



La Sapienza

Università degli Studi di Roma

Dipartimento di Informatica e Sistemistica

Computer Networks II

NAT - VPN - FIREWALLS

Luca Becchetti

Luca.Becchetti@dis.uniroma1.it

A.A. 2009/2010

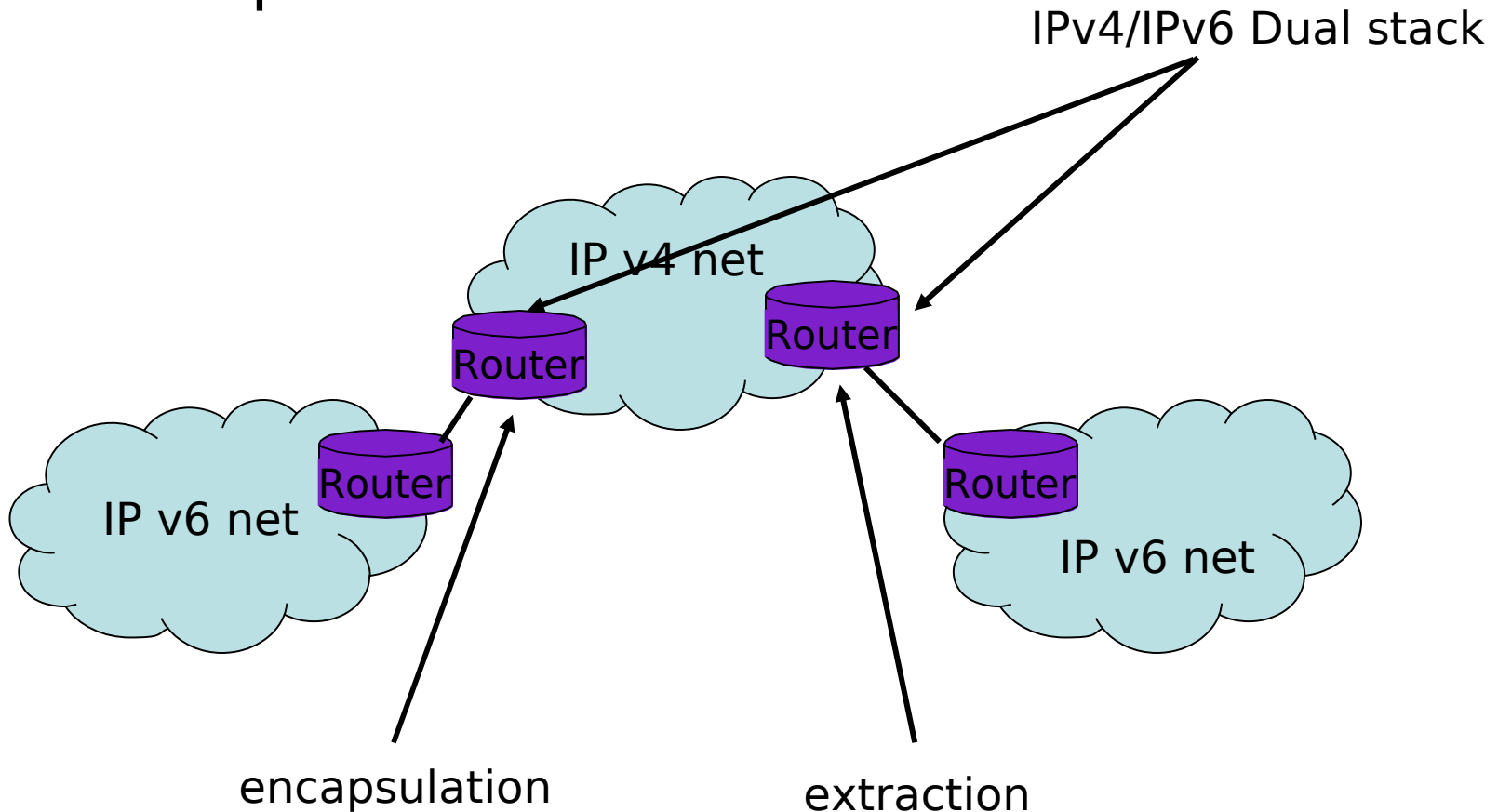
Overview

- Tunneling
 - Generic Route Encapsulation
 - Virtual Private Networks (VPN)
- Private addressing
- **Network Address Translation**
 - Private and public addresses
 - Main requirements
 - Reuse of IP addresses
 - Protection of subnets against external attacks
- Application gateway e firewall

