNS. Se non c'è in cache arriverà nel server autoritativo di Google dove ci sarà il mapping tra il SIP server di Google e il suo indirizzo IP. Per la posta però uso TCP sulla porta 25 & altro se voglio la sicurezza. Ad ogni modo porta e protocollo sono ben definiti. Qui però possono avere servizi SIP basati su UDP, TCP o TLS quindi nel file config del server autoritativo di un certo dominio ci sono 1 entry per TCP, 1 per UDP e 1 per TLS. Quindi se voglio localizzare un SIP server in grado di offrirmi un servizio SIP su UDP, la prima parte del messaggio sarà _sip._udp.foo.com dove foo.com è il dominio. Ogni riga è "duplicata" una si riferisce ad un altro server. Il numero dopo SRV è la priorità (più alta è associata a 0). Il server 2 è contattato solo se il primo non è raggiungibile. Si può anche pensare di lasciare le priorità uguali e agire sul numero dopo. Il 2 significa che ogni 6 query 4 volte si contatta il server 1 e 2 volte il server 2. Non sottende alcun numero,

messaggio di INVITE ad una certa punto il dest. alza il telefono

e così invia un 200 OK che implicitamente fa un ack all'INVITE. Poi parte il flusso RTP. Dal punto di vista della segnalazione SIP, usando UDP non avrei modo di confermare a B di aver ricevuto la 200 OK. Quindi con TCP c'è il suo ACK, con UDP questo. In realtà lo mando anche con TCP, perché anche l'ACK può inviare un messaggio SDP. Per esempio in INVITE dico che voglio usare il codec ϕ 3 5. B nella 200 OK dice che lì d'accordo su ϕ e 3, e con l'ACK A decide di usare ϕ. RTP permette anche di cambiare codec e l'ACK può servire a confermare. Con BYE si termina la chiamata. Con CANCEL cancello le richieste pendenti di un setup di chiamata. SIP permette di essere loggati al contempo su più dispositivi, e quando qualcuno cerca di contattarmi, SIP deve gestire una fork della INVITE. Tutti i dispositivi squilleranno, ma risponde solo uno dei tanti. La rete SIP manda quindi un CANCEL agli altri per farli smettere di squillare. Con OPTIONS si scoprono le funzionalità disponibili su certo User Agent. SUBSCRIBE e NOTIFY sono usati per l'e-presence, mentre MESSAGE permette di trasferire un messaggio testuale.

Main SIP headers

Gli header sono info aggiuntive che descrivono meglio il messaggio inviato. From e to sono gli identificativi di chiamante e chiamato, quali fanno una data comunicazione SIP. Contact permette ad uno user agent (o un proxy) di comunicare ad un altro il proprio IP address, oppure a un proxy di comunicare ad uno UA l'IP address di un altro. Sapendo l'IP posso poi mandargli direttamente i successivi messaggi SIP e il flusso RTP senza passare dal SIP server tutte le volte. Con alcuni NAT dopo una prima triangolazione su un Relay, posso andare direttamente no per i messaggi SIP che per il flusso Media. Con alcuni NAT non si può fare e per il flusso Media si passa dal Media proxy e anche i SIP devono passare dai proxy. I proxy possono settare un header per vincolarmi a passare sempre da loro.

Simile ad HTTP, e 200 OK è proprio uguale. Esso si manda quando si riesce a gestire in modo corretto tutti i messaggi (BYE, REGISTER etc..) (Slide 37 NO). Il 401 -> Unauthorized si vede spesso. (Slide 39-40 NO)

Esempio (Slide 41)

È di tipo INVITE. Dopo il messaggio type ci sono gli Headers. Il VIA indica che si è passato da gli. dom. com. From -> Chiamante, To -> Chiamato. Il SIP server si capisce che è DOM.EDU confrontando VIA e FROM. Siamo passati dal SIP server del chiamante, perché esso e il SIP Server che ha inserito il VIA sono nello stesso dominio DOM.EDU. La riga content-type ci fa vedere che SDP è catalogato come application nel contesto di MIME. Poi c'è una riga bianca e dopo il corpo della sessione. Ad un certo punto il chiamato alza la cornetta e risponde con un 200 OK Gli headers VIA sono 2 perché l'INVITE è passato precedentemente dal SIP server del chiamante come abbiamo visto, e sicuramente sarà passato (non si vede nella slide) dal SIP S. del chiamato perché doveva essere localizzato in qualche modo. È chiaro che al contrario dovrà ripassarci partendo da quello del chiamato. Nella parte Media di SDP il chiamato conferma tutto tranne il codec 5. (Slide 42 NO).

