Voice Over IP

NAT Traversal

Giorgio Zoppi

zoppi@cli.di.unipi.it

Tecnologie di Convergenza su IP
a.a.2005/2006

Che cosa è il NAT (Network Address Translation)? (1)

Definizione 1.1. Un regno d'indirizzo (**realm**) è un dominio di rete nel quale gli indirizzi di rete sono univocamente assegnati ad entità in modo che i pacchetti possano arrivare ad esse. I regni d'indirizzi possono essere pubblici o privati.

Ricordo che gli indirizzi privati sono [RFC1918] definiti mediante le classi :

10/8

172.16/12

192.168/16

Definizione 1.2. Dati due regni d'indirizzi $\mathbf{R_1}$ ed $\mathbf{R_2}$ si definisce una funzione di traduzione $\mathbf{F_{nat}}$ così fatta:

Date le porte **P1**, **P2**, per ogni indirizzo $\mathbf{IND_1}$ appartenente ad $\mathbf{R_1}$ esiste un $\mathbf{IND_2}$ appartenente a $\mathbf{R_2}$ tale che le seguenti condizioni siano verificate:

- $\mathbf{F_{nat}\ (IND_1,P1) = (IND_2\ ,P2)\ .}$
- $\mathbf{F_{nat}^{-1}\ (IND_2,P2) = (IND_1,P1).}$

• Nel caso le porte P1 e P2 non siano significative si assume P1 = **undef** e P2 = **undef**. Per i pacchetti diversi da TCP e UDP le porte non sono significative.

• Sia trasparente agli end system (**transparent routing**): l'esito della traduzione non deve propagare informazione su un regno d'indirizzo ad un sistema che non vi appartiene.

Che cosa è il NAT (Network Address Translation)? (2)

La funzione di traduzione $\mathbf{F_{nat}}$ viene implementata da una periferica di NAT (tipicamente router) ed è basata sul concetto di sessione.

Definizione 1.3. Si definisce una sessione come l'insieme di traffico a cui è applicata la medesima $\mathbf{F_{nat}}$. Ogni sessione ha una direzione e possono esservi i seguenti tipi:

- **Sessioni TCP/UDP**, univocamente identificate con la **tupla T = (indirizzo sorgente, porta sorgente, indirizzo destinatario, porta destinazione).**
- **Sessioni ICMP**, univocamente identificate con le **tupla I = (indirizzo sorgente, tipo ICMP, indirizzo destinatario)**.
- **Sessione di altri protocolli,** sono univocamente identificate con la **tupla O = (indirizzo sorgente, indirizzo destinazione, protocollo IP)**.

Le sessioni sono uniche e la traduzioni sono basate sulle sessioni applicando per una stessa sessione sia $\mathbf{F_{nat}}$ per i pacchetti in uscita che $\mathbf{F_{nat}^{-1}}$ per quelli in entrata. La direzione delle sessioni viene identificata dalla direzione del primo pacchetto della sessione e il NAT ne tiene traccia nella sua tabella.

Classificazione di NAT

NAT Statico: In questo caso la funzione di traduzione è 1:1. Per ogni IND_1 esite un unico IND_2 per ogni possibile sessione.

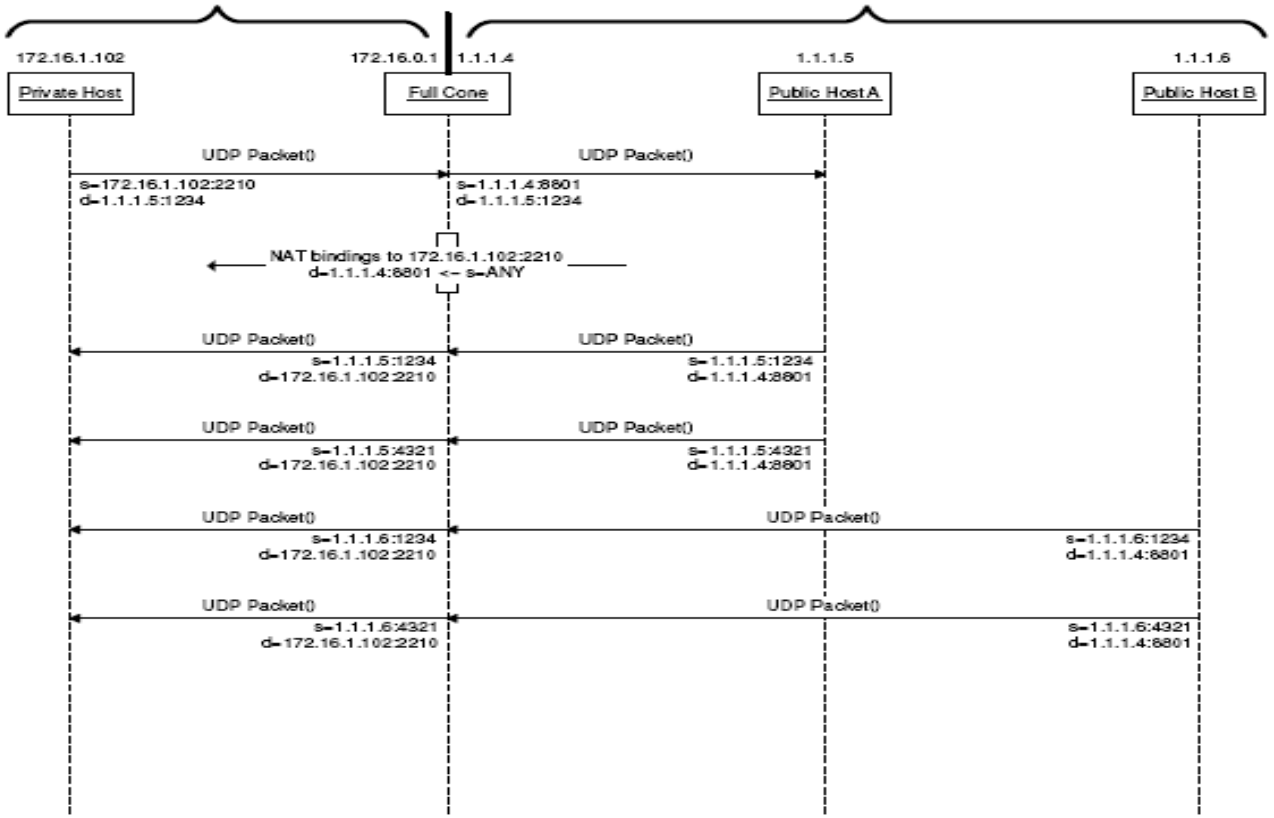
NAT Dinamico: L'assegnamento $IND_2 = F_{nat}(IND_1)$ viene determinato dalla periferica di NAT. Quando l'ultima sessione che usa questo assegnamento è terminata IND_2 può essere nuovamente riassegnato e la relativa entrata nella tabella di NAT liberata.

Quando noi parleremo di NAT intenderemo il NAPT (Network Address Port Traslation): traduco anche le porte UDP/TCP e gli identificatori ICMP.

Considieriamo un NAPT e classifichiamolo secondo le seguenti tipologie:

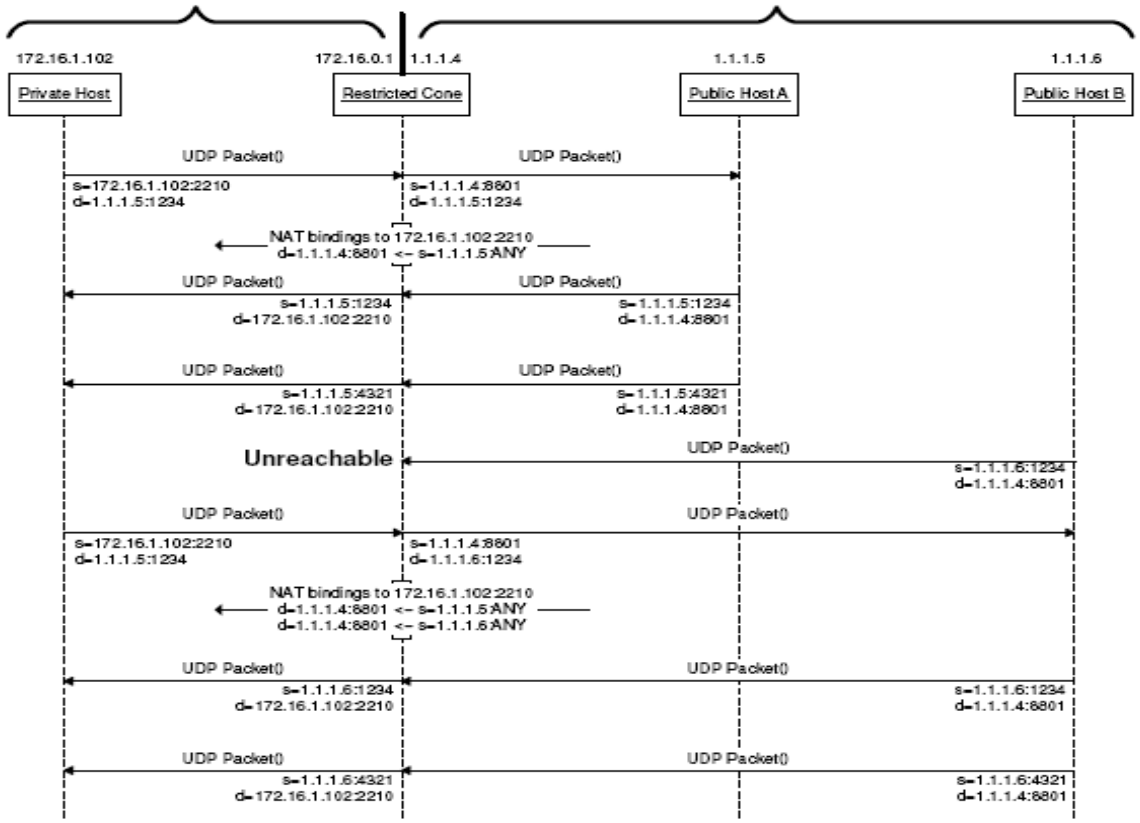
- NAT a cono pieno (Full Cone NAT).
- NAT a cono ristretto (Restricted Cone NAT).
- NAT con porte a cono ristretto (Port Restricted Cone NAT).
- NAT Simmetrico (Symmetric NAT).

NAT a cono pieno.



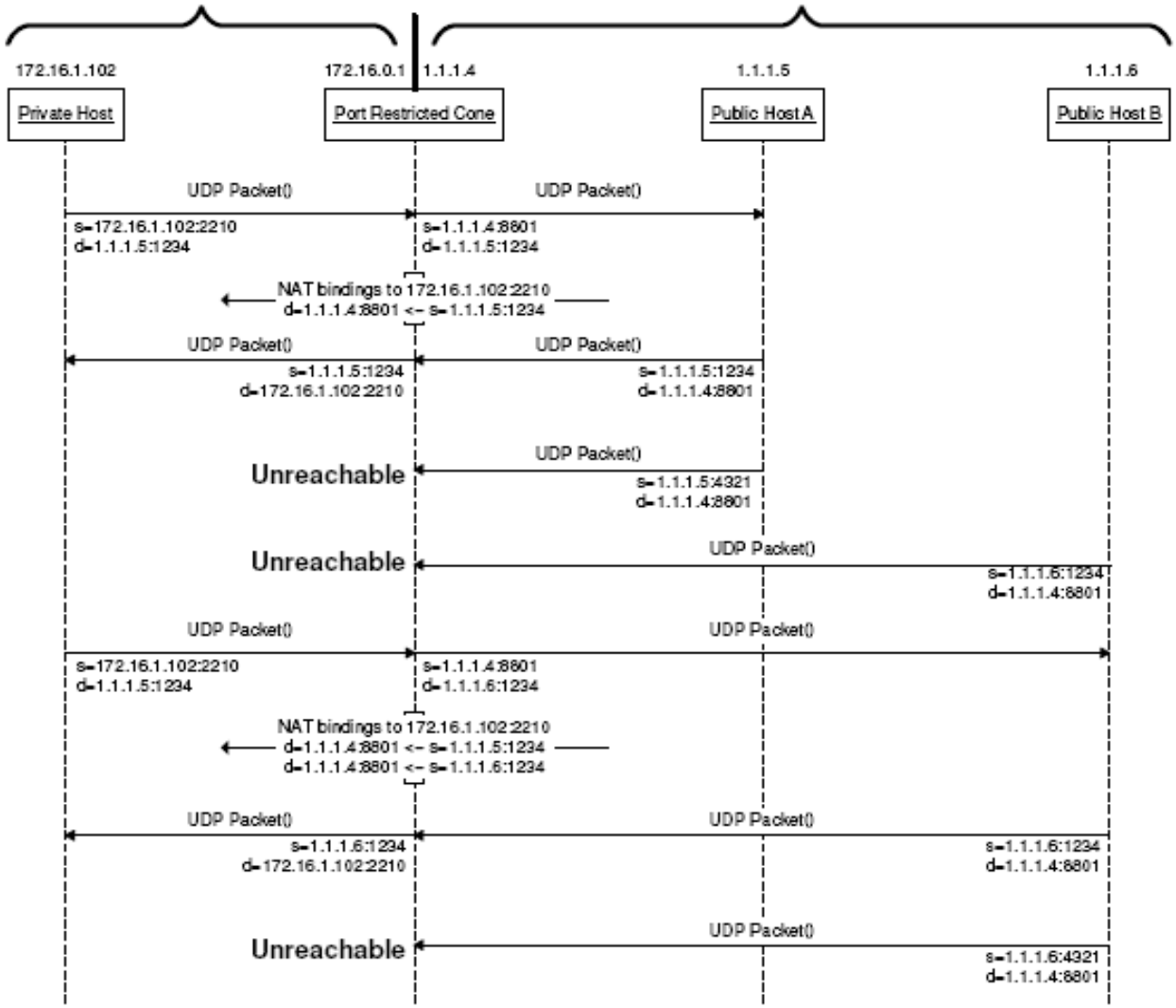
Tutte le richieste dallo stessa coppia (IND_1,P_1) sono mappate alla stessa coppia (IND_2,P_2). Ogni host esterno può mandare un pacchetto ad un host interno impostando l'indirizzo alla (IND_2,P_2).

NAT a cono ristretto.

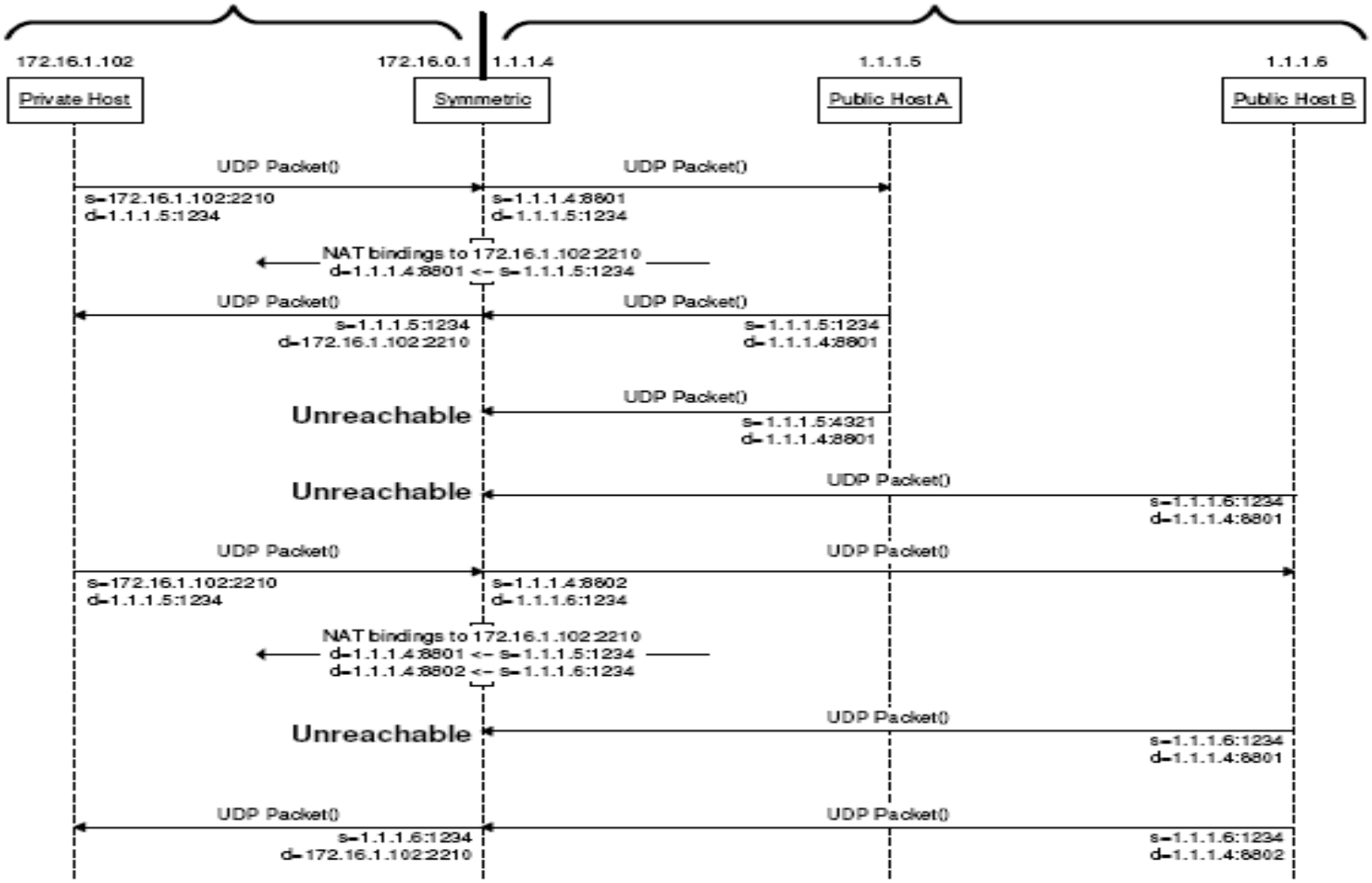


A differenza del cono pieno: un host esterno con indirizzo IND_2, può mandare un pacchetto ad un host interno se e solo se l'host interno ha mandato precedentemente un pacchetto ad IND_2

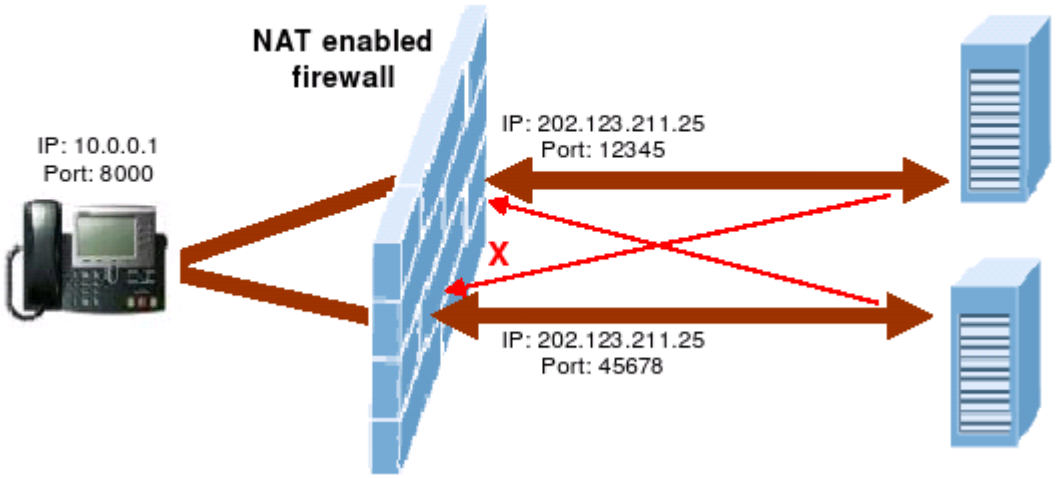
NAT con porte a cono ristretto.