Tunneling

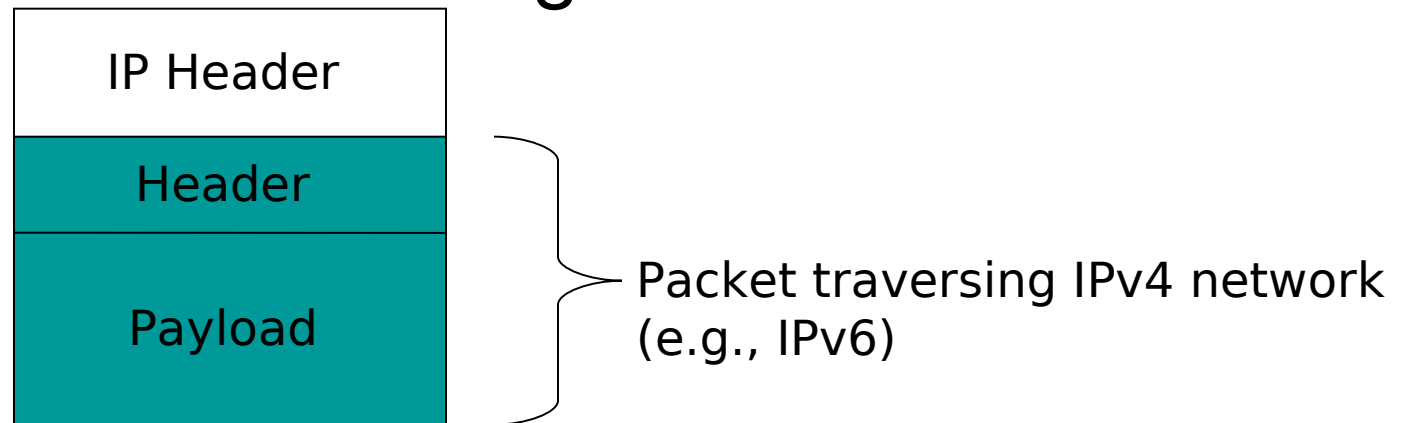
Tunneling

- Example: IPv6 over IPv4



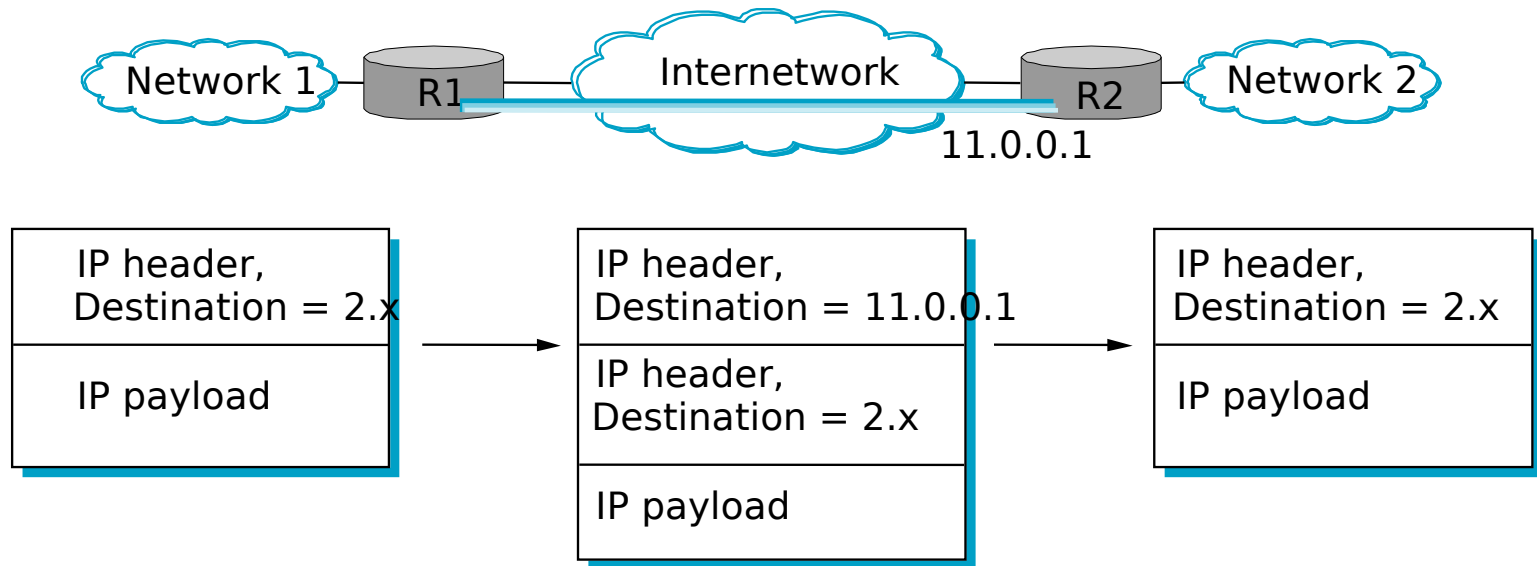
Tunneling

- General technique
- For IP networks: technique to carry non IP (or special purpose IP traffic - see further) traffic over an IP
- Main tools: encapsulation of non IP packets within IP datagrams



IP tunneling: example 1/3

- IP Tunnel over IP: allow a point-to-point virtual link [VC] between arbitrarily far apart nodes
- In the example: R1 and R2 are end points of IP tunnel

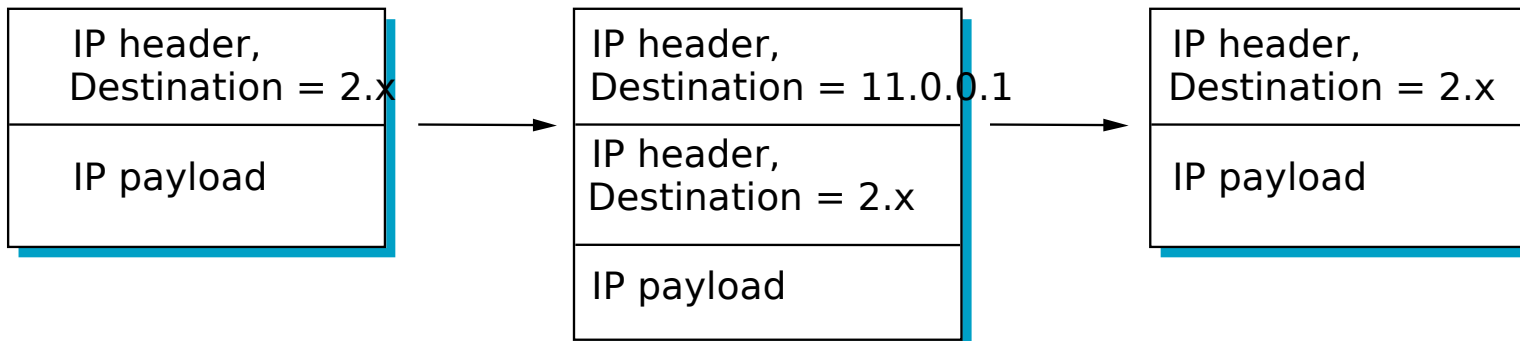


IP tunneling: example 2/3

RT at R1

NetNum	NextHop
1	Intf0
2	VIntf1
Default	Intf1

- Intf1 connects to Internetnetwork
- R1 encapsulates packets directed to virtual link within IP datagrams towards R2