Adesso si capisce dove viene usata la risoluzione DNS. Essa si usa per due tipi di operazioni. La prima è quella legata all'invio di una INVITE. Partendo dal SIP Server che funge da proxy qui, ed esso lancerà una risoluz. DNS di tipo NAPTR e poi SRV e A, per scoprire dove è il SIP server del chiamato. Questa procedura la si fa anche col messaggio REGISTER che lo fa per trovare il proprio SIP Server (anche se potrei config. nello User Agent) per inviargli appunto dale messaggio. Ci sono casi particolari in cui nello User Agent scrivo qualcosa del tipo SIP.server.polito.it / 5060 (porta tipicamente usata) e allora serve solo una risoluzione di tipo A. Un altro caso strano è che faccio una Discovery del SIP Server ma che ha configurato il server DNS del dominio, non ha messo il record NAPTR. Non ricevendo risposta dovrei fermarmi, ma invece alcuni client vanno a tentativi partendo proprio da UDP, che tipicamente funziona sempre, con una SRV. Se non funziona provo

Supponendo che la location Procum abbia successo, dopo aver propagato l'INVITE posso dire allo UA che ho ricevuto il messaggio, l'ho elaborato, ma ancora non so cosa succederà dopo averlo propagato, il 200 OK non posso mandarlo e mando un 100 → Trying. L'INVITE arriverà al SIP S. del chiamato che risolve il mapping e lo forwarda verso la UA finale. Anche qui c'è un 100. Il telefono squilla e con un 180 → RINGING si sincronizzano tutto a modo di questa cosa. I passaggi tra i vari SIP S. si ottengono tramite gli header VIA che i 2 S.S avevano inserito nell'INVITE. Quando il 180 arriva al chiamante, esso sente il tono di quando dall'altra parte il telefono squilla. Appena il chiamato alza la cornetta, parte il 200 OK fino al chiamante. Tutti i server intermedi lo ricevono perché è giusto informare la rete, non solo per l'eventuale tariffazione. In tutti i messaggi visti, ce ne è uno di ritorno che fa esplicitamente da ACK, mentre per il 200 OK non c'è, ergo lo invio esplicitamente perché potrei essere su UDP. Eventualmente nel body metto una SDP se serve, magari per decidere tra i codec proposti. L'ACK va direttamente fino al chiamato, a meno che non è settato un Record Routing, perché i 200 e 180 hanno portato al chiamante l'header Contact. Non c'è il Media Flow, e quando uno dei 2 mette giù parte il BYE e l'ACK dopo. Per la forma, questo scambio di messaggi si chiama Trapezoide SIP, ma se i 2 UA sono nello stesso dominio, collassa in un triangolo (Unico SIP Server)

ENUM standard: E.164 addresses

Non è legato necessariamente a SIP, ma in generale a VOIP (e forse non solo). Gli account definiti sono buoni per Smartphone, PC, ma non per i tradizionali telefoni a tastierino (scrivere UserName@Domain.com non è comodo). Ci vuole un modo per assegnare ad uno UserName SIP anche un numero telefonico (una rec. E.164). Se compongo un numero da un terminale SIP, come fa questo a sapere che è un num. telefonico vero e proprio da cercare sulla rete e cercarlo (andare verso il gateway) e non è un numero SIP che posso raggiungere attraverso la rete IP? Interviene ENUM che usa il DNS con un metodo molto prossimo alla query inverso

perché il num. è associato allo UserName SIP.

PSTN to IP connection

Il num. è +39-011-090 7027 e la rete PSTN capisce che +39 -> Italia, 011 -> Torino, 090 -> Politecnico e arriva subito al Gateway del Poli. Ora supponiamo che dentro il Poli ci siano sia utenze a pacchetto che a circuito. Il Gateway deve capire a prescindere nella rete a circuito o uscire nella LAN Ethernet per raggiungere l'utente tramite IP. Interroga quindi il DNS come visto, che ritorna la lista dei servizi disponibili tra cui SIP. Capisce che è un utente SIP e interroga di nuovo col DNS per trovare l'ind IP del SIP Server (se non lo ha già). A quel punto questo inoltra la INVITE all'utente