NAT Simmetrico.



NAT Simmetrico.



NAT: Vantaggi e Svantaggi

Vantaggi:

•Permette di risparmiare indirizzi IP. (Motivo per cui è stato introdotto)

•Quando l'assegnamento viene fatto in maniera dinamica il NAT rafforza la security, in quanto non consente ad un host apparetenente ad un realm esterno di accedere alla rete privata.

Svantaggi:

• Viola il principio "***end to end argument***".

End to End Argument.*"In un sistema di comunicazione la funzione può essere completamente e correttamente implementata solo con la conoscenza e l'aiuto dell'applicazione che sta ai capi del sistema di comunicazione. Quindi, fornire tale funzione come una caratteristica stessa del sistema di comunicazione non è fattibile.(Talvolta viene implementata una versione incompleta della funzione per motivi di prestazioni)."*

Diversi protocolli Internet non funzionano con il NAT:

per esempio FTP,SDP,SIP,H.323,Kerberos,RTSP,ecc.

NAT e Voice Over IP: Problema

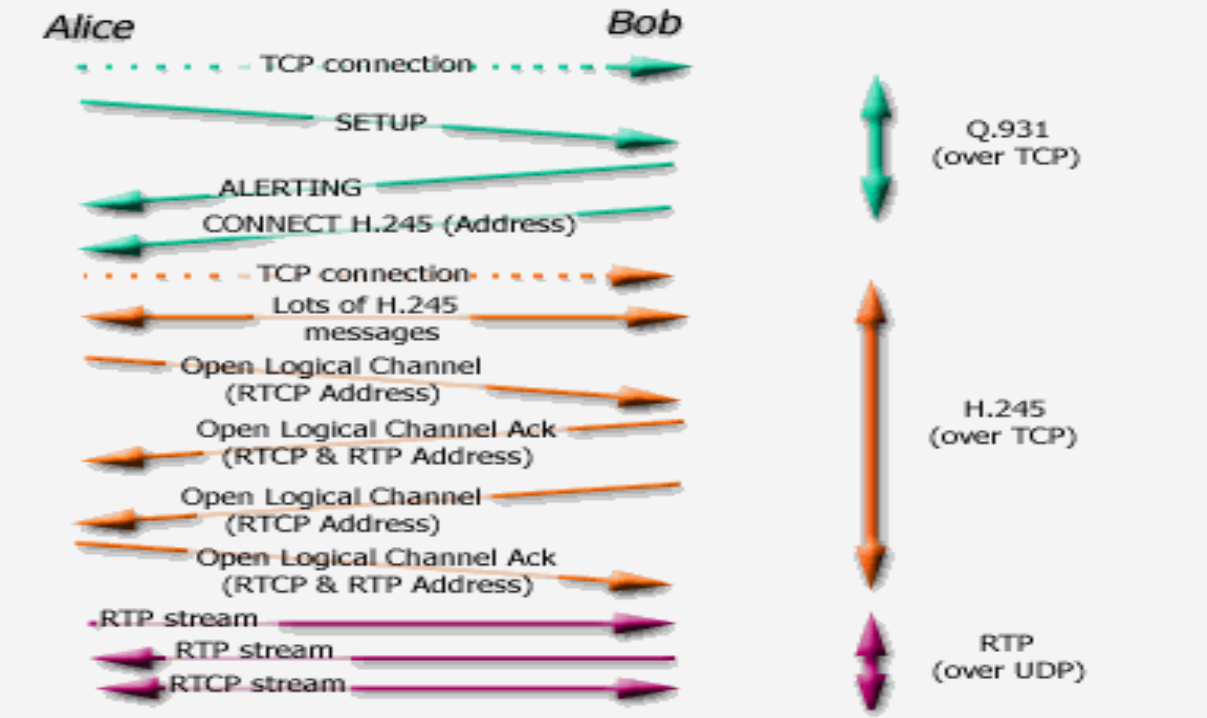
Problema:

I protocolli Voice Over IP (SDP,H.323,SIP) senza supporto da parte della periferica non funzionano attraverso il NAT.

Caratteristiche che il NAT viola:

- Indirizzi specifici di un realm o informazioni sulle porte nel payload. Il NAT non le traduce. Questo è fa in modo che per esempio i pacchetti SDP abbiano indirizzi non validi.
- Applicazioni che hanno incorporata una sessione di controllo, come in SIP, H.323 e RTSP. Queste applicazioni non funzionano perchè la sessione di controllo viene usata per scambiare indirizzi e porte al fine di stabilire la successiva sessione trasferimento dati. Il NAT in genere non conosce il protocollo e il concetto di "sessione" per quel protocollo. I motivi di fallimento sono essenzialmente due:
  1. Le informazioni di indirizzamento nel payload sono specifici di un realm e non sono più validi quando il pacchetto lascia tale realm.
  2. La sessione di controllo permette alle sessioni di trasferimento dati di aver origine nella direzione che il NAT non permette. N.B: Il trasferimento via RTP può non essere **simmetrico**.

Connection Setup in H.323 (1) :Chiamata diretta tra due sistemi non registrati ad un Gatekeeper.



•Alice crea una connessione TCP verso la porta 1720 di Bob; questa è una porta cosiddetta "ben nota" per l'H.323, ove i client si mettono in ascolto. Attraverso questa connessione i due endpoint si scambiano un indirizzo TCP, e cioè un numero di porta, dinamico (varia chiamata per chiamata, casualmente) e maggiore di 1024, da usare per la connessione H.245. Viene effettuata la connessione sulle porte negoziate attraverso il flusso H.225.0. Attraverso lo scambio dei messaggi H.245 vengono determinate le capacità degli endpoint, i codec audio e video, e viene stabilito chi dei due è Master e chi Slave.
•Alice manda il messaggio Open Logical Channel, che contiene il suo indirizzo RTCP.
•Bob risponde con il messaggio OpenLogical Channel Acknowledge, che contiene il suo indirizzo RTCP e quello RTP. Gli indirizzi in questione sono semplicemente dei numeri di porte UDP: i numeri delle porte RTCP e RTP sono consecutivi, con RTP numero pari e RTCP uguale al numero dispari immediatamente superiore. Dato che i canali sono unidirezionali, la procedura viene ripetuta per entrambe le direzioni, e per ogni tipo di media che si vuole trasmettere (c'è uno scambio OLC-OLCACK per il video, uno per l'audio, etc.)
•A questo punto sia Alice sia Bob conoscono le porte UDP alle quali mandare i pacchetti, e i flussi audio e video possono partire.

Connection Setup in H.323 (2):Chiamata tra due endpoint registrati allo stesso gatekeeper

•Alice manda un messaggio di Admission Request al gatekeeper sulla porta 1719: all'interno del messaggio è specificato l'alias H.323 del destinatario della chiamata. Il gatekeeper risolve l'indirizzo del chiamato, e restituisce, nel messaggio Admission Confirm, un indirizzo di segnalazione di chiamata H.225.0: nel caso presente (modalità direct) tale indirizzo è quello dell'endpoint chiamato.
•Alice adesso conosce l'indirizzo al quale mandare il setup a Bob: crea una connessione sulla porta ben nota 1720 e ripete la procedura vista nel caso precedente, negoziando la porta TCP sulla quale creare la connessione H.245
•Inizia lo scambio di messaggi H.245 sulla connessione TCP creata: Terminal Capability Set, Master Slave Determination, informazioni sui canali logici.
•Entrambi gli endpoint conoscono le porte UDP sulle quali mandare i flussi audio e video, e inizia la chiamata vera e propria

NAT e Voice Over IP(1): Esempio H.323, H.245

Chiamata diretta tra due sistemi non registrati ad un un gatekeeper

Supponiamo che Alice, host ad indirizzo privato, conosca l'indirizzo H.225 di Bob, host ad indirizzo pubblico.

- Alice manda il messaggio di setup a Bob. Il pacchetto contenente il messaggio è destinato a un host che non si trova sulla stessa LAN di Alice e viene inviato al gateway verso la rete Internet, e instradato verso Bob.
- Bob riceve il pacchetto, e all'interno trova l'indirizzo al quale inviare il messaggio di risposta, vale a dire quello di connect. L'indirizzo è quello H.225 di Alice, composto da un indirizzo di rete privata e da un numero di porta TCP. Il pacchetto viene inviato da Bob al proprio router di default, il quale non instrada i pacchetti diretti all'indirizzo privato. La chiamata non può avere luogo.

NAT e Voice Over IP(2): Esempio H.323, H.245

Nota Bene: Il Gatekeeper deve conoscere entrambe i realm.
Vediamo come si evolve la chiamata:

- Alice manda un messaggio di Admission Request al gatekeeper sulla porta 1719: all'interno del messaggio è specificato l'alias H.323 del destinatario della chiamata. Il gatekeeper risolve l'indirizzo del chiamato, e restituisce, nel messaggio Admission Confirm, l'indirizzo di Bob.
- Alice manda il messaggio di Setup, ma il pacchetto contenente tale messaggio ha nel campo destinazione un indirizzo privato, e viene scartato dal primo router che incontra. Chiamata fallita!

NAT e Voice Over IP(3): Esempio H.323, H.245

Senza esaminare nel dettaglio tutti i casi possibili, diremo che nel caso H.225 routed la segnalazione è inoltrata dal gatekeeper, e quindi i messaggi di Setup, Alerting e Connect arrivano a destinazione. In tali messaggi viene specificato un indirizzo H.245, formato da un numero di porta e da un indirizzo di rete privato, il che provoca il fallimento della chiamata per i motivi ampiamente specificati.

Nel caso H.225 e H.245 routed, anche il controllo di chiamata passa per il gatekeeper, e la chiamata viene instaurata; i flussi RTP e RTCP, d'altro canto, sono trasportati da pacchetti che hanno nella loro intestazione un indirizzo di destinazione privato, e non vengono instradati. La chiamata non è fallita, ma non è neanche molto utile, dato che, nel migliore dei casi, solo uno dei due endpoint riceverà l'audio inviato dall'altro.

NAT e Voice Over IP(4): Esempio SIP/SDP

Problemi:

- Indirizzi privati presenti nel messaggio SIP nei campi Via e Contact
- Indirizzi privati presenti nel messaggio SDP: Questi fanno si che la connessione media (RTP) non abbia luogo perchè non sono indirizzabili da un realm pubblico.
- SIP su UDP puo' creare problemi dovuti al fatto che il NAT ha un arco temporale dopo il quale libera l'assegnamento
- RTP deve essere simmetrico per passare il NAT: una sessione RTP deve avere origine dalla stessa porta dove l'altra sessione ha avuto luogo. (Attributo direction in SDP).
- Nel caso volessimo interagire sul messaggio mediante un Application Level Gateway il messaggio SIP non può usare TCP e TLS come da specifiche: non può essere cifrato.

NAT e Voice Over IP(6): Il problema con RTP (RTP simmetrico)

- RTP è unidirezionale
  - A indica la coppia (IP,porta) da cui ricevere da B
  - B indica la coppia (IP,porta) per ricevere da A
  - Se entrambi sono dietro il NAT, i dati RTP non fluiscono.
- Soluzione: rendere RTP un protocollo client-server:
  - A indica una coppia (IP,porta) per ricevere da B
  - B manda ad A, A manda indietro la risposta a B usando la coppia (porta,sorgente) di RTP
  - Di fatto quello di cui abbiamo bisogno è solo l'indirizzo di UNO dei partecipanti – il “server”.

Soluzioni

•**Soluzioni strutturali**: Il supporto del protocollo viene incluso all'interno della periferica di NAT oppure il client apre un "buco nel NAT". Esse sono Application Level Gateway, MIDCOM, Universal Plug and Play. Richiedono il supporto da parte della periferica di NAT.

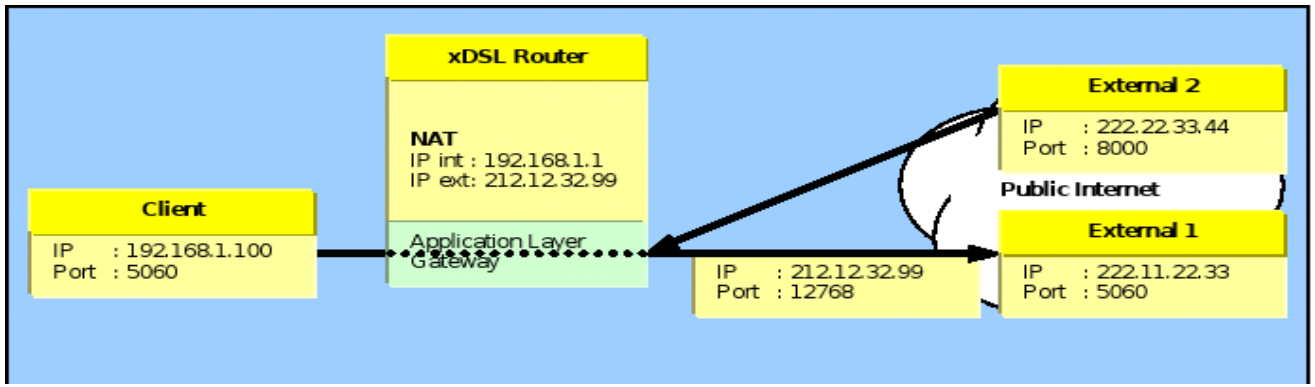
•**Soluzioni non strutturali**: Il client determina il tipo di NAT mediante STUN e usa un host esterno con indirizzi pubblici, che diventa un server STUN/TURN ed grazie a questo server riesce a fare UDP/TCP Hole Punching.

Applicazioni esistenti: NUTSS, HandyTone, Skype e tutte le applicazioni P2P.

Il motivo percui Skype ha avuto successo è proprio questo: funziona ovunque.

Skype usa una variante di STUN per determinare il tipo di NAT e il p2p per convogliare il traffico voice, facendo uso anche del UDP Hole Punching.

Soluzioni strutturali: Application Level Gateway

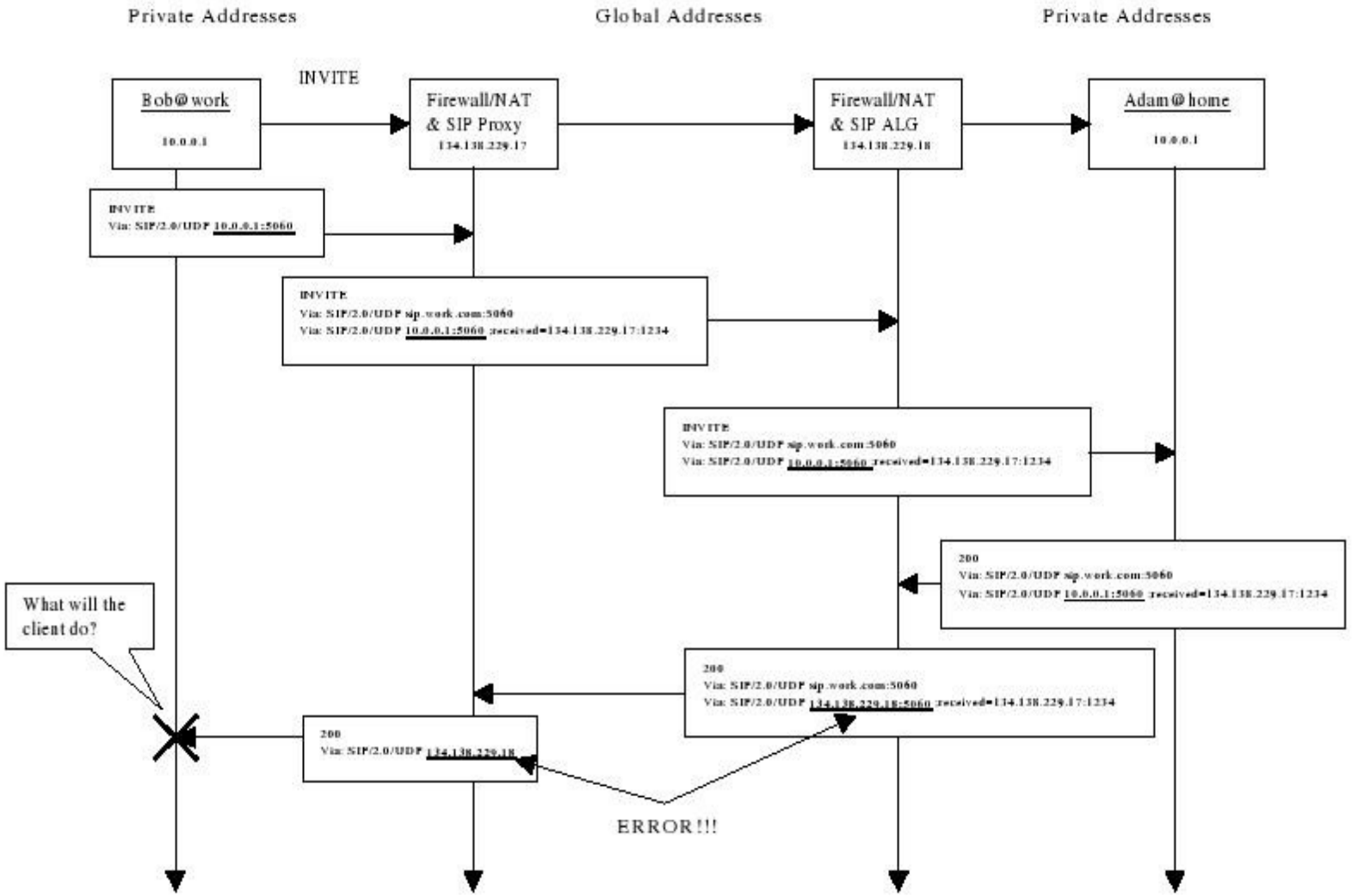


Gli ALG (gateway al livello di strato di applicazione) fungono da collegamento tra due reti; sono delle entità capaci di "guardare dentro" i messaggi di protocollo e di permettere di passare solo a quei messaggi conformi a certe politiche predefinite. A volte ci si riferisce erroneamente agli ALG come a dei proxy. C'è una differenza: gli ALG sono trasparenti alle entità che scambiano flussi multimediali, mentre i proxies non lo sono.