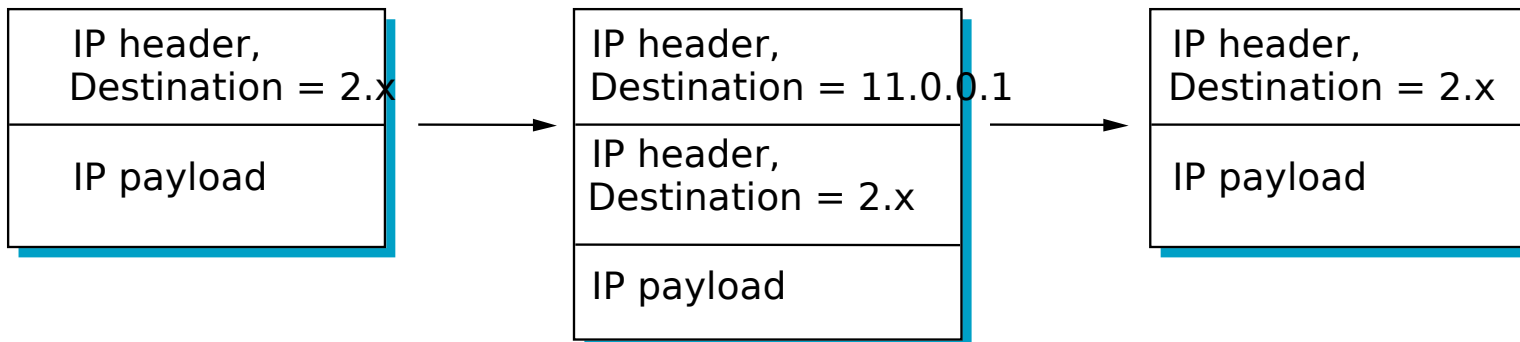


IP tunneling: example 3/3

RT di R1

NetNum	NextHop
1	Intf0
2	VIntf1
Default	Intf1

- R2 extracts encapsulated packet and “sees” that it is directed at network 2
- R2 delivers packet to destination



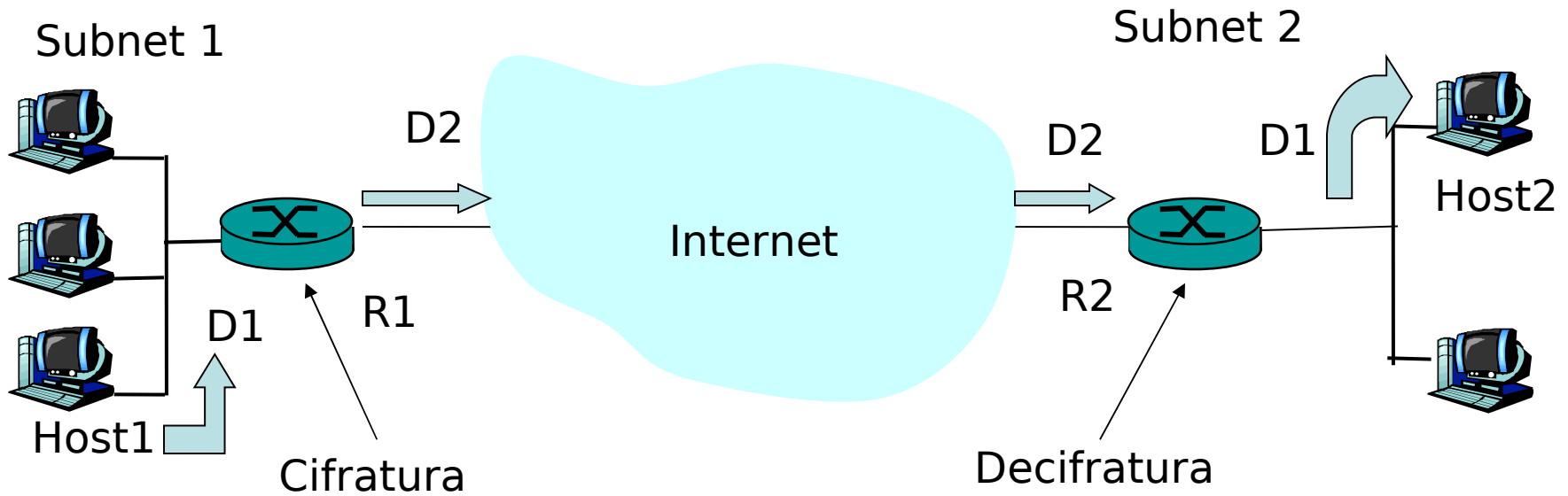
Why IP tunneling

- security
 - Tunneling + encryption allow safe connectivity over public networks
 - Solution often used in practice
 - Corresponds to most common definition of VPN
- Link to routers with capabilities not available elsewhere in the network
 - E.g., Multicast
- Traversing of IP segments by non IP traffic
 - E.g.: IPv6
- Mobile IP