IP to IP connection

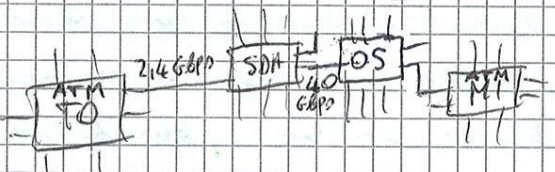
Devo capire se andare verso il gateway o rimanere nella rete IP. Chiamo il num., parte la query DNS e questo rimanda la lista indietro. Capisco che l'utente è di tipo SIP e con la solita procedura contatto un IP dell'esterno (Tutto il resto lo salto fino alla 62)

Il protocollo sta avendo molto interesse da parte dei tecnici che si occupano delle installazioni telefoniche. In questo momento una rete VoIP, la si installa tendenzialmente con SIP e non H.323 o altro. Ci sono alcune problematiche nella comunicaz. IPV4 -> IPV6, e con il NAT / firewall (Se serve riascoltare esercizi simili)

17/11/14

MPLS è molto importante perché permette agli operatori, ai servizi provider, di superare dei limiti intrinseci del protocollo IP. IPV6 alla fine dei conti ci permette solo di superare i probl. dello spazio di indirizzamento IP non è pensato per offrire servizi commerciali e per questo ci viene MPLS (molto usato a differenza di IPV6 fino adesso). Si è investito e lavorato per standardizzare MPLS ma ci voleva del tempo, così i produttori di apparati di rete si sono uniti in associazione come MPLS forum per mettersi d'accordo e far funzionare i loro dispositivi insieme fin da subito. Il problema pratico è che si vuole fare una rete pubblica (cioè non di uno solo) perché si vogliono creare

parte della rete posso e ci si deve assicurare che tale parte abbia abbastanza risorse. Una tecnologia che funziona è ATM che però è molto sofisticata e quindi ha costi elevati. I commutatori ATM hanno tante interfacce che costano molto dato che sono sofisticate. Se c'è un nodo a Torino e un altro a Milano, può non essere economicamente conveniente avere un'interf. ad altissima velocità che esce dal nodo. Quindi da un lato tra TO e MI c'è bisogno di portare tante interconnessioni, dall'altro se dovessimo farlo solo con switch ATM, le loro interf. dovrebbero essere velocissime tipo 40 Gbps e costerebbe troppo. Si risolve collegando lo switch ATM di TO ad uno switch SDH. SDH è una tecnologia a commutazione di circuito che permette di creare, attraverso una rete di commutatori, dei circuiti a velocità predefinita. Ora lo switch



SDH avrà magari da una parte 2,4 Gbps e dall'altra una velocità maggiore, per esempio 40 Gbps. Tutto il traffico che arriva da TO da tanti switch ATM, viene messo insieme usando uno switch SDH/Sonet e inviato a MI su un canale a 40 Gbps. Questa cosa all'operatore conviene anche perché tale rete SDH tra TO e MI ci lavora già, l'interfaccia dello switch ATM gli costerebbe tanto, così come la nuova fibra per collegare i due switch tra TO e MI, quindi è meglio usare la rete SDH. Ora il canale da 40 Gbps tra TO e MI richiede una fibra, che però è bene usare per metterci più canali perché costa tanto e quindi a volte dello switch SDH c'è uno switch ottico che prende tanti canali da 40 Gbps provenienti da diversi switch SDH e li mette insieme su un'unica fibra usando la tecnologia WDM (Multiplazione tante lungh. d'onda, tanti canali ottici, su un'unica fibra). Ecco la "cipolla": ci sono switch ottici collegati a delle fibre a cui sono collegati switch SDH ai quali sono collegati switch ATM. Tutto ciò per portare un giro dei pacchi IP quando ci sarà un router IP, la cui interf. verso lo switch ATM