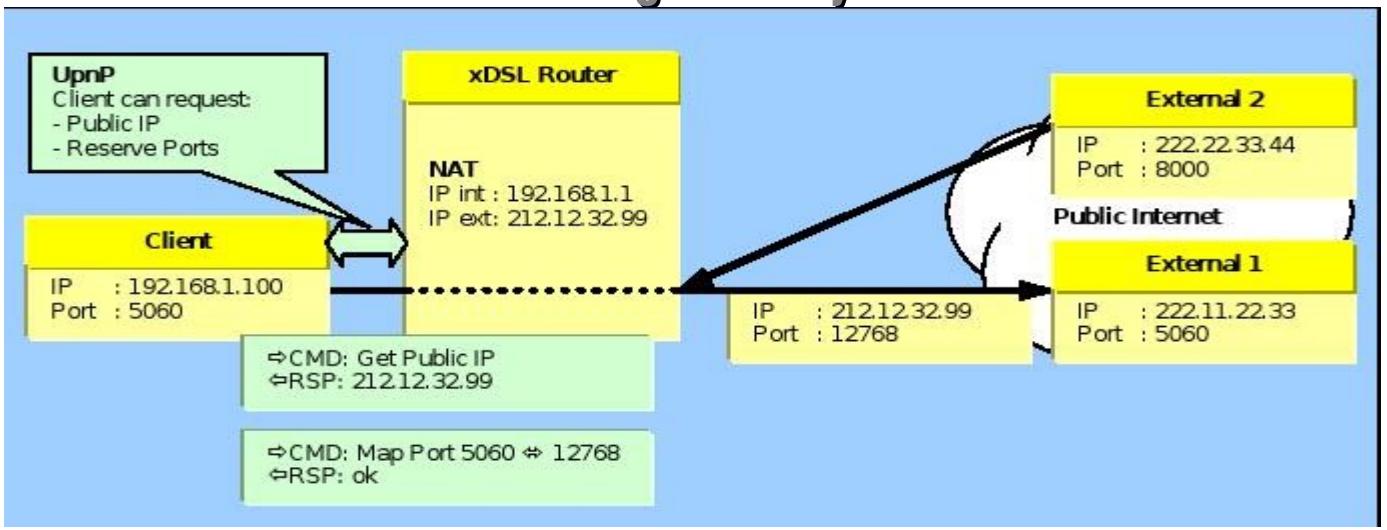
Soluzioni strutturali: Application Level Gateway (2)

Vantaggi:
Gestisce il protocollo
Svantaggi:
1) Il protocollo SIP non può essere cifrato perchè l'ALG non funzionerebbe: non posso fare SIP over TLS.
2) Non scala sul numero di connessioni
3) Cambio periferica di NAT, deve saper parlare tutti i dialetti del protocollo.
In alcuni scenari, non si puo dover aggiornare il NAT dove si ha bisogno di SIP o H.323.
Deve tenere la sessione SIP, che è diversa dalla sessione del NAT.

Soluzioni stutturali : ALG



Soluzioni strutturali: Universal Plug and Play



Svantaggi:

- Non funziona con NAT in cascata
- Security, un qualsiasi programma può aprire il NAT.
- La maggior parte dei NAT devices non lo supporta

Vantaggi:

Funziona con tutti e 4 i tipi di NAT.

Soluzioni non strutturali: Determinare il NAT con STUN

Le soluzioni strutturali implicano l'introduzione di un nuovo elemento nella topologia di rete, che può non essere accettabile quando non si ha il controllo dell'infrastuttura di rete. Per garantire una maggior diffusione, in sviluppatore di applicazioni Voice over IP può voler a questo punto:

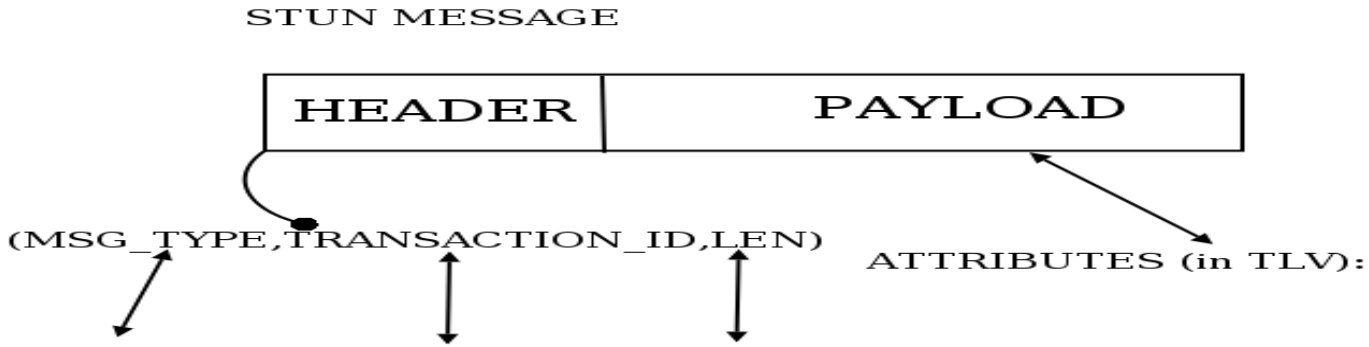
a)Capire il tipo di NAT mediante il protocollo STUN.

b)Applicare una serie di hacks per far si che sia possibile trasmettere il traffico di segnalazione e di multimediale attraverso il NAT questi sono: UDP/TCP hole punching, TURN, ICE.

c) Usare una infrastuttura peer-to-peer (p.e. CAN,DHT,ecc.) per applicare queste tecniche.

Vedremo ora queste tecniche.

Come funziona STUN : Tipi Messaggio



- BINDING REQUEST
- BINDING RESPONSE
- BINDING ERROR RESPONSE
- SHARED SECRET RESPONSE

serve per correlare le richieste alle risposte.

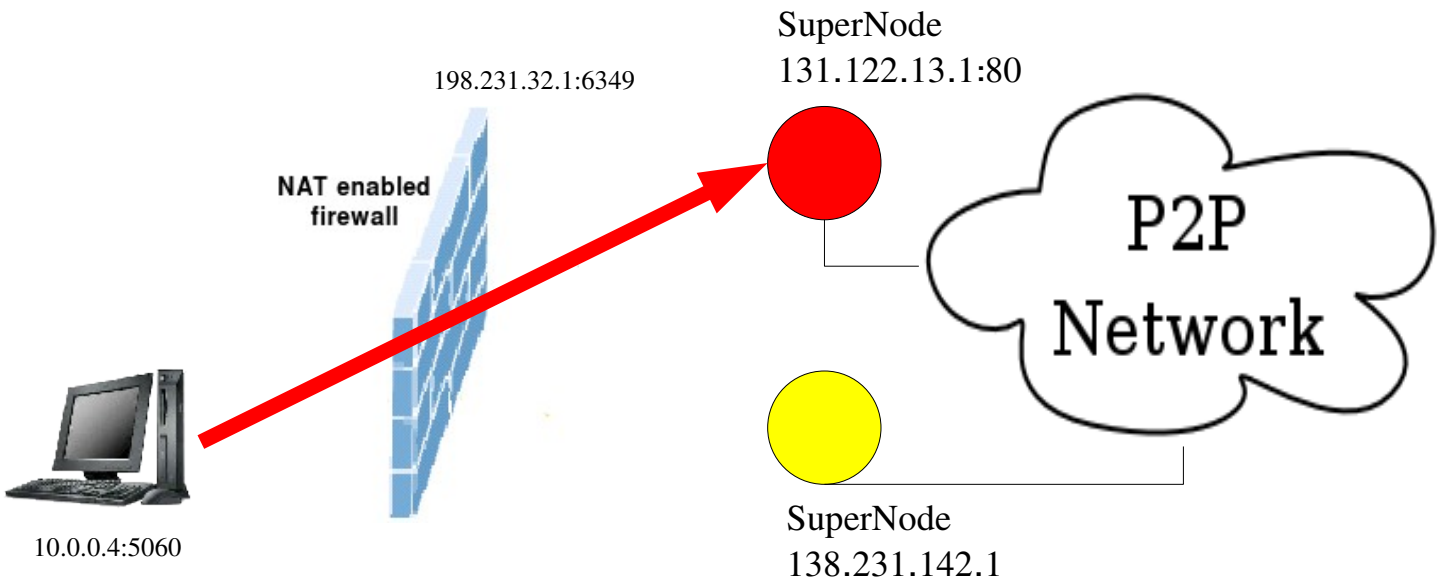
Lunghezza del Payload

(1)MAPPED-ADDRESS
(2)RESPONSE-ADDRESS
(3)CHANGE-REQUEST
(4)SOURCE-ADDRESS
(5)CHANGE-ADDRESS
(6)USERNAME
(7)PASSWORD
(8)MESSAGE-INTEGRITY
(9)ERROR-CODE
(10)UNKNOWN
(11)REFLECTED-FROM

(MAPPED-ADDRESS,LEN,(IP1,Port1)): IP, porta, visti dal server STUN
(RESPONSE-ADDRESS,LEN,(IP2,Port2)): IP, porta dove il server deve inoltrare le BINDING RESPONSE
(CHANGE-REQUEST,LEN,(IP3,Port3)); IP,porta che il server mette nelle BINDING RESPONSE
(CHANGE-ADDRESS,LEN,(IP4,Port4)): IP, porta presente nelle BINDING RESPONSE.
Questo attributo informa il client dell'(IP,porta) che sarebbe usato se il cliente richiedesse il comportamento "change IP", "change porta".
(SOURCE-ADDRESS,LEN,(IP5,Port5):IP,porta da dove è stata mandata la BINDING RESPONSE

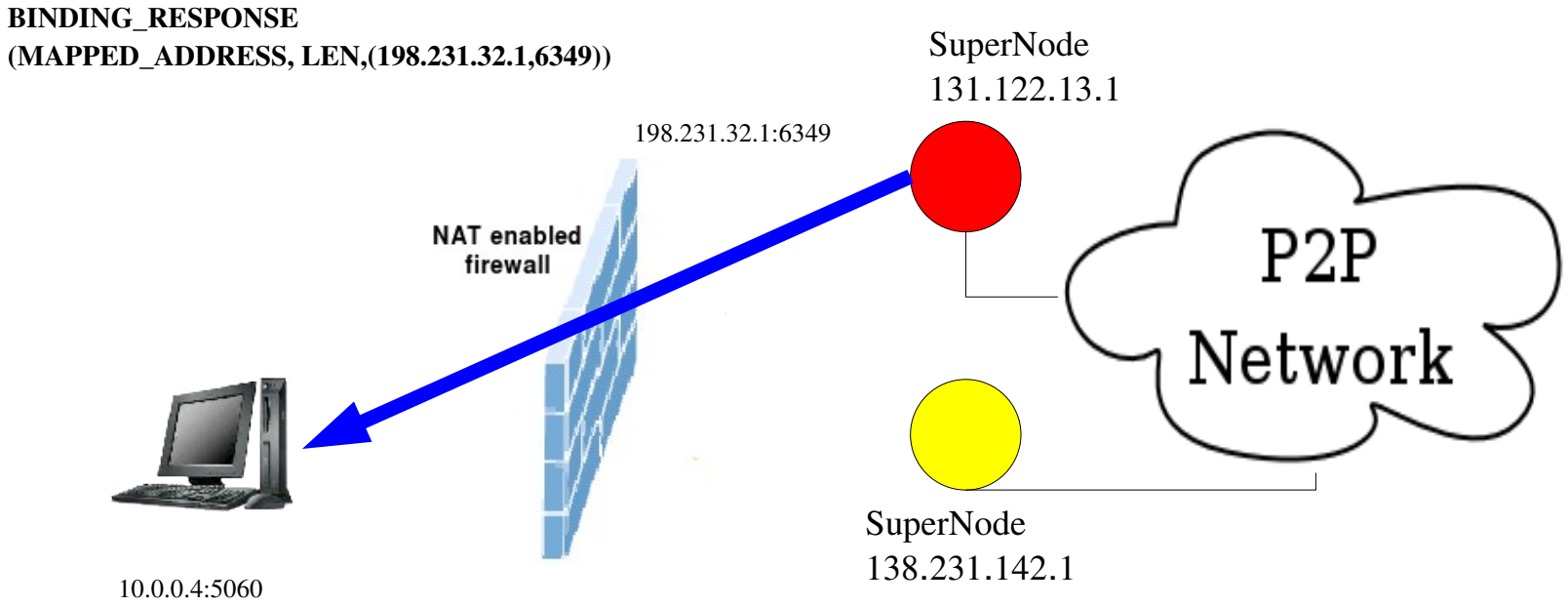
Come funziona STUN (TEST 1)

Dopo essermi autenticato al server STUN mediante uno scambio di chiave condivisa usando TCP/TLS



Effettuo una BINDING REQUEST R via UDP

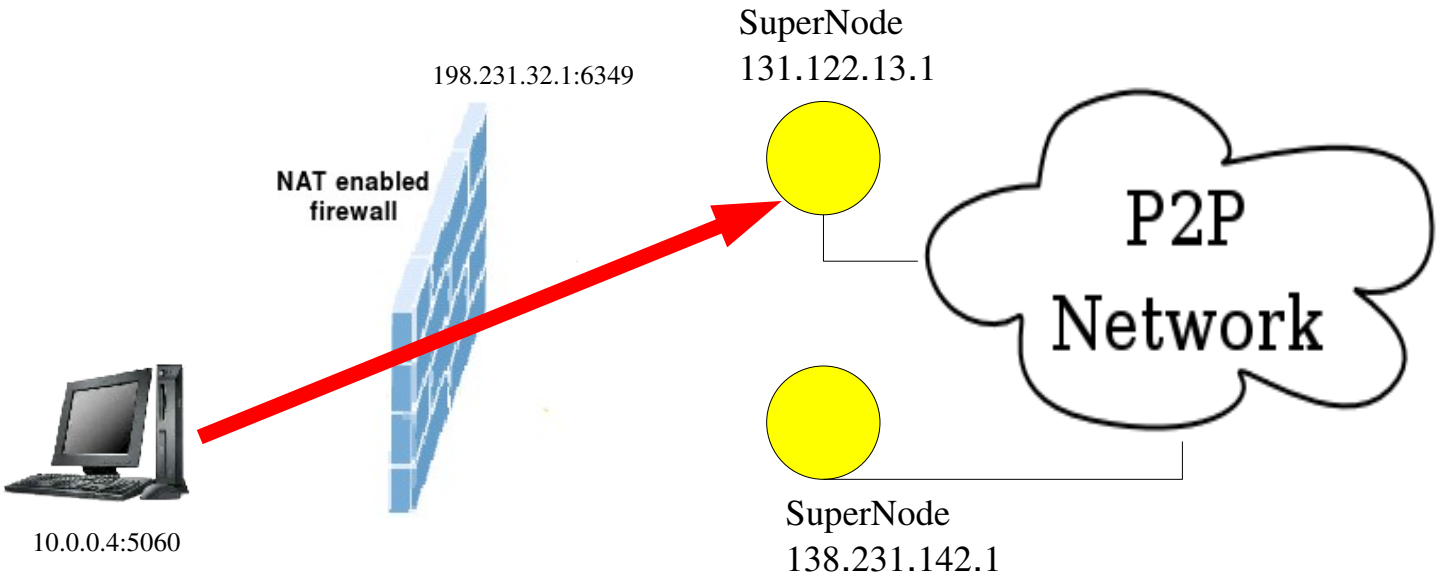
Come funziona STUN (TEST 1)



Se il peer non ottiene una BINDING_RESPONSE il traffico UDP viene bloccato dal FW: so che sono dietro a un FW che blocca gli UDP.
Se ottiene una risposta si esamina l'attributo MAPPED_ADDRESS, se l'IP e la porta corrispondono a quelli dell'host locale siamo in un ambiente senza NAT. Sono dietro il NAT e passo al TEST 2.

Come funziona STUN (TEST 2)

BINDING_REQUEST
(CHANGE-REQUEST, LEN,(138.231.142.1,80))

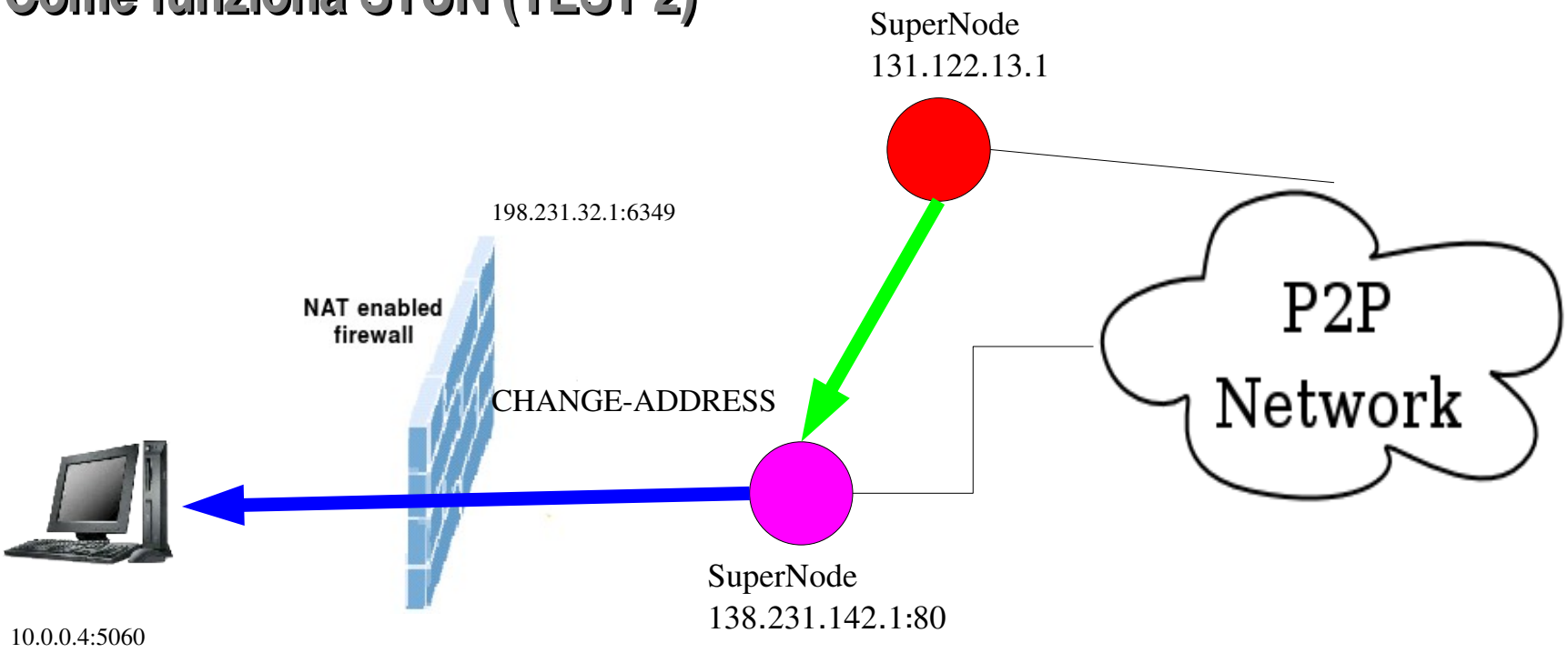


Effettuo una BINDING REQUEST R via UDP con attributo (CHANGE-REQUEST, IP_2,P_2) con