Cryptography and tunneling

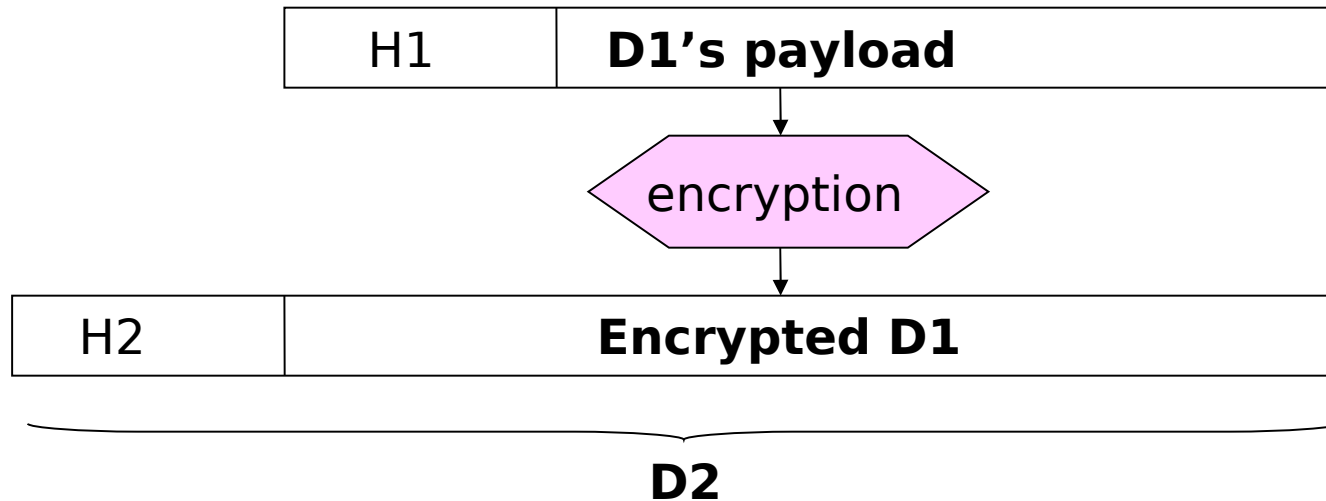
- Cryptography
 - Datagrams encrypted at source and decrypted at destination
- Tunneling
 - Encrypted datagrams encapsulated within standard IP datagrams as payload
- Only visible part: external datagram's header
 - Payload contains original, encrypted packet
 - Source and destination IP addresses (of original packet) encrypted

E.g.: Datagram from Host1 to Host2



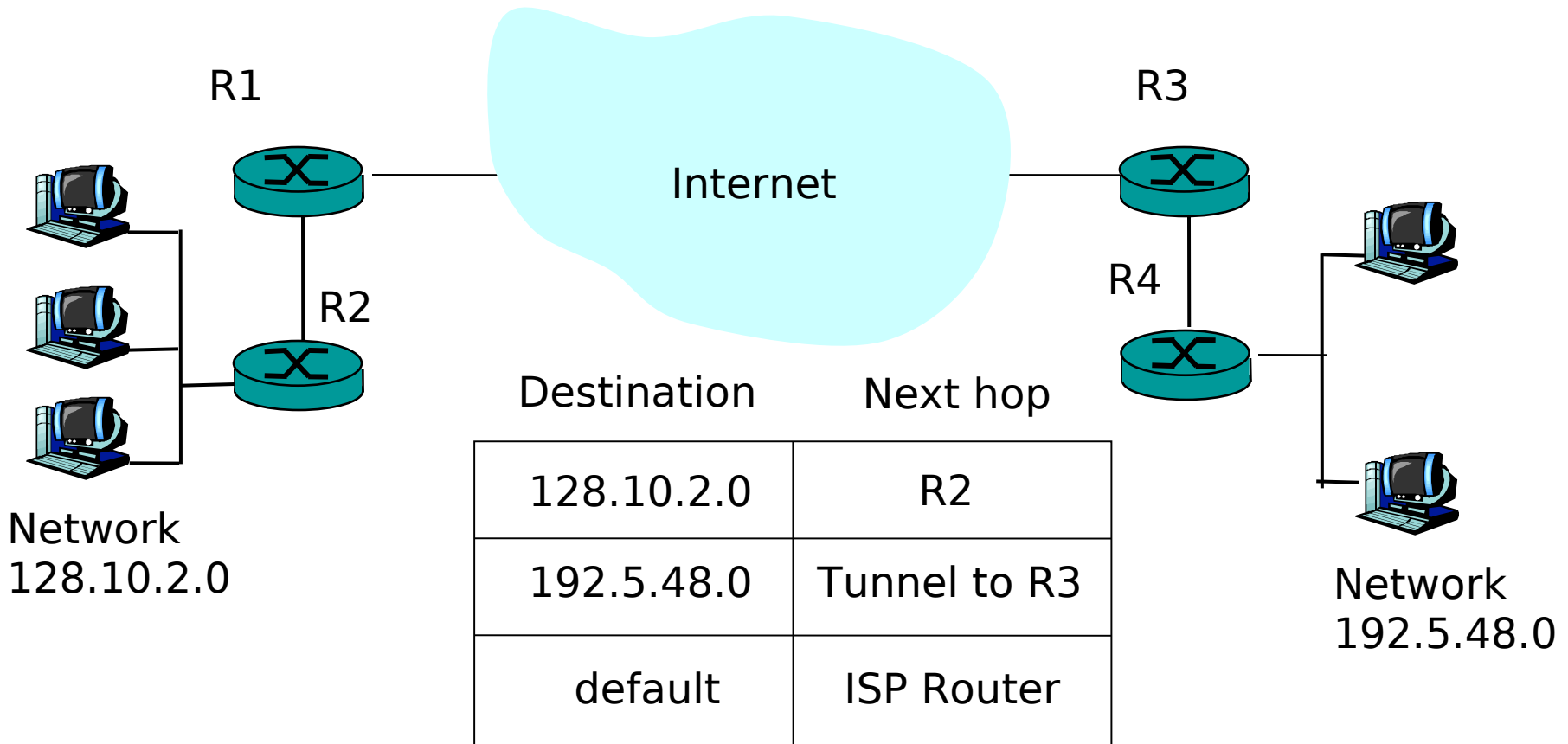
- Datagram D1 from Host1 to Host2
- D1 encrypted at R1 \Rightarrow D2
- D2 reaches R2 and is decrypted \Rightarrow D1
- D1 is delivered to Host2

Example - cont.