in qualche modo. Parla da tanti utenti e i utenti ed entrambi parlano la stessa lingua. È una tecnologia per il backbone delle reti e questo è un altro vantaggio perché se arrivano al Desktop, questo dovrebbe capirlo e aggiornarsi ai nuovi cambiamenti. Con IPV6 tutte le applicazioni devono poterne girare e ciò è stato un disincentivo. Esistono dei meccanismi di transizione IPV4 -> IPV6 basati su MPLS. Le nuove reti si fanno con MPLS ma in zone periferiche etc.. si usano ancora le vecchie tecnologie. L'idea base di MPLS è quella di mettere un'etichetta davanti ai pacch. IP e la commutazione dei pacch. la si fa in base a quell'etichetta. I router non guardano più dentro il pacch. ma decidono dove mandarlo in base alla label. In tale modo si velocizza perché non si accede alla tab. di routing dove ci sono prefissi + Next Hop (in IPV6 il prefisso è un prefisso con la sua lunghezza, mentre in IPV4 è una coppia Indirizzo/Network). Per trovare il Next Hop il router deve fare un match facendo una AND bit a bit tra l'indirizzo dest. del pacch. e la Network della tabella. Partendo dalla prima riga, quando ne trovo una che fa match non ho conclusione perché potrebbero esserci altre e questo vale anche (soprattutto) in IPV6 per il discorso dell'aggregazione degli indirizzi. Si deve quindi prendere la riga che fa match e ha il prefisso più lungo. In realtà ci sono degli algoritmi sofisticati per non dover guardare ogni riga, ma un trucco c'è ma full scan della tabella. Mettendo quella label non si troverà più di una LPM ma dovrà trovare la riga corrispondente all'etichetta, con il Next Hop. Inoltre c'è bisogno di un'etichetta per ogni comunicazione ma sono sul link quindi sono relativamente poche (sono da 20 bit). Siccome non sono molto lunghe le si può usare come un indice (pacch etichetta 5 -> guardo la riga 5 etc..). Così non accedo alla tab. trovo il Next Hop. Con gli IP address non si può fare perché sono 2^{32} (2^{128}) e non li si usa mai tutti. Inoltre tutti quelli che iniziano per 10.0 ad esempio, hanno lo stesso Next Hop e la cosa diventerebbe ingestibile. I router girano a questa label, e perciò si velocizza

È chiaro che ad un certo punto vorrano comunicare e concentrarsi, una FaceBook avrà un Data Center con 100 mila server ad esempio, distribuiti in un certo modo. Alla fin fine MPLS introduce un paradigma connection oriented per le reti IP. ATM e FR fanno commutazione basata sulle etichette. Una bella caratteristica di un protocollo connection oriented è che prima di comunicare si stabilisce una connessione. Lo svantaggio è che se devo mandare 1 solo pacch., con un invio diverso prima per creare la connessione, cosa inefficiente. MPLS non impone di creare la connessione ed è una cosa particolarissima perché unisce i vantaggi del CO e del CL. ATM, FrameRelay, SDH, sono tutte CO perché progettate per fare reti pubbliche, diversamente da IP. quindi in MPLS ci sono le etichette ma gli END System non ne devono sapere nulla. C'è una zona della rete dove c'è una MPLS che si chiama MPLS Network e ai suoi bordi ci sono dei Label Edge Router. Dentro la rete invece stanno i Label Switch Router, che guardano l'etichetta per capire dove inviare i pacchetti. Differenza tra Switch e Router. Il primo guarda l'ind. dest. e se lo conosce, inoltra il pacch. dalla porta in cui l'ha ricevuto ad un'altra, se non lo conosce lo rinoltra su tutte le porte. L'operaz. di spostamento dalla porta in cui l'ha ricevuto a quella in cui inoltrarlo è detta switching. L'operaz. di scegliere la porta in base all'indirizzo MAC (di dest.) si chiama routing. Un router fondamentalmente fa la stessa cosa, concettualmente non c'è differenza. Negli switch si vuole in evidenza che sono molto veloci a spostare i pacchetti. Esso ha tante porte di IN e tante di OUT e la funzionalità in cui è più bravo è spostare i pacchetti velocemente, da qui il nome switch. Il router invece ha un modo più intelligente di scegliere la porta d'uscita, la strada, e da qui il nome Router. Anche Bridge e switch sono la stessa cosa solo che il Bridge (nome invisibile) aveva poche porte. Quando li hanno inventati a fare con tante porte li hanno chiamati switch per mettere in evidenza la capacità di spostare i pacch. dalle porte di ingresso a quelle di uscita. Ora abbiamo gli LSR che sono bravi sia a switchare che a fare routing e

per X nella MPLS cloud, sapendo dove si trova X. Le due porte della rete hanno usato la stessa logica. Se tutti i nodi LSR fossero dei router IP i pacchi seguirebbero la stessa strada che seguono così. La logica di base è la stessa perché MPLS è CO e si possono fare delle migliorie.