IP_2 = indirizzo con cui il server forgia la risposta

P_2 = porta con cui il server forgia la risposta

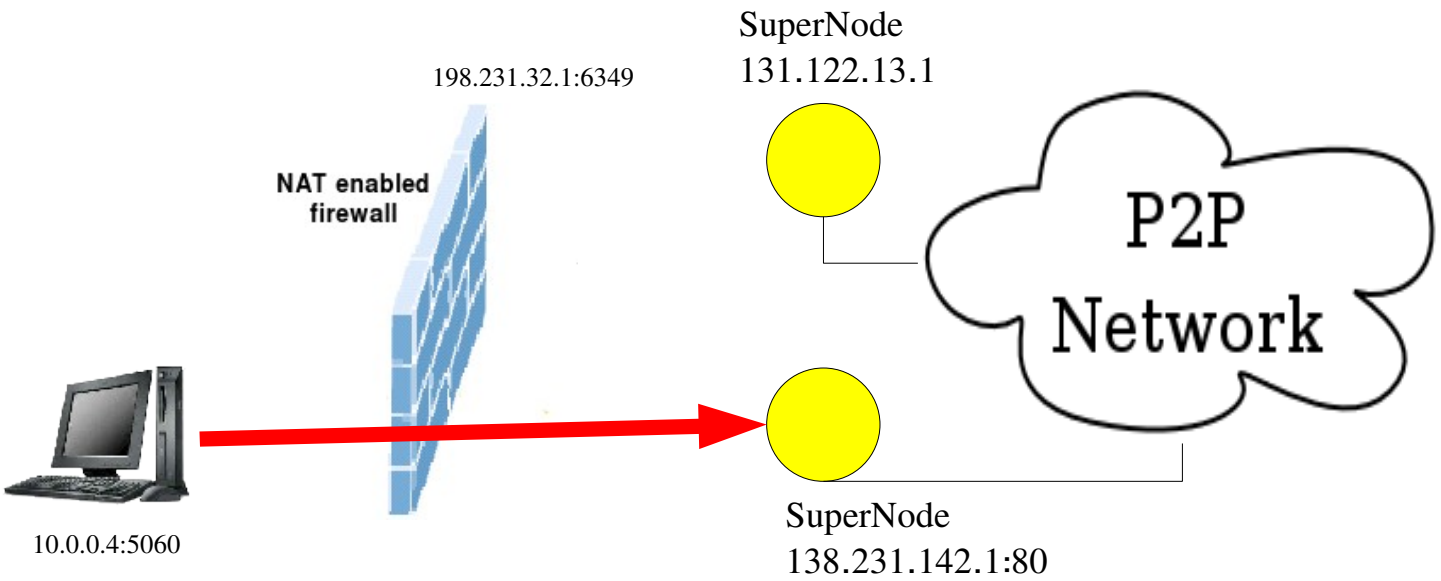
Come funziona STUN (TEST 2)



Se ottengo una risposta il client sa che ha un accesso aperto ad internet, oppure sono dietro un NAT a cono pieno.
Se invece non ottengo risposta sono dietro un firewall UDP simmetrico.

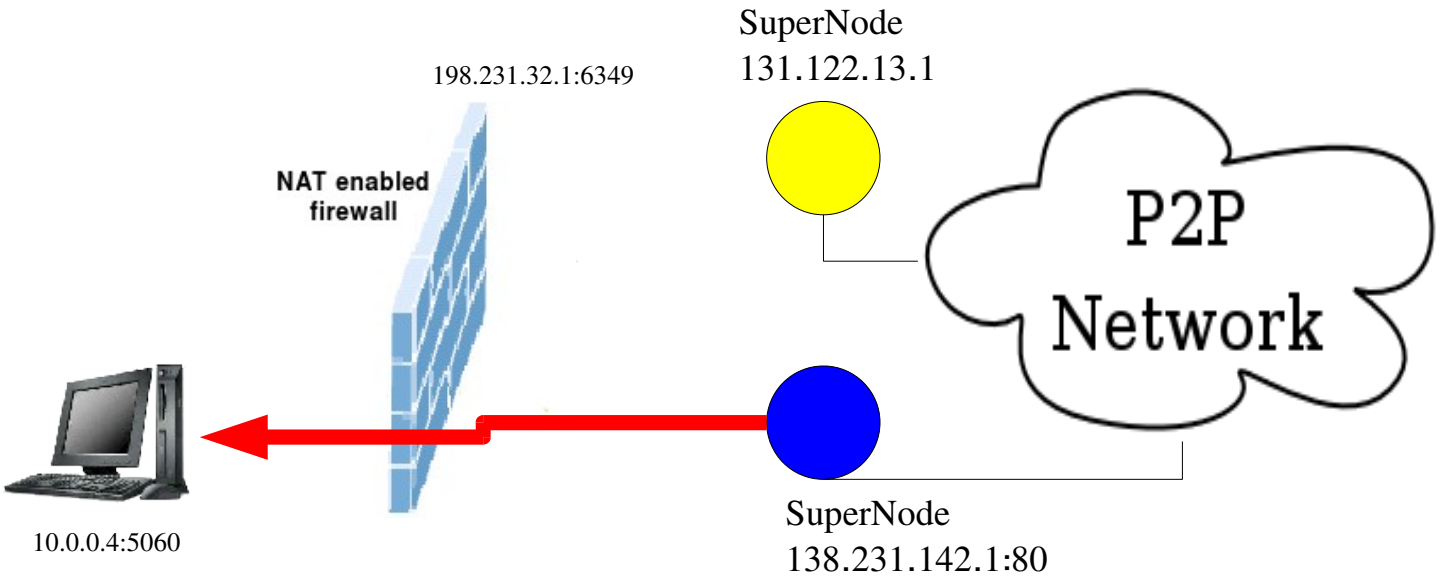
Se non ottengo rifaccio il TEST3 nel seguente modo

Come funziona STUN (TEST 3)



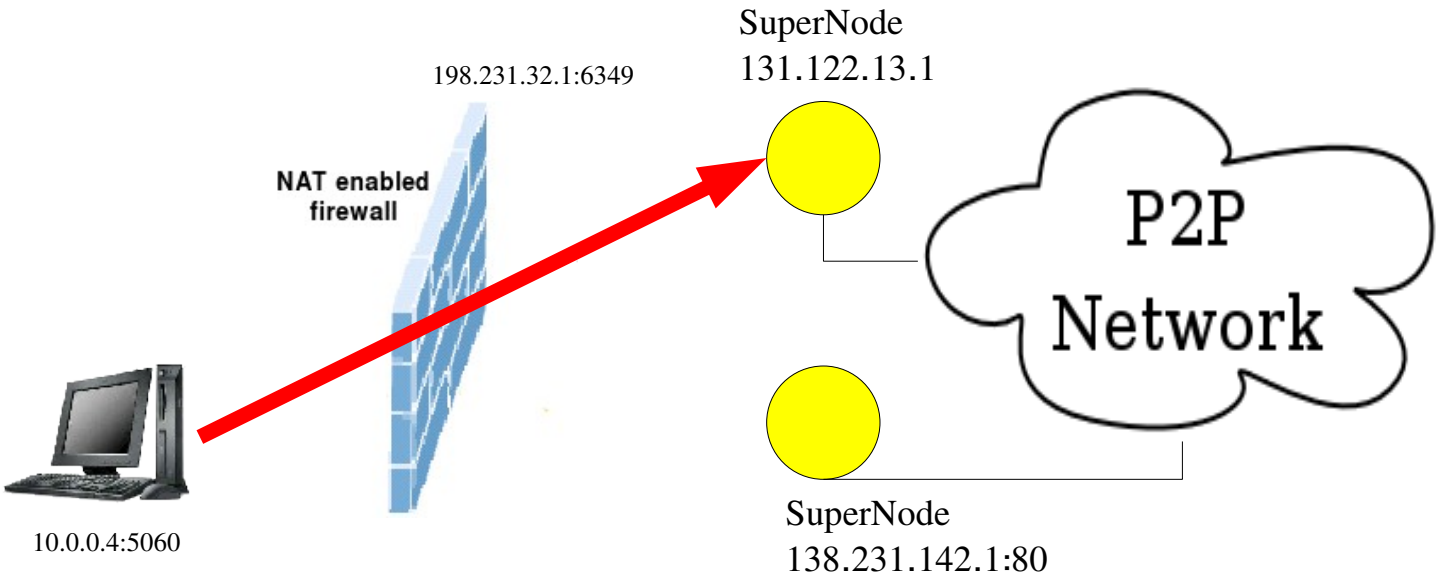
Effettuo una BINDING REQUEST R

Come funziona STUN (TEST 3)



Controllo l'attributo MAPPED_ADDRESS, se l'IP e la porta corrispondono a quelli di TEST1 siamo in un ambiente senza NAT. Sono i due indirizzi non sono gli stessi sono dietro un NAT simmetrico, se sono gli stessi sono dietro un NAT a cono ristretto o con porte a cono ristretto. Allora TEST 4.

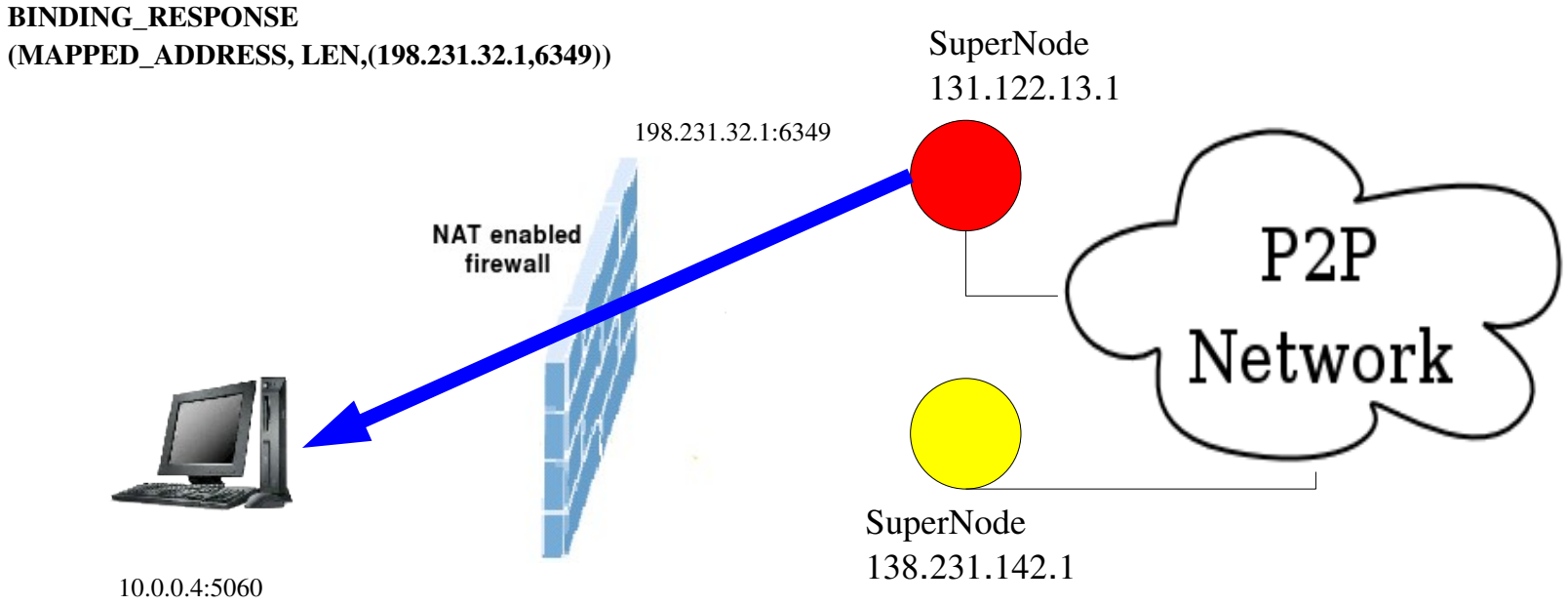
Come funziona STUN (TEST 4)



Effettuo una BINDING REQUEST R via UDP con attributo (CHANGE-REQUEST,NULL,P_2) con

P_2 = porta con cui il server forgia la risposta

Come funziona STUN (TEST 4)



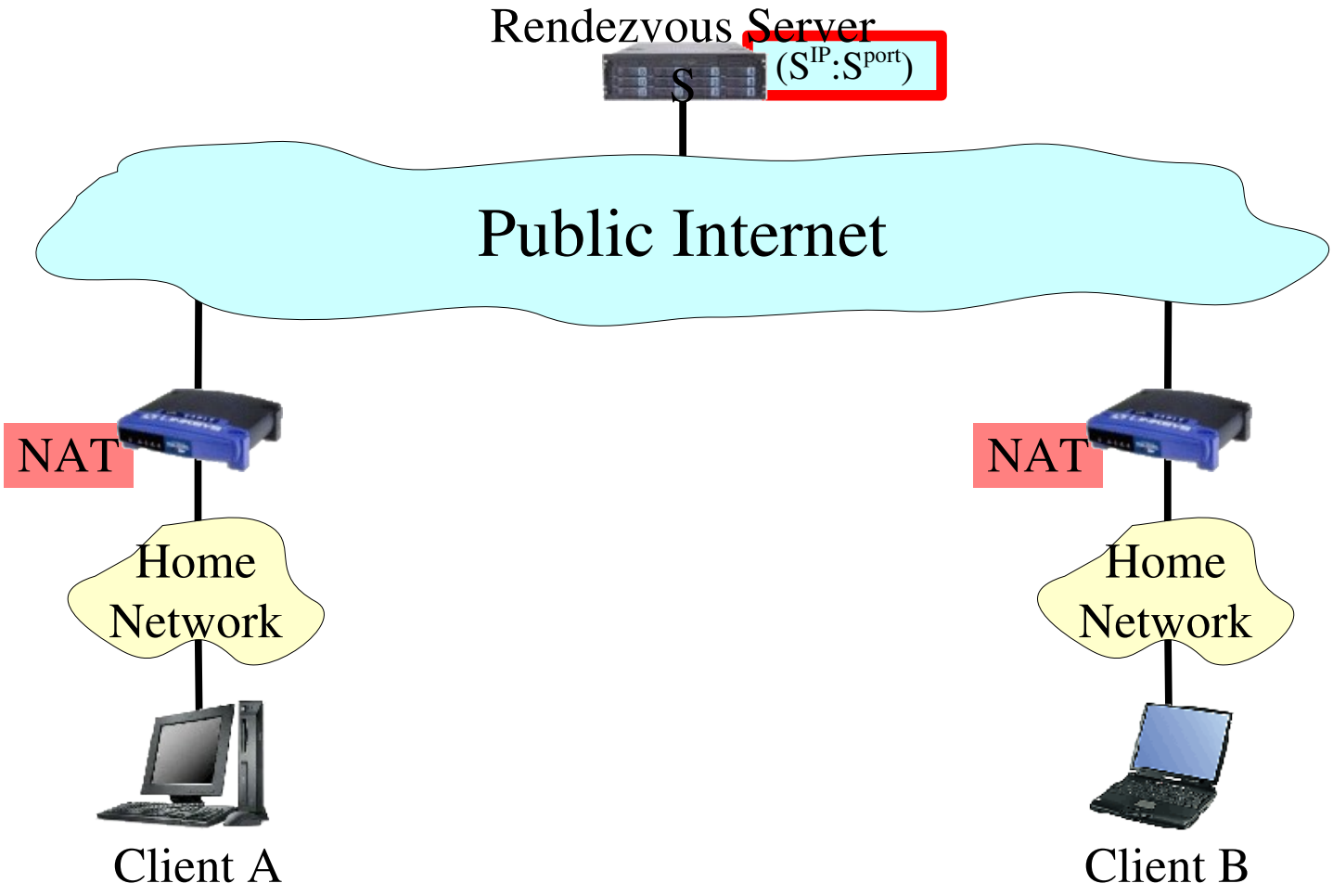
Se il peer non ottiene una BINDING_RESPONSE sono dietro un NAT con porte ristretto, altresi sono dietro un NAT ristretto

UDP Hole Punching

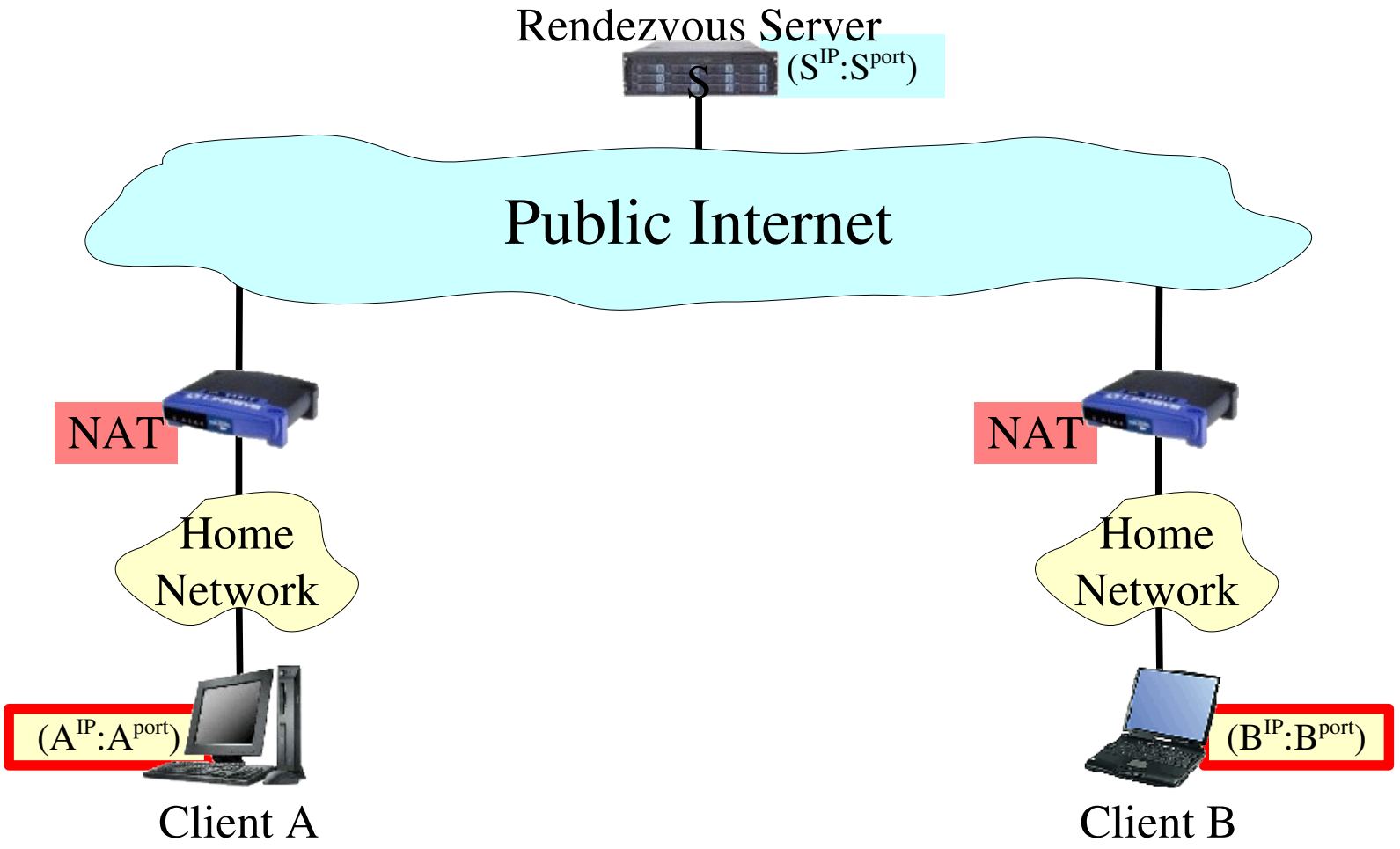
Assunzioni sul modello di utilizzo:

- I client VoIP si registrano a dei rendezvous peer per essere accessibili agli altri client.
- L'applicazione implementa la nozione di “identità”
  - Username, public key, etc.
- I rendezvous peer (o SuperNode) facilitano l'impostazione della chiamata man non partecipano nella risultante sessione P2P.