- H2 not encrypted
- IP addresses in H2 are those of R1 and R2 respectively
- Addresses of Host1 and Host2 in H1 [hence encrypted]

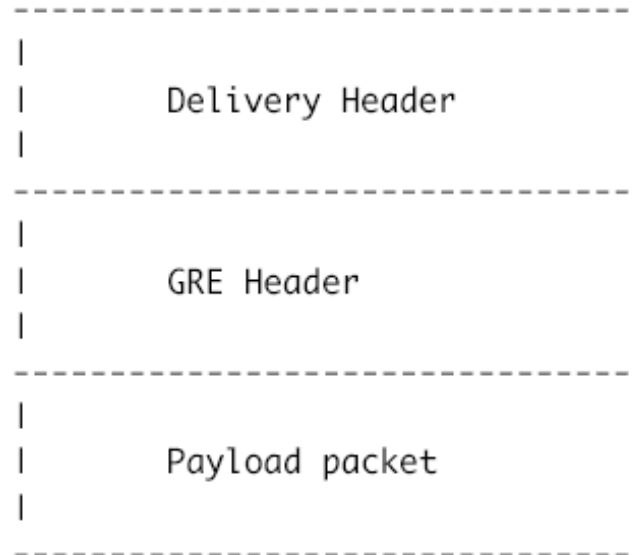
E.g.: R1's routing table



- Default entry indicates that Datagram is sent to the Internet without encryption and tunneling

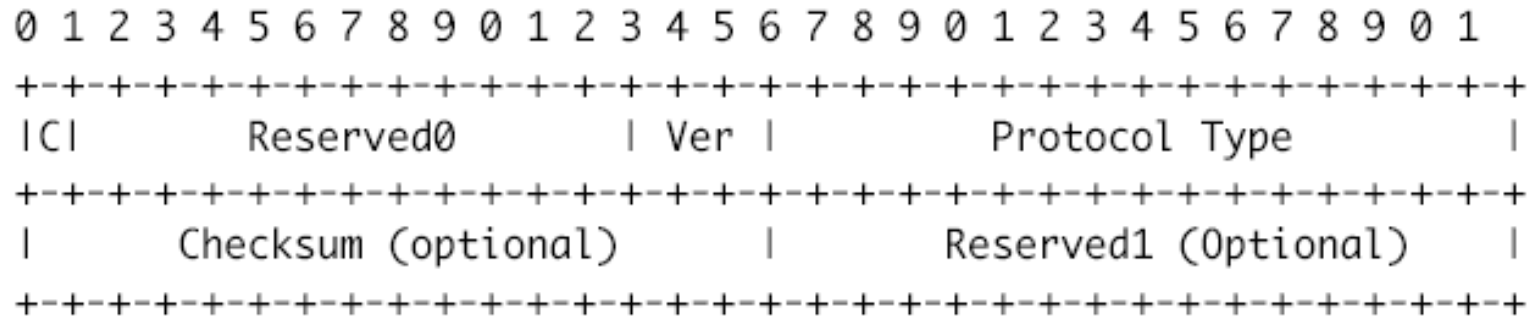
Case study - GRE

- Generic Routing Encapsulation (Cisco)
 - RFC 1701 (first version)
 - RFC 2784 and 2890
- “This document specifies a protocol for encapsulation of an arbitrary network layer protocol over another arbitrary network layer protocol.” - RFC 2784



GRE/cont.

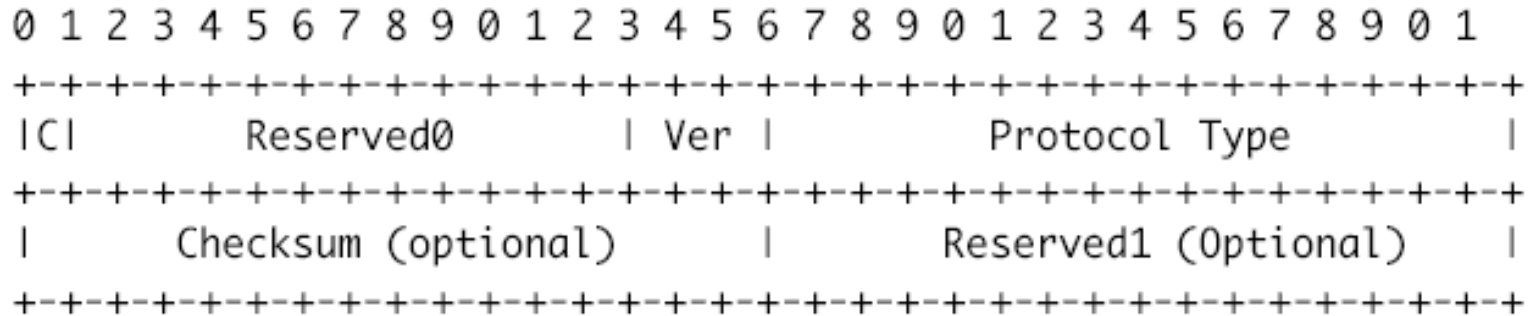
The GRE packet header has the form:



- Bit C: if 1 --> Checksum and Reserved1 fields are valid
- Reserved0: bit 1-5 used if receiver implements RFC 1701
- Ver: must be 0

GRE/cont.