19/11/14 Affinché i pacchetti siano instradati nella MPLS Cloud, bisogna creare l'equivalente di una connessione (anche che MPLS non è necessariamente connection oriented). Si parla di Label Switched Path o di LSP tunnel. I nodi si mettono d'accordo per far sì che un pacchetto che deve andare da un certo punto di ingresso ad uno di uscita, passi da questo tunnel. Se un pacch. vuole andare da un indirizzo IP D (uno stesso di prima), i router esterni ad una certa punto lo faranno entrare in una MPLS Cloud scelta con logica e dove è D. Il LER (Ingress LSR) guarda la sua tabella in cui, dato un indirizzo IP destinato, c'è riportato l'etichetta da aggiungere e la porta su cui inoltrare il pacchetto. Il pacch. arriva poi all'LSR X che ha una tabella diversa, guardano alla sola label. Capisce per esempio che un pacch. con label verde va inoltrato nell'interfaccia 2 e la label va cambiata (Label swapping) da verde a grigia. Il successivo LSR cambia ancora etichetta (grigio - nero) etc. Alla fine l'ultimo LSR lo inoltra su una porta in base alla sua tabella e toglie la label (di nuovo pacchetto IP). Da notare che tutti i pacch. che all'inizio ricevono l'etichetta nera, fanno tutti la stessa strada. Non è detto che D sia un indirizzo, può anche essere un prefisso, infatti tutti i pacch. verso un prefisso fanno la stessa strada nella cloud. La label ha validità solo sul link tra 2 LSR ed ecco perché la si cambia. Se ne scegliessimo una che non viene scambiata ci sarebbero problemi di instradamento a meno che ogni label non sia univoca nella cloud. Chi decide la label dovrebbe sapere tutte le etichette usate dagli altri LSR (grossi scambi di info). Inoltre se avessimo 1 milione di LSP ci vorrebbero 1 miliardo di etichette, e tali label dovrebbero lunghe (non si potrebbero più usare come index). Quindi con lo swapping

commutazione. In realtà stiamo parlando di logica nella scelta dei percorsi, la commutazione tecnicamente deve solo spostare da ingresso a uscita. Si è quindi generalizzato i protocolli con gli standard G-MPLS. Quando MPLS è nato per fare router veloci, non piace a tutti perché fornisce un piano di controllo unificato. È vero che ci permette di liberarci della "cipolla" ma permette anche ai vari vendor di implementare meglio è vero che lo stato ATM fa la commutazione guardando delle etichette all'interno delle celle ect... ma se la logica per scegliere etichette e percorsi è la stessa che usano i router IP, è intuitivo che la rete funzioni meglio. MPLS introduce un header da mettere davanti ai pacchetti IP per contenere l'etichetta ed è la parte fondamentale ma meno importante, perché il grosso vantaggio non è la velocità di commutazione in base all'etichetta, ma l'uso dei protocolli per mettersi d'accordo sulle etichette, e di quelli per scegliere i percorsi da mettere all'interno della MPLS cloud. Questi sono i normali protocolli delle reti IP potenziati. Li potenziamo perché nelle reti IP si cerca il percorso migliore, ma MPLS è una orientata connection oriented e ci sono situazioni in cui non si vuole per forza il percorso migliore, ma quello migliore che soddisfa certi vincoli, per esempio i cui bank non vuole banche oltre il 75%.

L'header di MPLS si chiama Shim header e viene messo tra la trama Ethernet e il pacchetto IP. Ora non ho una lunghezza fissa. C'è un modulo di base di 32 bit e poi se ne possono aggiungere altri (con trama e pacchetto che si allontanano → ecco perché Shim = Cuneo). Ogni modulo ha una label di 20 bit (grande ma non di certo quanto i 128 bit di una sub IPV6) Il TTL viene decrementato da ogni LSR e quando arriva a zero viene scartato. Ora serve a non dover supporre che non ci siano loop nelle reti, cosa molto problematica (si deve usare lo spanning tree e la cosa diventa estremamente inefficiente), ma permettere che ci siano per un breve periodo. È chiaro che i protocolli di routing devono occuparsi del problema e cambiare le tabelle, ma senza il TTL, anche se per poco tempo bastassimo ci fosse un loop nella rete, i pacchetti non uscirebbero mai e man mano