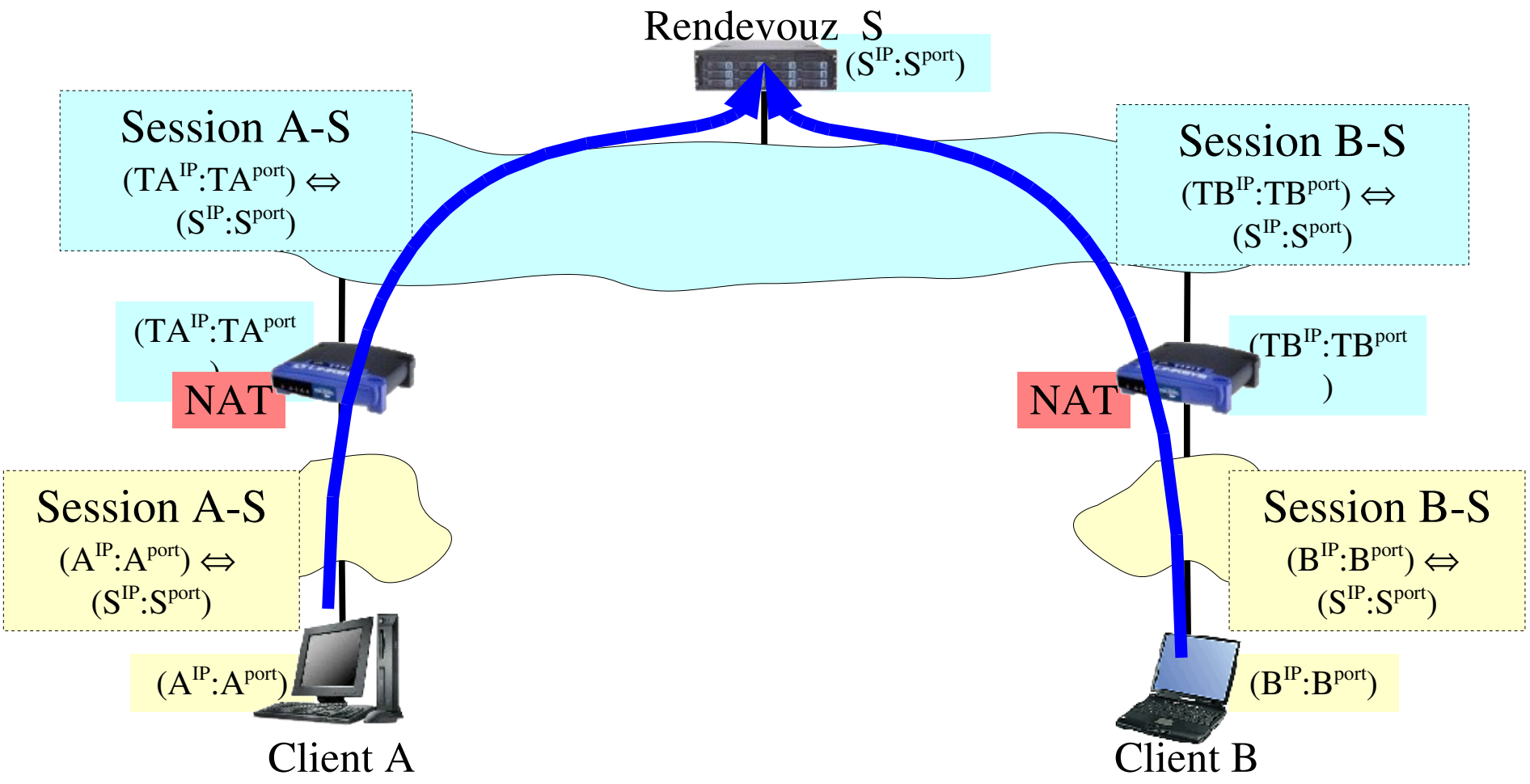
UDP Hole Punching



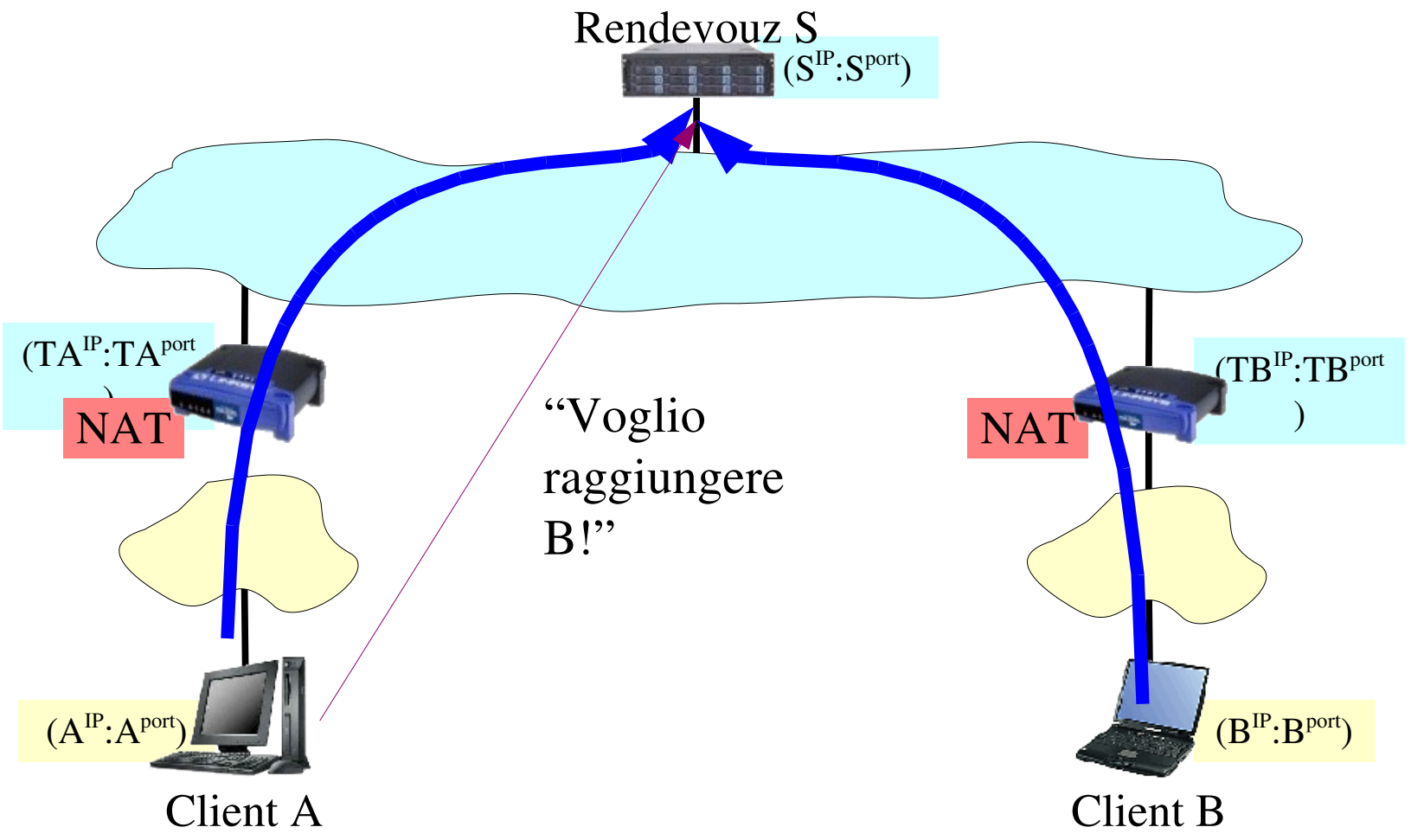
UDP Hole Punching



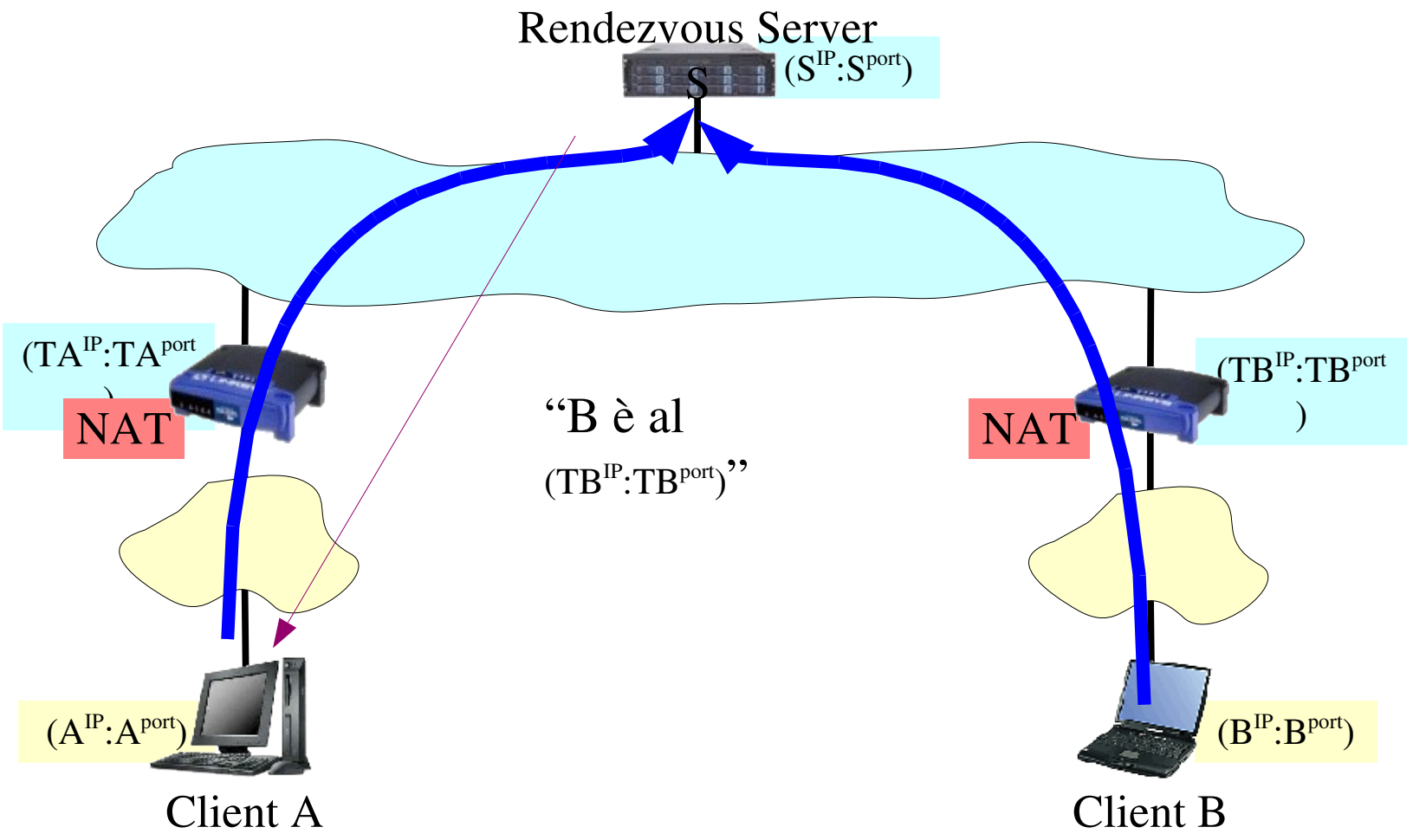
UDP Hole Punching



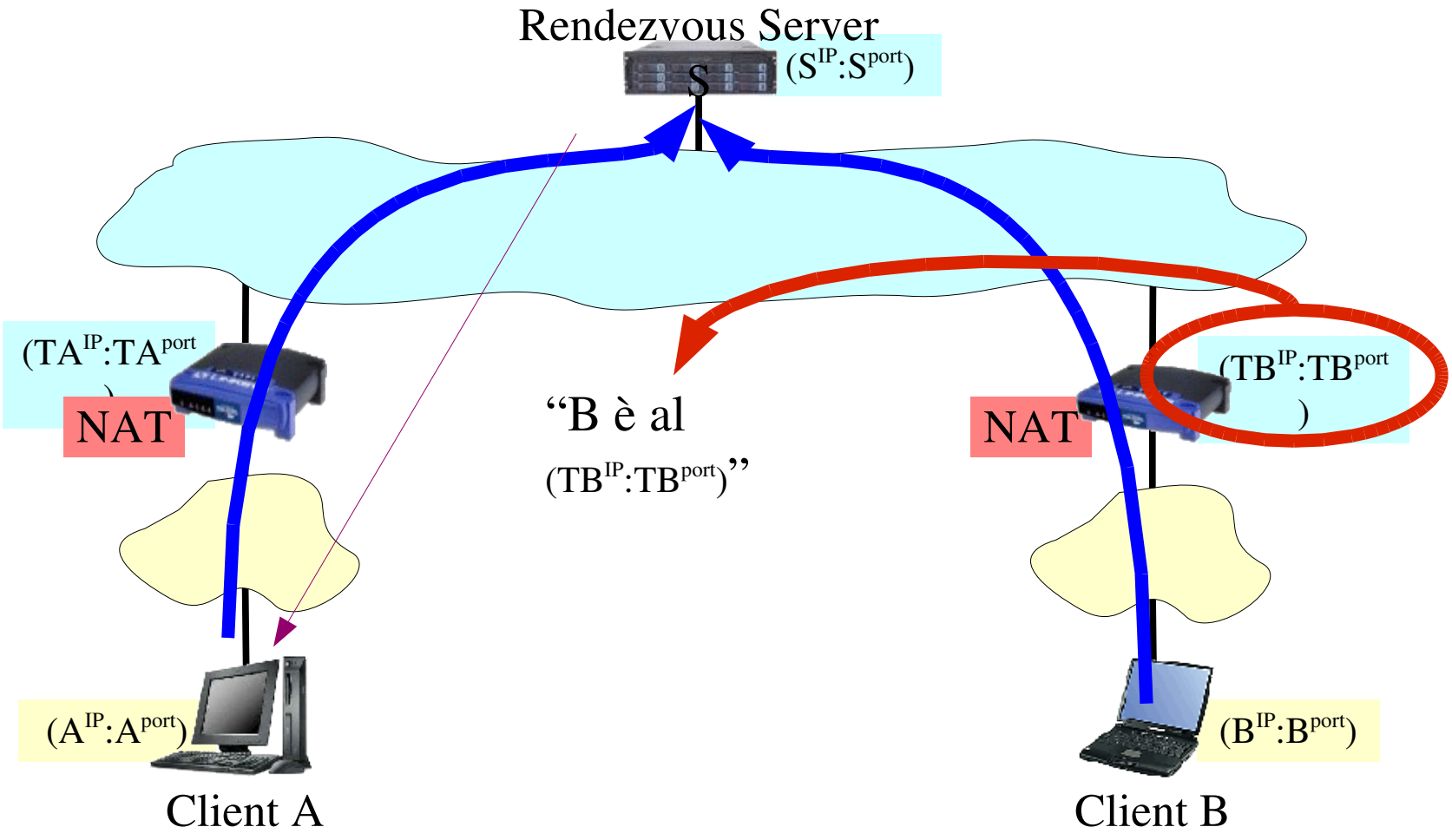
UDP Hole Punching



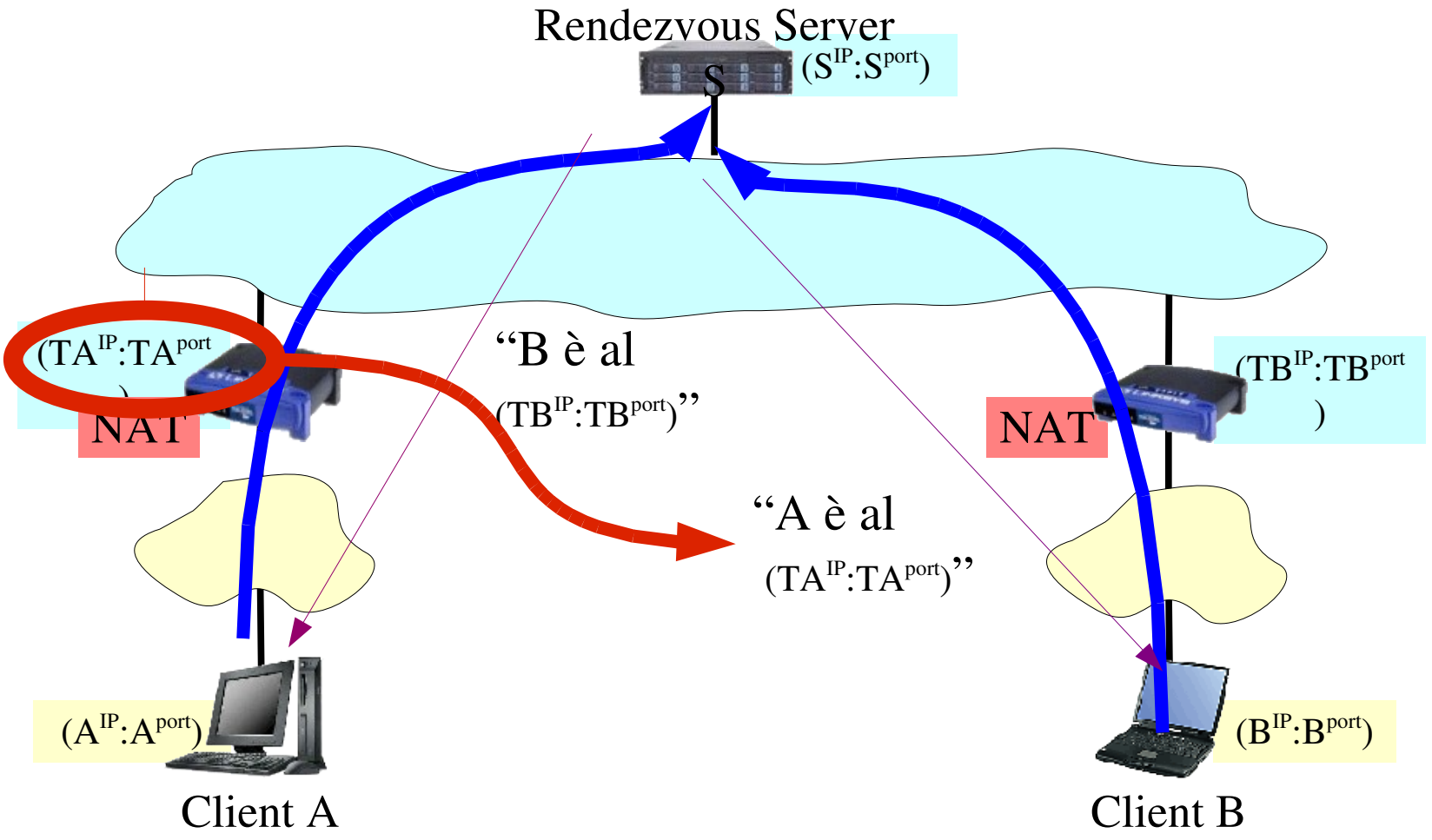
UDP Hole Punching



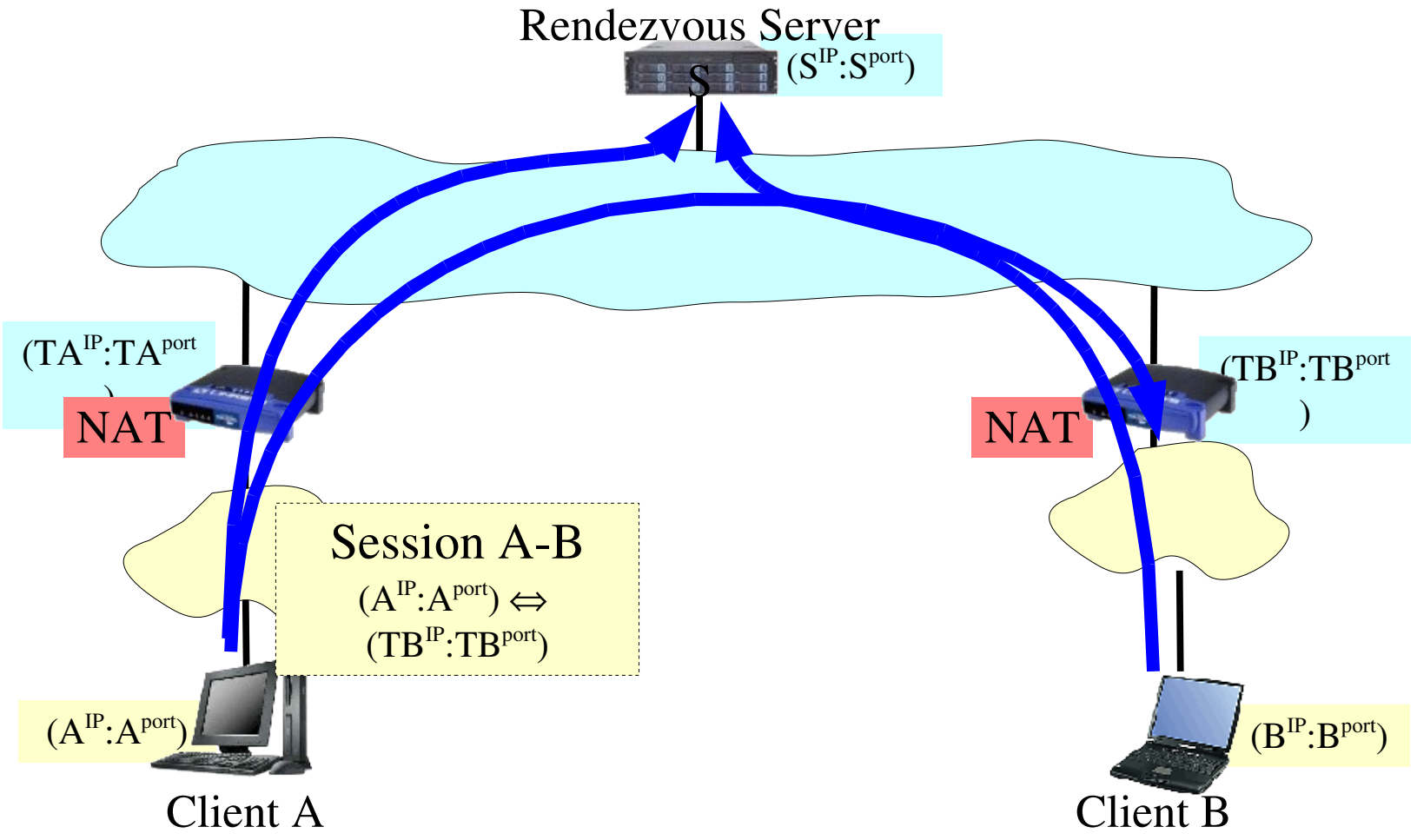
UDP Hole Punching



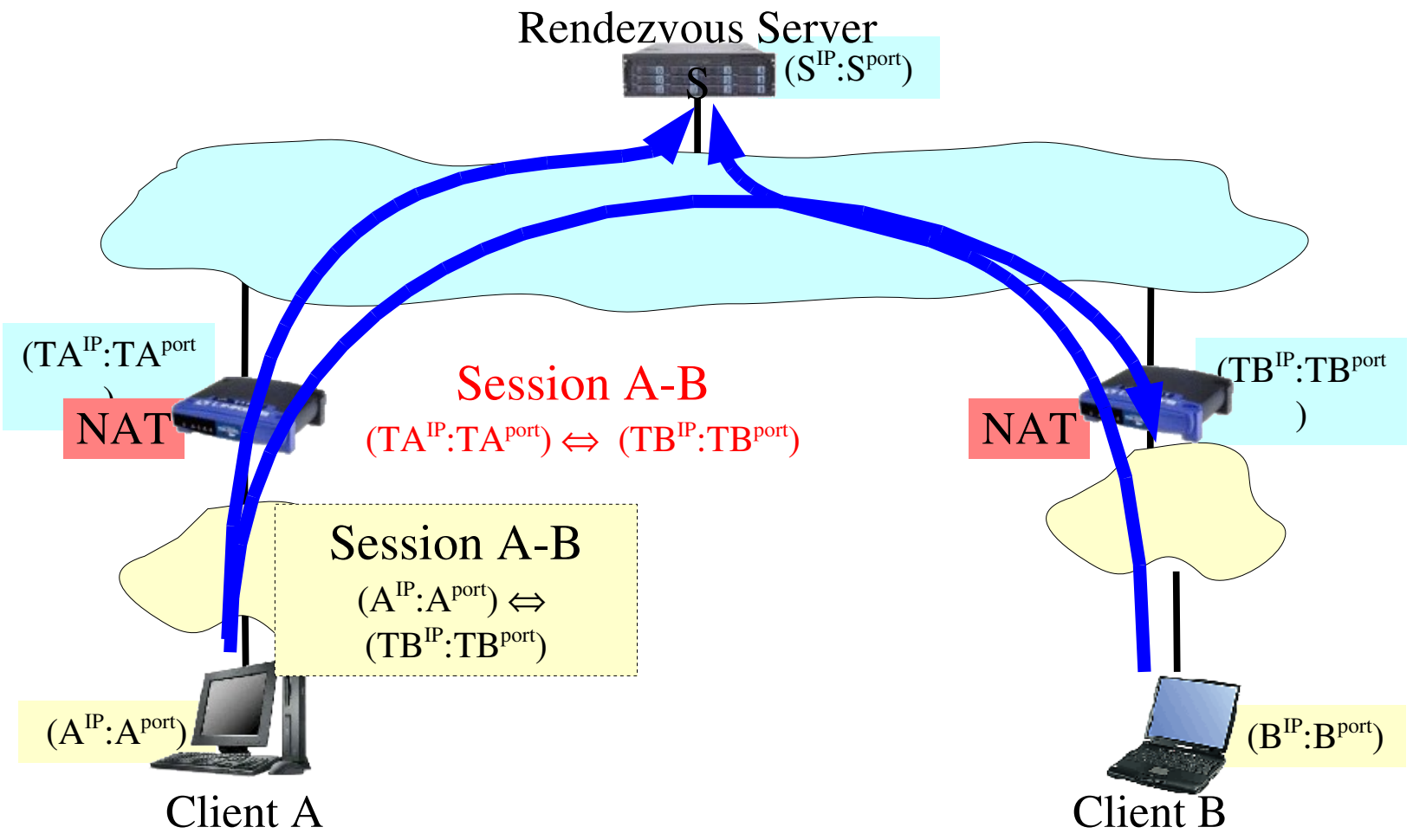
UDP Hole Punching



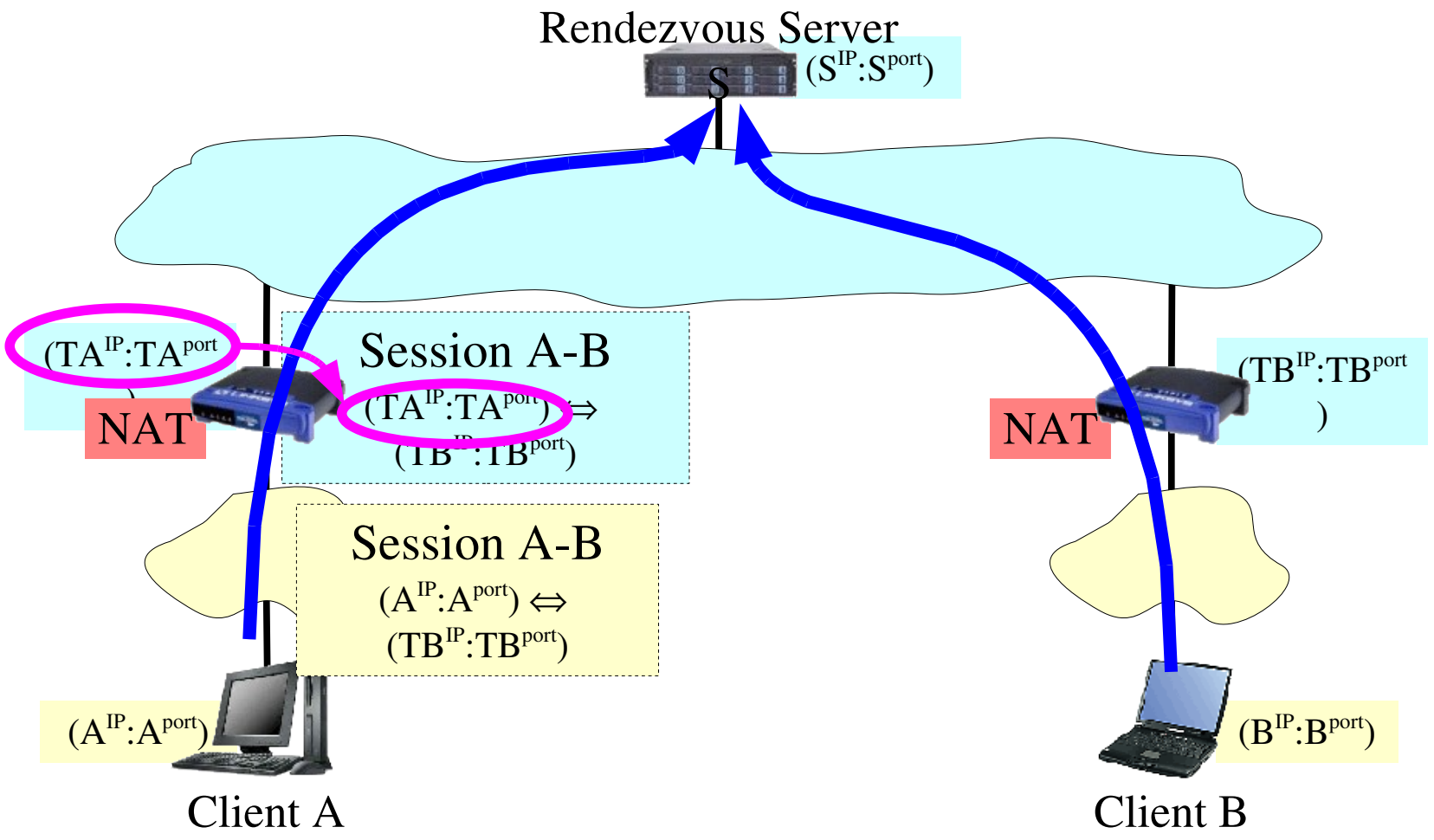
UDP Hole Punching



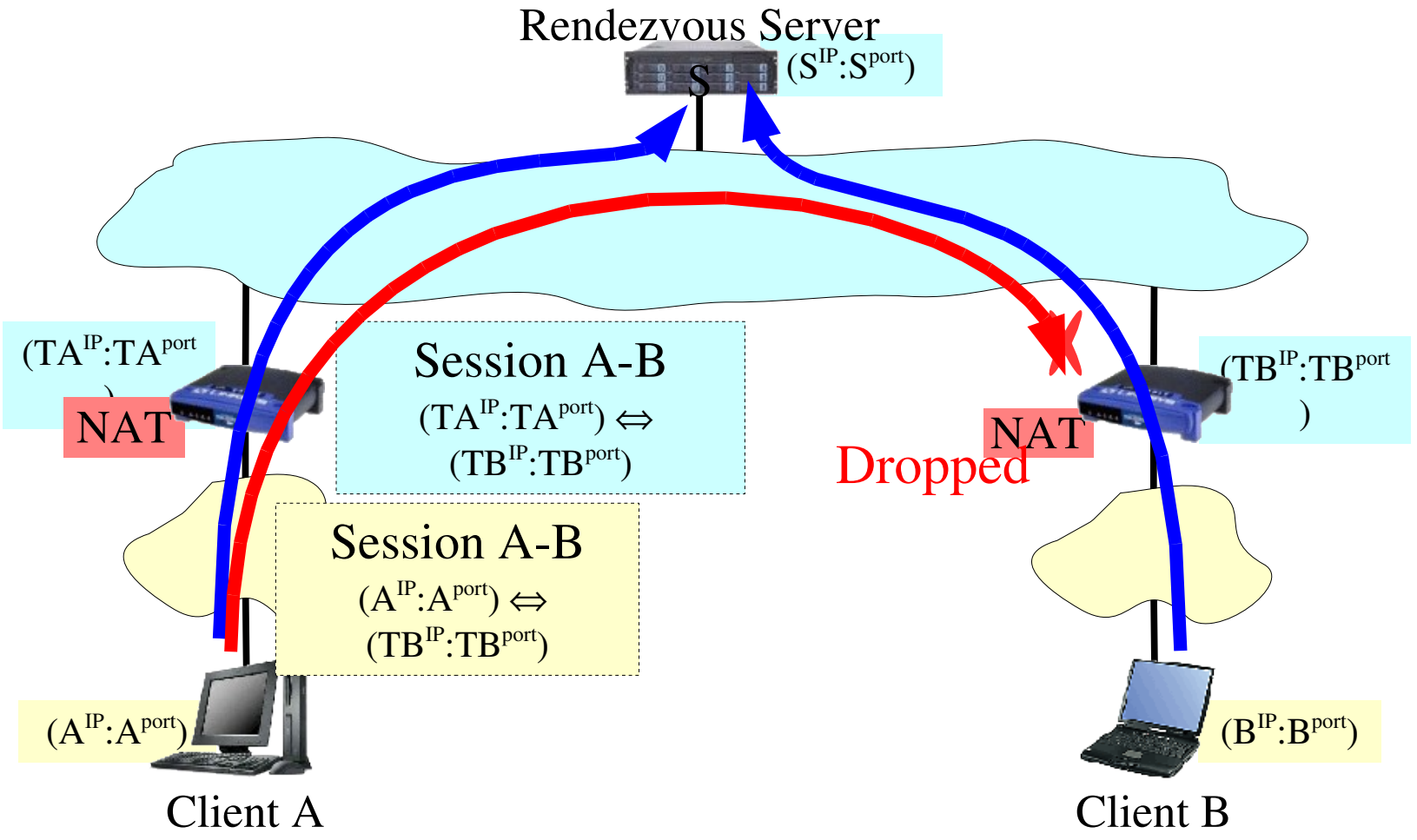
UDP Hole Punching



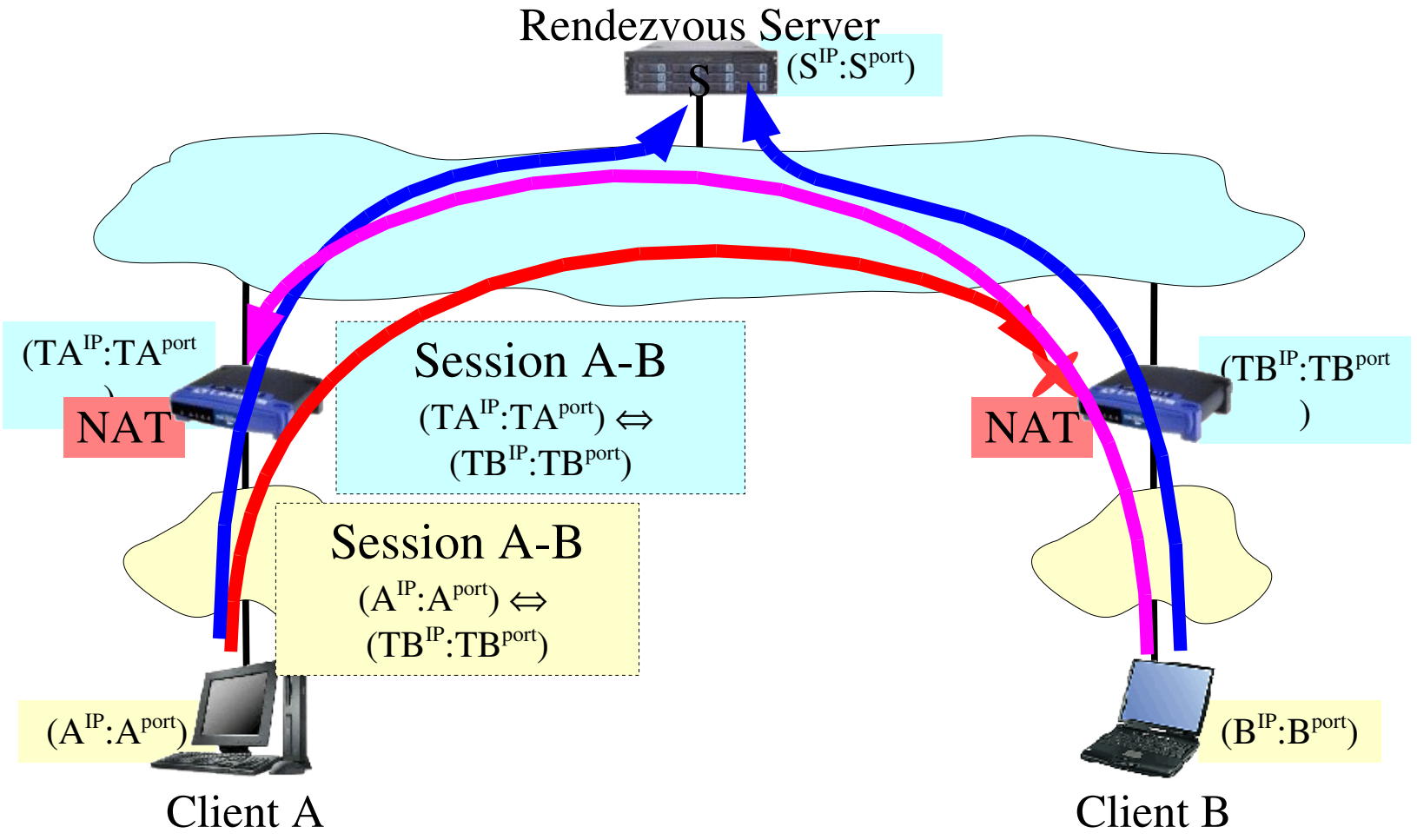
UDP Hole Punching



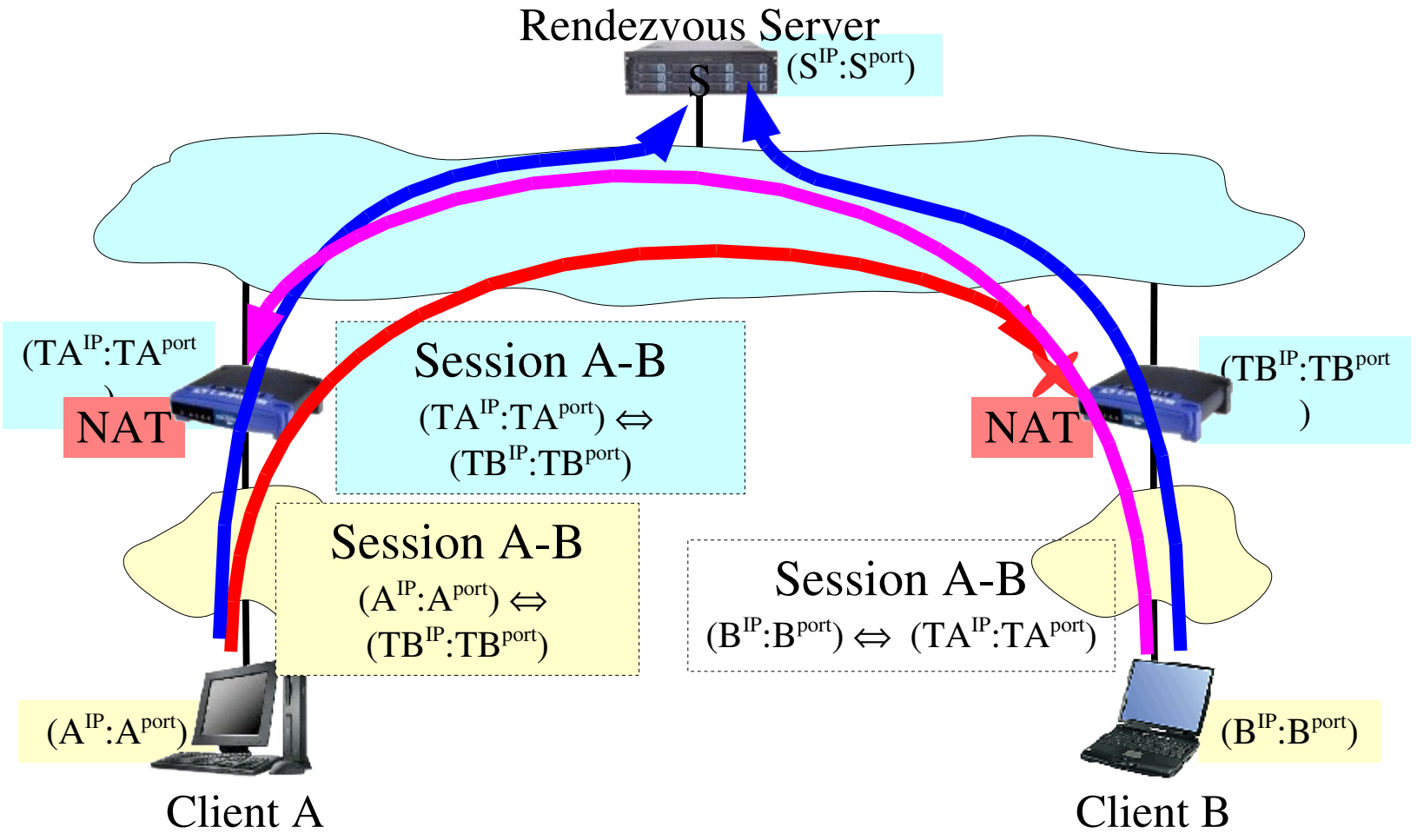