The GRE packet header has the form:



- Protocol type: identifies protocol of encapsulated packet
- Checksum: Internet checksum over GRE header and payload (excluding checksum itself)
 - Present if C = 1
- Reserved1
 - Present if C = 1
 - Used in RFC 1701 compliant implementations

Payload - IPv4

- From RFC 2784
 - “When a tunnel endpoint decapsulates a GRE packet which has an IPv4 packet as the payload, the destination address in the IPv4 payload packet header **MUST** be used to forward the packet and the TTL of the payload packet **MUST** be decremented. Care should be taken when forwarding such a packet, since if the destination address of the payload packet is the encapsulator of the packet (i.e., the other end of the tunnel), looping can occur. In this case, the packet **MUST** be discarded.”
- IPv4 over IPv4 obviously possible. Example: VPN’s using IP tunneling
 - Example: RFC 1701 compliant tunneling

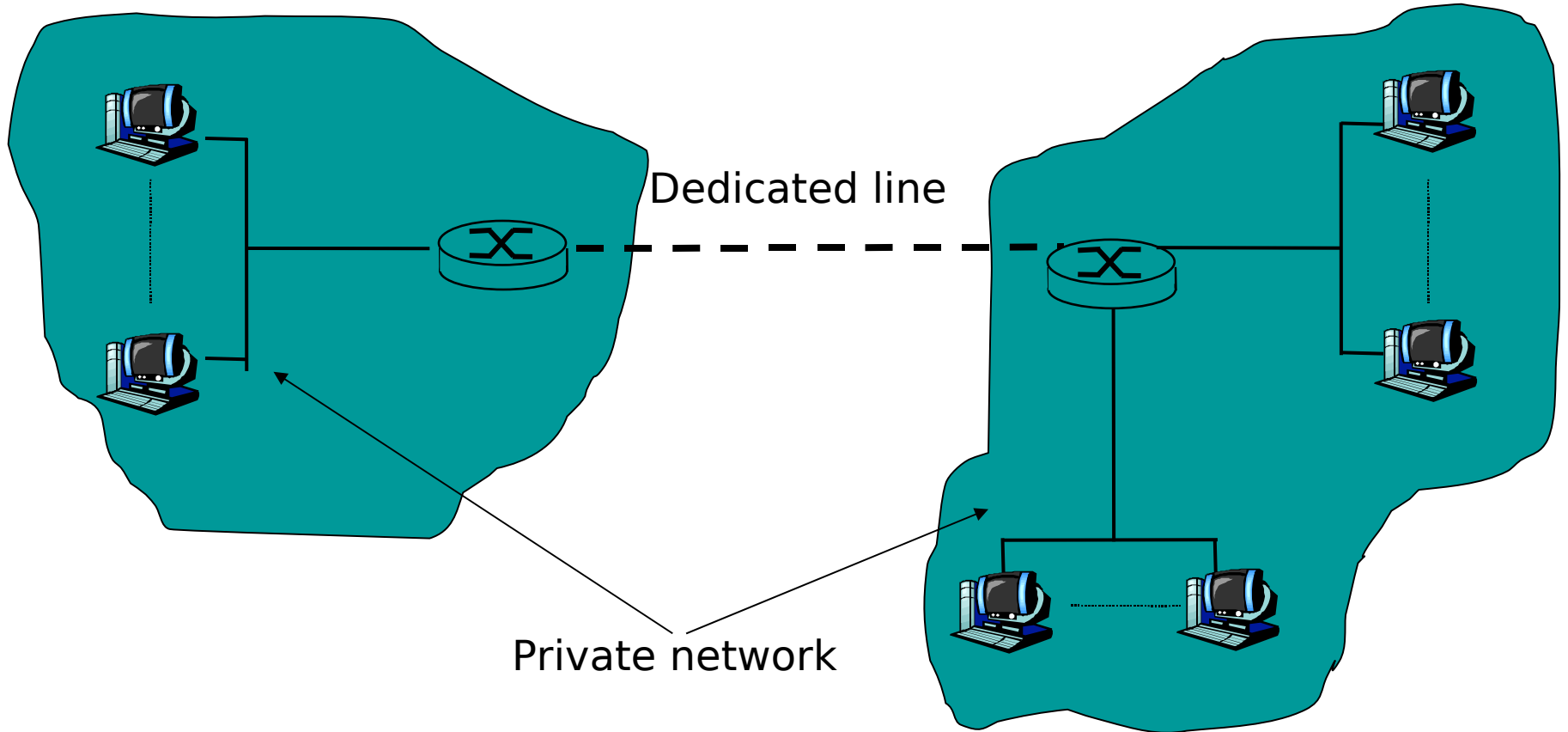
Extensions

- Set of options in first protocol version (RFC 1701)
 - Valid when flag C = 1
- Possible to specify routes (sequence of routers to traverse)
- Encryption
 - Key field
- Sequence numbers
- Checksum

Private networks

- Private network: group of nodes (servers, hosts, routers) not accessible to other networks' traffic
- Typical implementation: rental of lines interconnecting nodes of private network
- Potential advantages
 - Address reuse
 - Privacy
- May be hard/expensive to implement