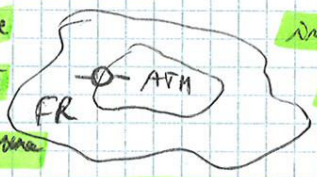
formato ma abbiamo detto che quello visto è voluto. FR ha un'etichetta che potrebbe essere più corta (16 bit (FORSE) al massimo), ma basta standardizzare la cosa e siamo apposto. Resta da fare in modo che per scegliere labels e percorsi si usino i meccanismi di MPLS. Con questi accorgimenti si ricicla un apparato FR/ATM che diventa una switch MPLS. L'HW rimane uguale e quando la commutazione (ricevere i pacch., guardare la label, scegliere la porta d'uscita) detto che è fatta in HW. La logica detta invece è SW e si può cambiare. Sono guadagni standard entrambi i modi di mettere l'header. Anche un switch in grado di commutare trame di liv. 2 con label diventa è ufficialmente uno switch MPLS e potrà usare i suoi protocolli per il piano di controllo. Le label delle trame ATM si chiamano VCI/VPI e quella di FR DLCI. Se devo fare il push di una label, nel caso classico di MPLS

ETH	4	3	IP

aggiungo un header in cima come visto. Nel caso mia mezza dentro l'header di liv 2 ATM/FR, se ma aggiungo una classica (l'header da 32 bit) dopo IP e si sovrascrive la label dell'intestazione ATM/FR. Tanto i commutatori vedono solo la più esterna.

ATM/FR	4	3	IP

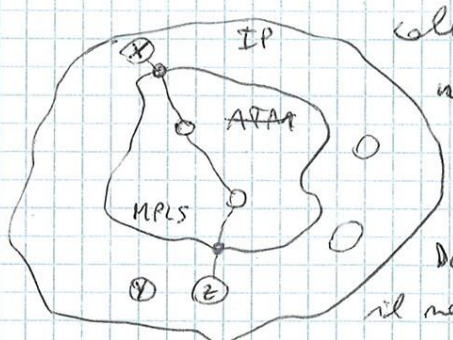
Se avessimo due apparati al confine tra FR e ATM che deve l'unica cosa che deve fare è FR e smontarlo in una trama



smontare un pacch. verso ATM, rimontare il pacch dal liv 2 di liv 2 ATM, e scrivere la label in modo diverso. Tutto ciò se tratto gli apparati usano il piano di controllo MPLS. Prima tale passaggio era un punto critico. Più etichette sono usate per ragioni di scalabilità, per creare delle MPLS cloud annidate. Ogni cloud aggiunge una label e guarda solo quella che ha aggiunto lui. Se in una cloud esterna arrivano 2 pacch., 1 con label rossa e l'altro verde, e l'LSR di confine a entrambi aggiunge un'altra label rossa, i 2 pacch. faranno la stessa strada nella cloud interna e usciranno nello stesso punto. Poi, tornato a essere 1 rosso e l'altro verde, nella cloud esterna prenderanno diverse strade. In genere ATM/FR è la cloud più interna, quindi gli arriva un pacch. IP + l'etichetta MPLS da 32 bit (se prima c'è Ethernet mettiamo) e siamo messo tutto dentro una cella ATM con la nuova label nel campo della label ATM. Se ATM è la cloud esterna, a quella interna Ethernet arriva la trama ATM con la label messa lì, e l'apparato di confine dovrà rimuovere il tutto mettendo dopo il pacch. IP lo shim header col valore dell'etichetta che stava nella trama ATM e poi un nuovo shim header con la nuova etichetta. È solo una questione di switch.