UDP Hole Punching



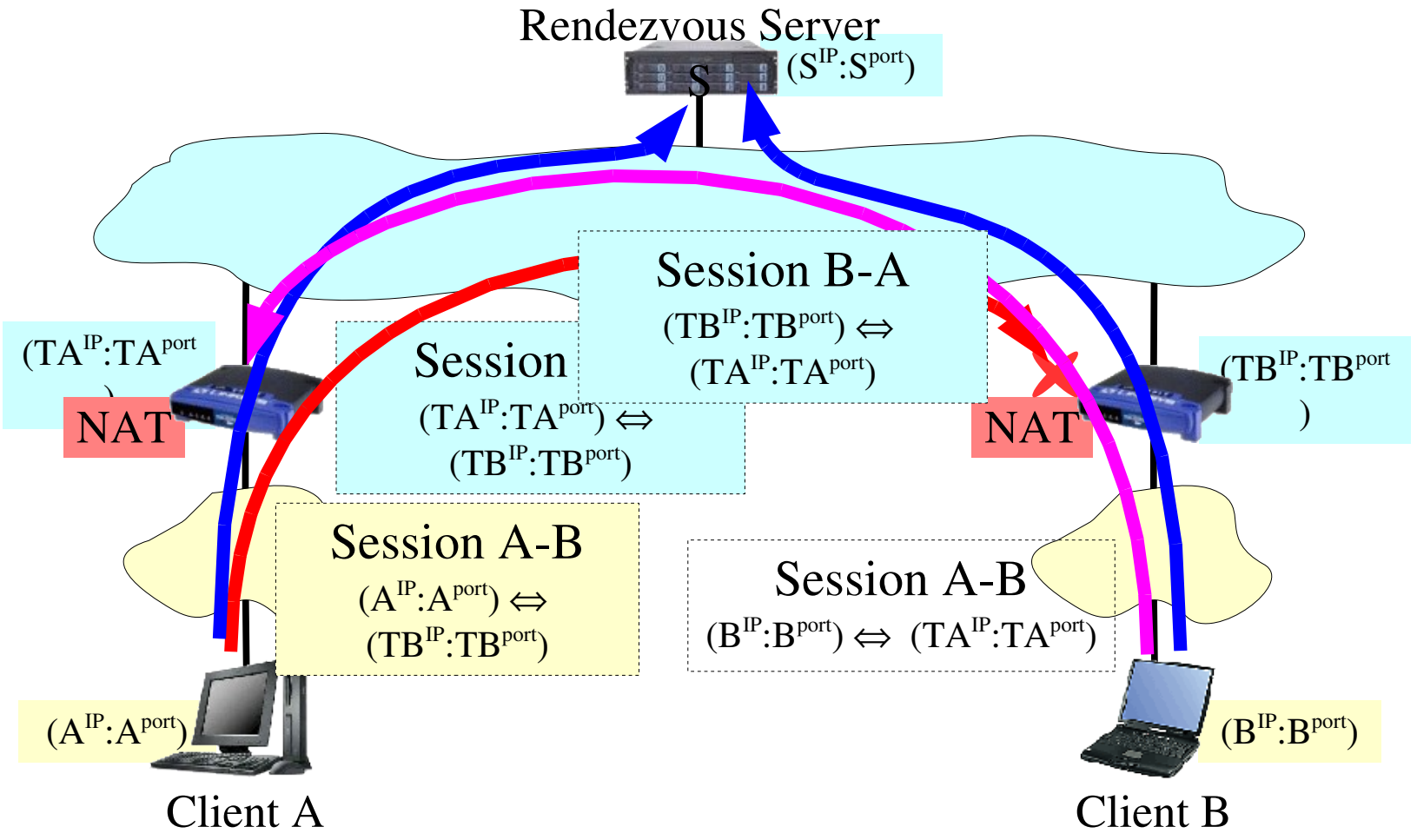
UDP Hole Punching



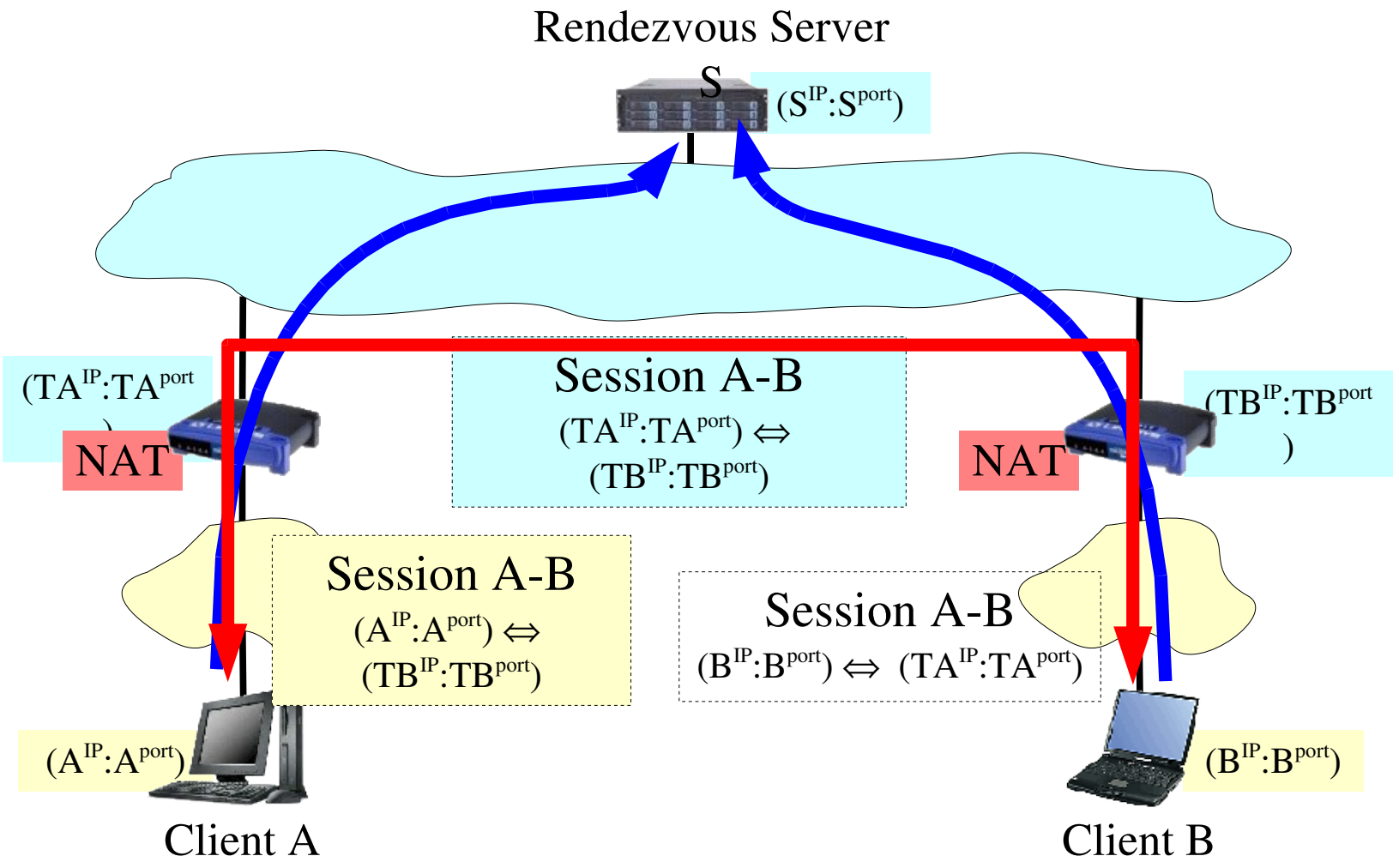
UDP Hole Punching



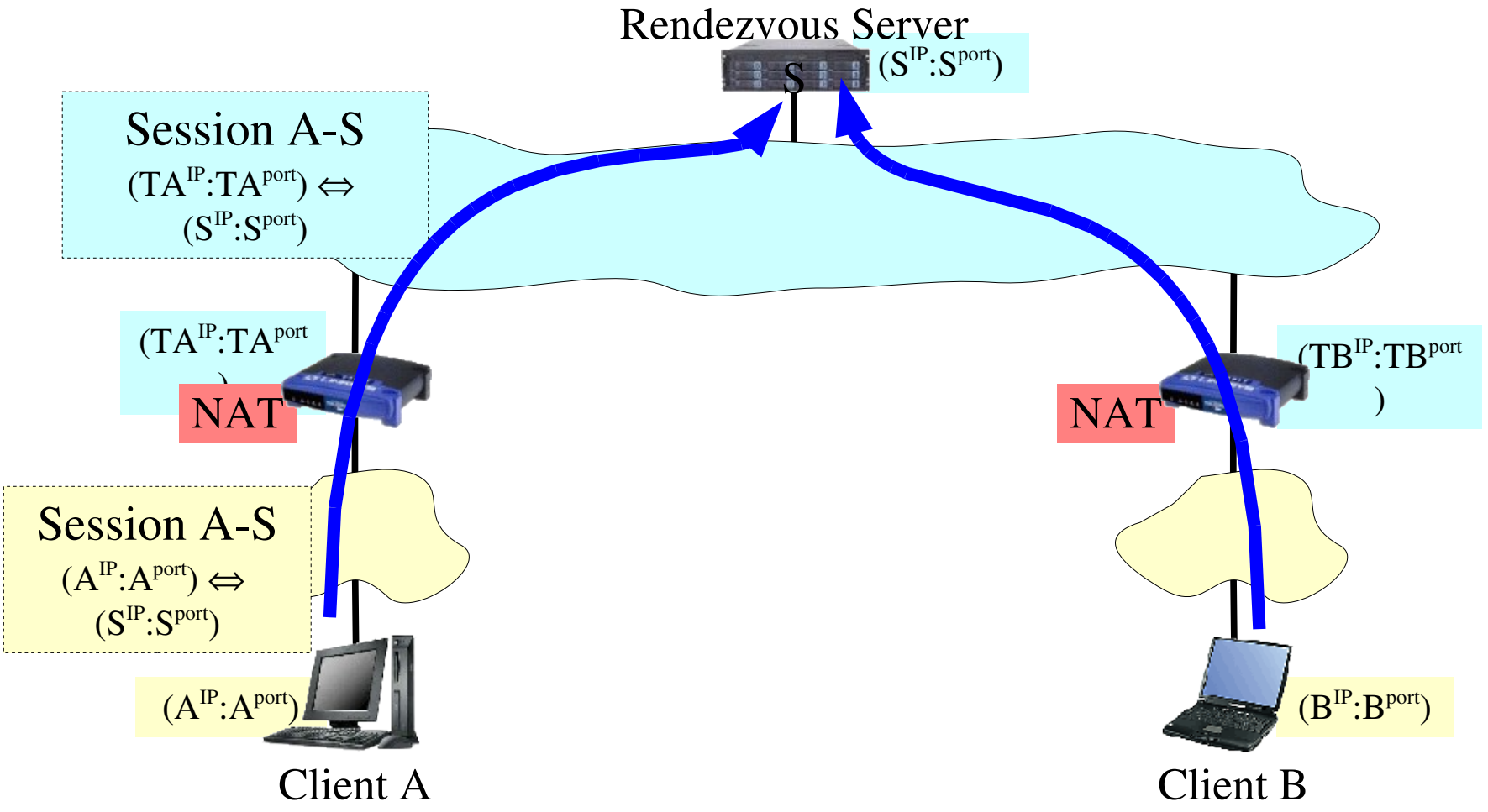
UDP Hole Punching



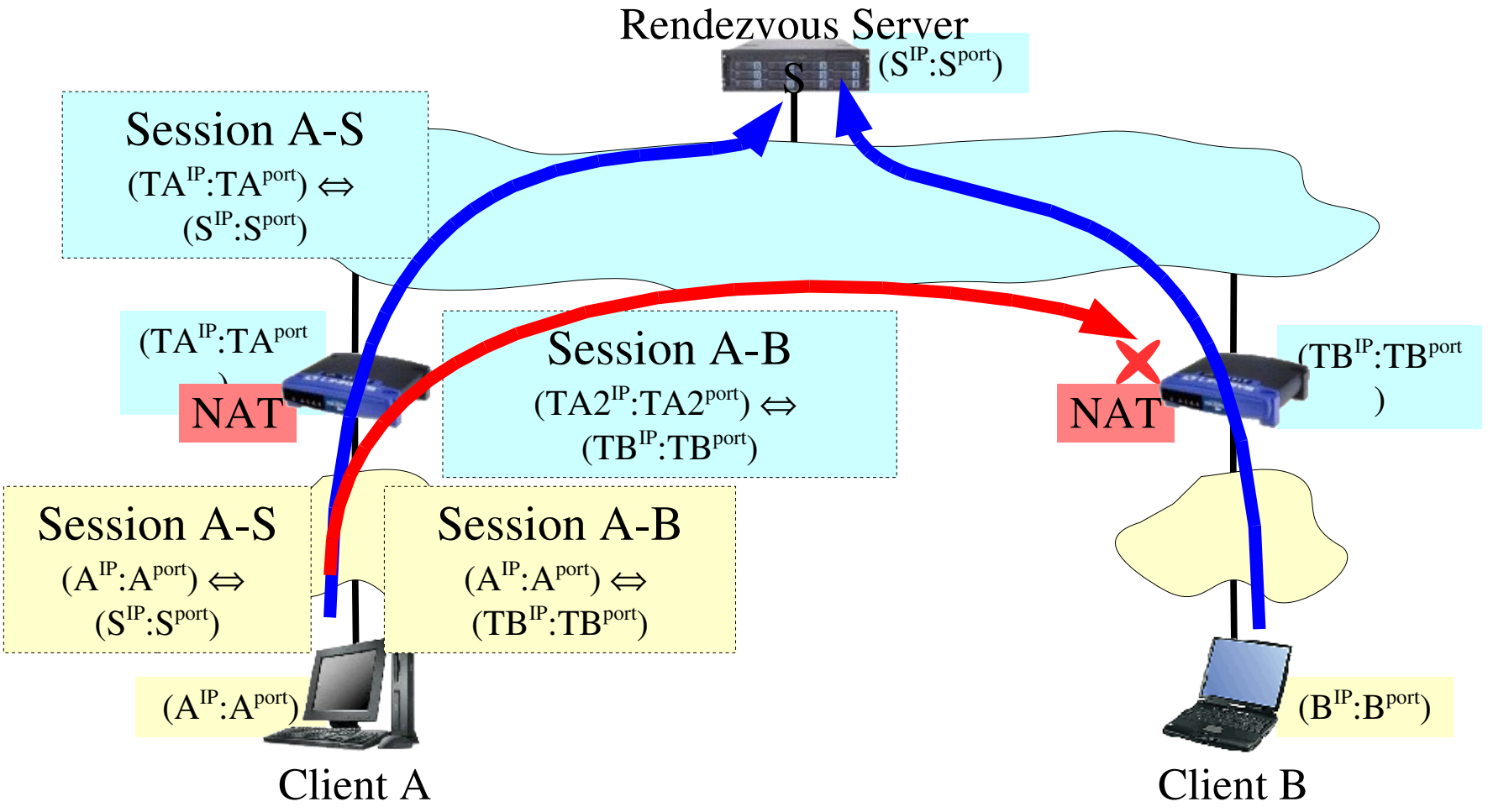
UDP Hole Punching



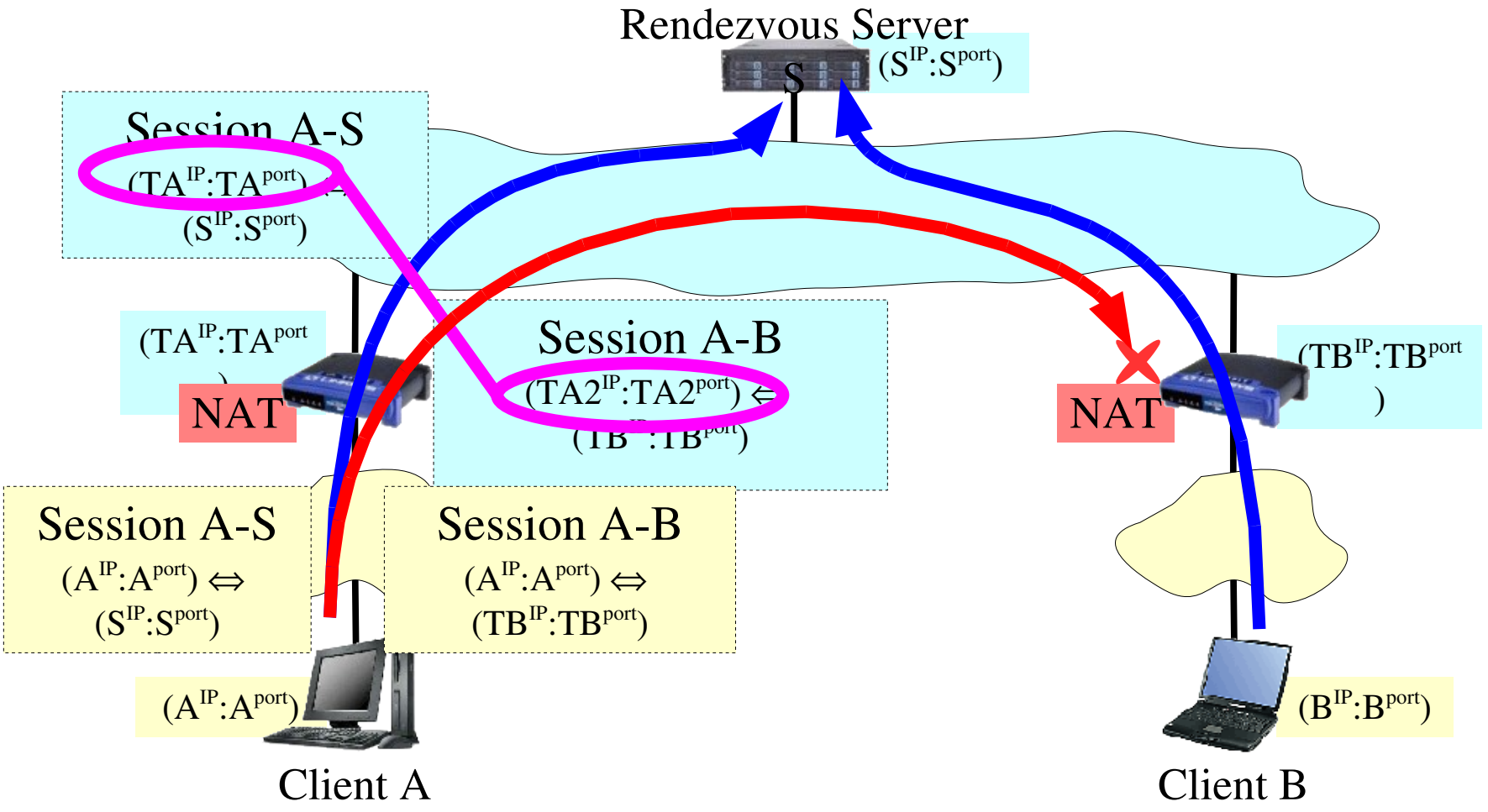
UDP Hole Punching Fallisce!



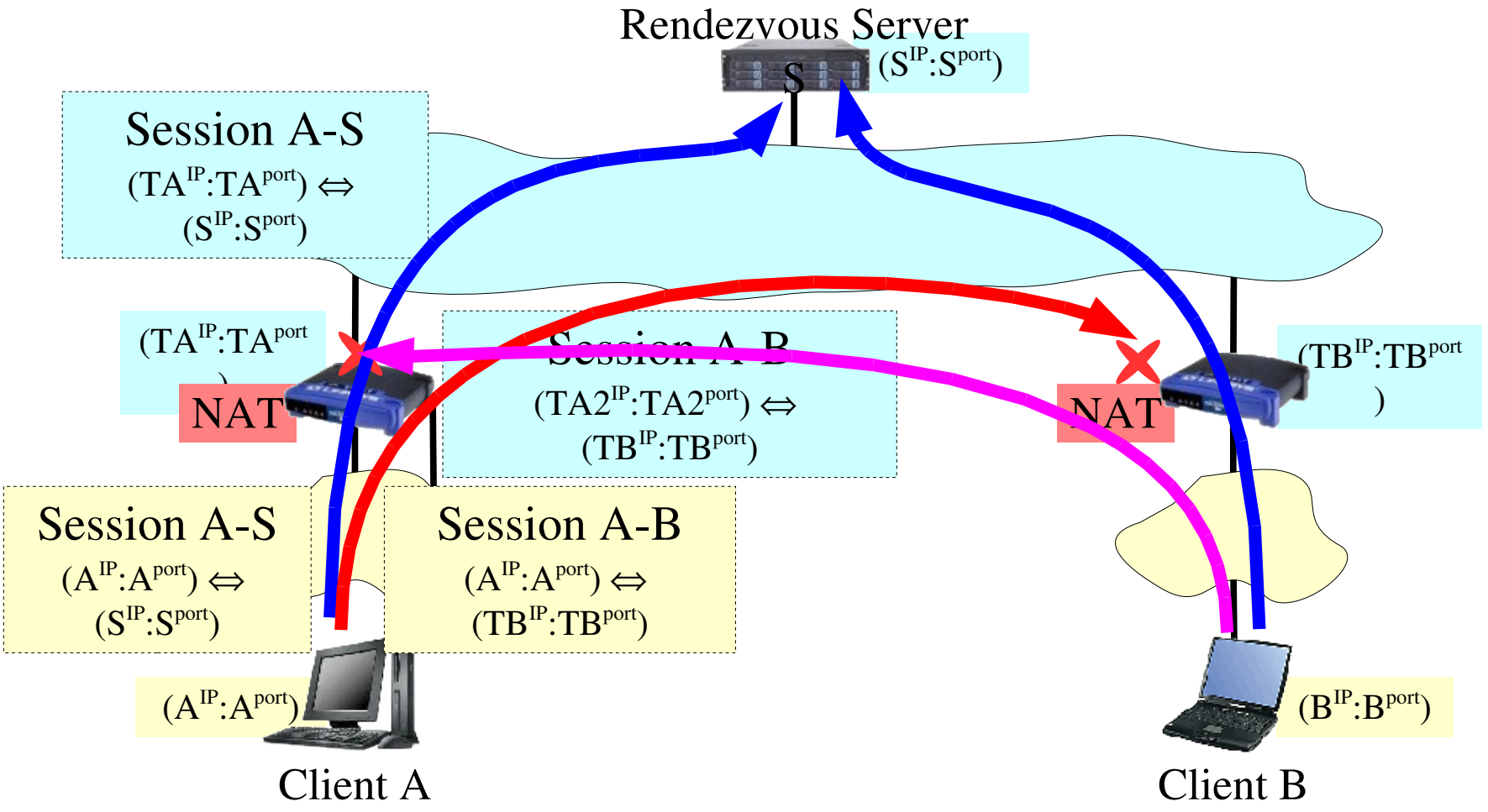
UDP Hole Punching Fallisce!



UDP Hole Punching Fallisce!



UDP Hole Punching Fallisce!



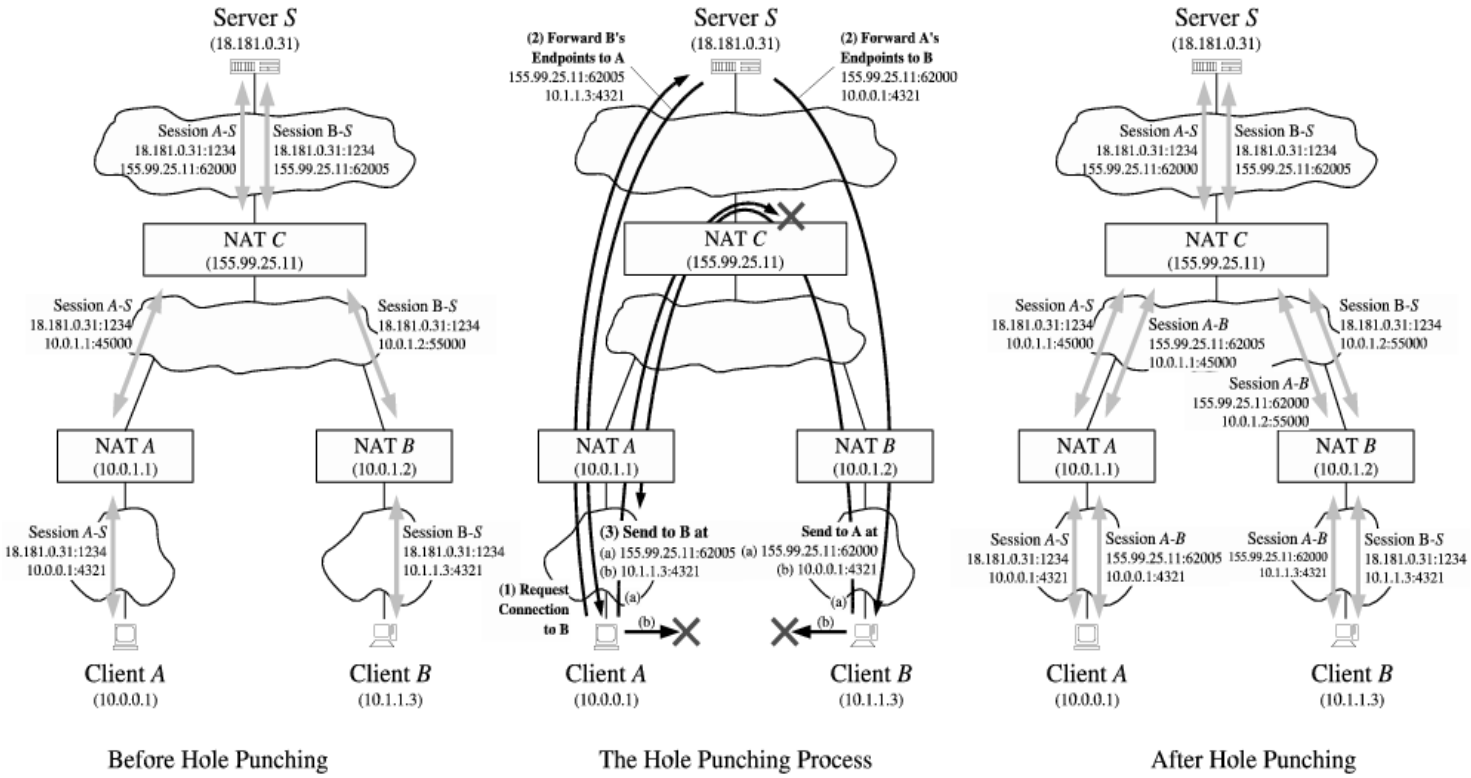
Note sul Hole Punching

L'hole punching lo posso fare anche mediante TCP
solo che ha maggiori limitazioni sul funzionamento.