Completely isolated network

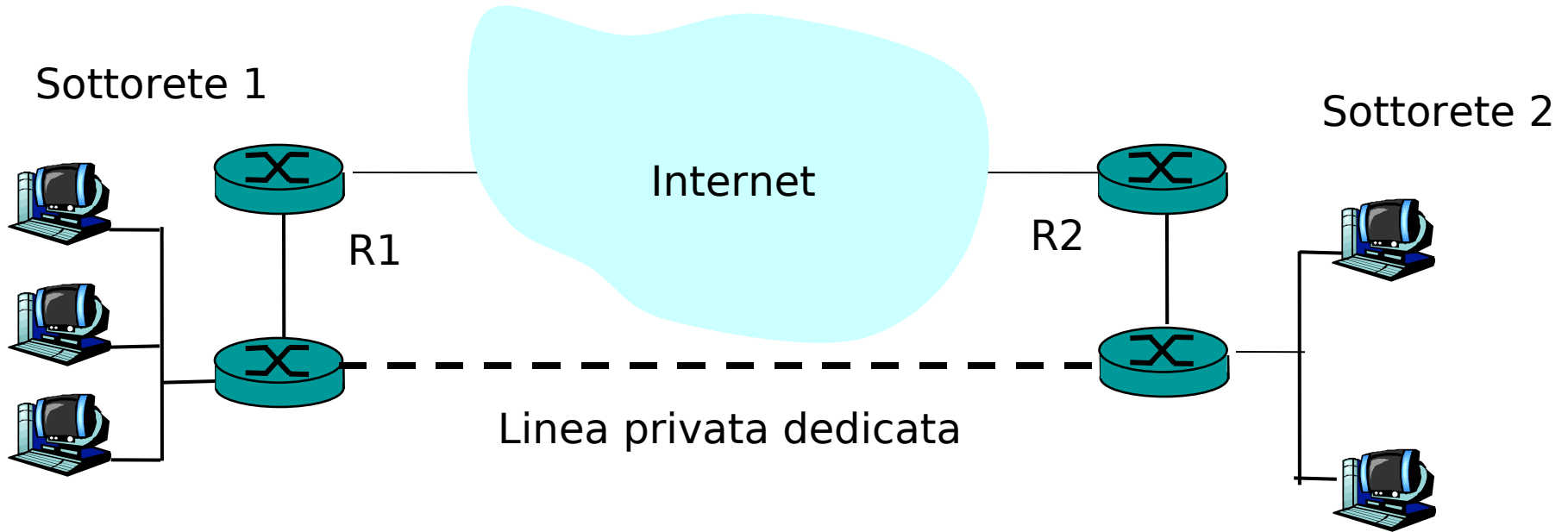


Not possible to access network from outside

Reti ibride

- Linee dedicate connettono i diversi gruppi di nodi che formano la rete privata
- Indirizzi IP pubblici
- Alcuni o tutti i gruppi sono connessi al resto di Internet
- Possibilita' di riservatezza. Es.:
 - Traffico privato su linee dedicate
 - Traffico da/verso il resto di Internet su linee condivise

Reti ibride - esempio



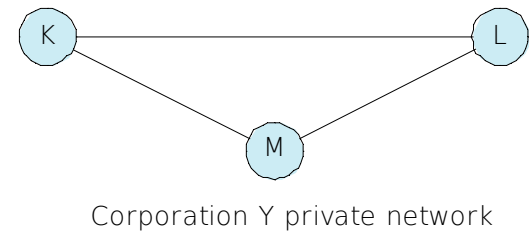
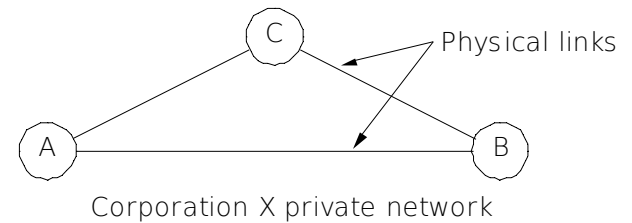
- La Sottorete 1 puo' comunicare con la Sottorete 2 in modo riservato usando la linea dedicata
- L'accesso a Internet avviene attraverso i router R1 e R2
- La rete usa indirizzi IP validi

Virtual Private Networks (VPN)

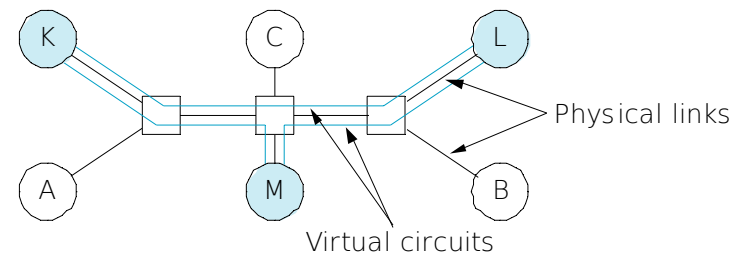
- Virtual Private Network (VPN)
 - No use of dedicated lines
 - Traffic to/from nodes of private network “virtually” separated from traffic to/from other networks
 - E.g.: by encryption
- Different solutions possible
 - Widespread solution: encryption + tunneling

Esempio - circuiti virtuali

- Nella figura (a) sono mostrate due reti private fisiche
- Nella figura (b) sono mostrate le corrispondenti reti private virtuali
- Si usano circuiti virtuali (es. ATM o Frame-Relay)
- I pacchetti della rete X non possono essere intercettati dalla rete Y se non esiste almeno un VC comune tra esse



(a)



(b)