Y viene avvisato da 2 di mandare i pacch. per la Dest. D con label 3, lui ha scelto per quelli che manda X la label 5, quando può fare il mapping. Questo lo può fare perché quando ha scoperto che esiste D, ha anche scoperto che la strada migliore per andarci è verso 2 (ha una tabella di routing che non usa perché non è un router IP ma ce l'ha, e il next hop per D è 2). La riga sarà del tipo (5 3 2). Se 2 si rende conto che per arrivare a D deve passare per un router IP, fa una POP della label. Se X fosse collegato ad un router IP, allora sarebbe un LER e si creerebbe la riga (D 5 Y). Quando tutti gli LSR sul percorso, compresi i due di confine, hanno fatto mapping, allora l'LSP è creato. Se ho dei router attaccati a delle interfacce Ethernet, basta cambiare del SW per renderli router MPLS perché le interfacce restano le stesse. Per la tab. di routing si possono usare i protocolli di IP, ergo l'unica cosa da fare è scrivere il SW che legge lo shim Header e la tratta pron, la parte di SW che fa il look up, e la parte di SW per lo scambio delle labels (anche se a livello il verso si è partiti da un protocollo esistente nato per altre cose). In realtà per gli apparati veloci la parte di look up delle labels andrebbe fatta in HW. Se ancora, si possono invece gli apparati FR/ATM, dopo aver estratto la label, il look up della tabella sarebbe già disponibile in HW, come è implementato anche tutto il resto. MPLS aggiunge solo la logica con cui fare il mapping che è SW. Funziona a datagram unsolicited, quando gli LSR scoprono una dest., scelgono delle etichette e se le comunicano, e abbiamo LSP predefiniti prima ancora che arrivino dei pacchetti per tale dest. Non c'è quindi lo svantaggio delle reti connection-oriented di dover, prima di trasmettere, creare la connessione. Inoltre una rete tech MPLS dovrebbe anche funzionare da router IP, quando se non sono state scelte delle etichette, i pacch. vengono instradati lo stesso. È comunque una scelta perché un LER può decidere di bloccare un pacch. a cui non sa che label dare, e aspettare il meccanismo di creazione dell'LSP prima di instradarlo.

Se abbiamo un core C.O come ATM e dei router IP intorno, è vero che possono collegare i router IP attraverso il core, ma quando un router riceve un pacch. per la dest. D, come fa a sapere a quale dei router in giro deve fare una chiamata (deve aprire una connessione)? Dovrebbe sapere l'indirizzo ATM di quel router. Quando il router X che vuole mandare un pacch. attraverso la nuvola interna ATM, deve sapere il num di telefono del router da chiamare.



dei pacchetti per D. Basta aggiungere l'info sull'etichetta al protocollo di routing, in particolare al BGP. Hanno scelto BGP perché è facilmente estensibile.

Usare un protocollo di routing per distribuire le etichette funziona solo nella modalità Topology based perché possiamo decidere di distribuire una etichetta solo quando riceviamo una info di routing topologica e la possiamo ad altri. Se vogliamo distribuire le etichette con una Explicit creation of LSP serve un protocollo diverso e CISCO (che inventò MPLS, allora detto Tag Switching) arrivava ad altri, prevedeva per usare il suo protocollo TDP. Altri costruttori hanno spinto per usare un altro protocollo che tutti hanno nei loro router, RSVP (Resource reservation protocol) che era stato inventato per prenotare risorse nei router, cosa che serve per poter dare un servizio con una certa qualità nota e priori. Per far sì che i pacch. vadano da Torino a Roma in un certo tempo, bisogna assicurarsi che ogni uno dei router nel percorso, abbia le giuste risorse per poterlo fare. I router si avvisano a vicenda di queste cose e si mandano messaggi di conferma. Basta poco per considerare lo scambio di etichette, cambiando un po' il pensato. Tutte e due le soluzioni sono diventate standard, sia LDP (TDP) che RSVP, lasciando che fosse il mercato a decretare la migliore. Il mercato finì per orientarsi verso RSVP, e LDP oggi, seppur valido, è deprecated da IETF, la quale non produrrà nuove funzionalità per lo standard. LDP e RSVP si possono usare sia on-demand che unsolicited, e possono sia funzionare Topology driven, solo che non è detto che y usi BGP per comunicare a x la label decisa (può farlo con altri protocolli di routing), che control driven. BGP non può funzionare control driven perché i protocolli di routing non si possono usare per chiedere a qualcuno di fare qualcosa, essi sono fatti per annunciare. RSVP va molto bene con MPLS perché la prenotazione viene fatta presso dal nodo a valle e la label possiamo vederla come una risorsa scelta quindi dal nodo a valle.

Rimane da parlare dei protocolli di routing che hanno un impatto nella fase di label mapping. Se un nodo sa che label si aspetta il nodo a valle, vuol dire che deve passare per andare verso D, e ciò richiede a monte una scelta del routing. Il protoc. di routing è usato per guidare il LM, ma siccome poi i pacch. sono instradati in base alle tab. di forwarding, ciò determina la route dei pacchetti. Si usano protocolli esistenti usati con IP, come OSPF, IS-IS, e BGP-4. Essi servono ai router per capire quale è il NEXT HOP per una destinazione, ma per fare ciò si passa da uno stato intermedio che è capire come raggiungere le varie