In generale l'hole punching funziona molto bene con
NAT a cono pieno, mentre
ha bisogno di euristiche nel caso degli altri NAT:

- predizione delle porte
- predizione degli indirizzi successivi

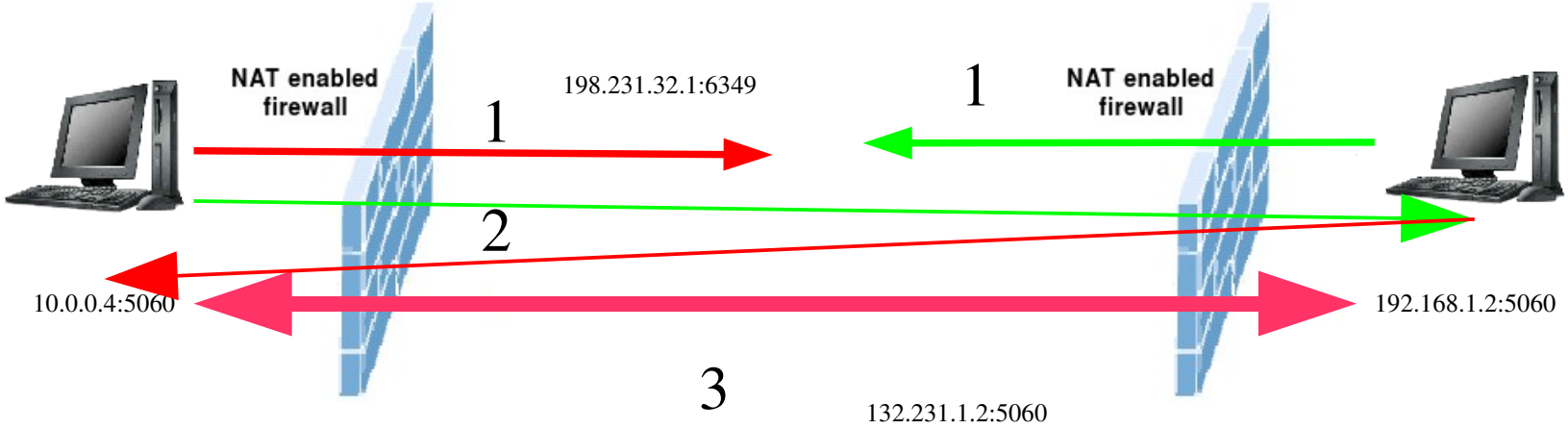
Per livelli multipli di NAT ha bisogno della hairpin translation: caratteristica supportata da sempre piu periferiche di NAT

Note sul Hole Punching: hairpin translation



NUTSS: TCP Hole Punching (libreria implementata in Java)

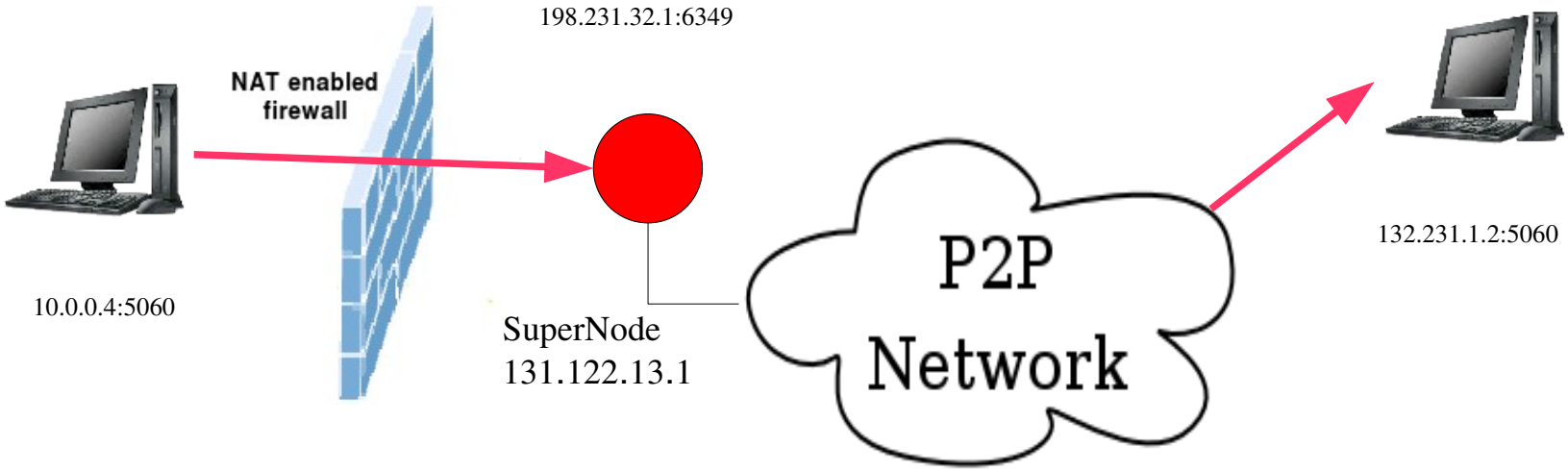
Usano un approccio sperimentale basato sul funzionamento del TCP, che secondo i loro test funziona negli ambienti SOHO



1. Ogni peer manda un SYN con TTL basso, tanto per passare il NAT: questo apre il buco.
2. Ogni peer manda un SYN/ACK all'altro
3. La connessione è stabilita

Come funziona TURN

BINDING_REQUEST
(ALLOCATE_REQUEST, LEN,(198.231.32.1,6349))



Nei casi in cui non riesco a far passare il traffico Voice over IP, mediante punching uso un SuperNodo come server TURN per il relay dei dati.
TURN : tiene dello stato ed ha lo stessi messaggi di STUN.

Riferimenti.

RFC 3489, STUN – Simple Traversal of UDP
through NATs, Rosenberg, Weinberger, Huitema and Mahy
NUTSS: A SIP Approach to UDP and TCP Connectivity.
Sito di NUTSS: http://nutss.net/stunt
Dan Kegel: UDP Hole Punching.
Rosenberg ICE.