Private addressing

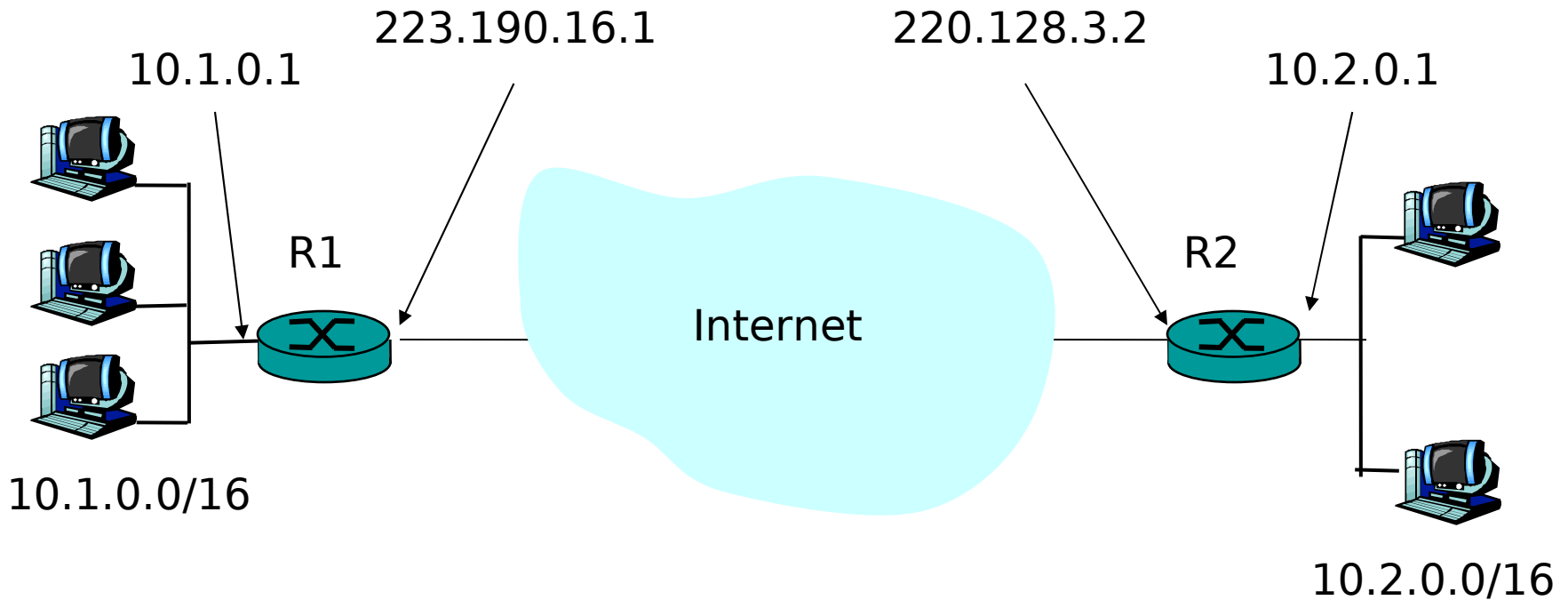
Private addresses

- CIDR specifies a set of address blocks for use in private networks, not directly connected to the Internet
- **Non routable** addresses
 - A datagram with a private destination address is recognised and dropped by an Internet router

Private addresses - cont.

- Private address blocks
 - 10.0.0.0/8
 - 172.16.0.0/12
 - 192.168.0.0/16
 - 169.254/16 (Link-local addresses)
 - Used when no available DHCP server
- More Internet subnets may use these address blocks, as long as subnets are not directly accessible from outside
- VPNs often use private addressing for internal nodes

VPN with private addresses



- 2 subnets forming VPN use private address blocks
- R1 and R2 need 2 *global* addresses for tunneling

Pros/cons

- Less global addresses needed
- Problem: hosts of private subnets have no direct Internet access
 - Global IP address needed to this purpose
- Fixes
 - Application gateways
 - NAT: Network Address Translation

Network Address Translation

NAT - Network Address Translation

- Association (mapping) between private and public addresses
- NAT allows IP level access to the Internet
- Requirements
 - Single access to the Internet
 - At least one global IP address available
- Application Gateways allow hosts to access specific services (e.g., HTTP, SMTP etc.) but not at IP level

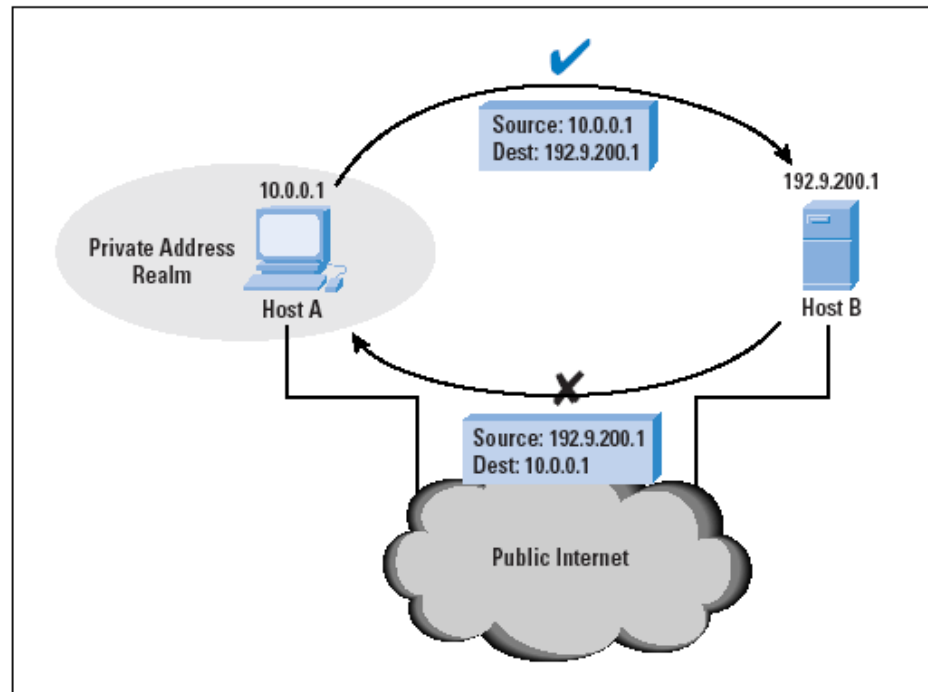
Motivations and history

- First proposed in 1994 to address IP address scarcity
 - Idea: at any given moment, only a subset of internal nodes needs access to the Internet
- Nowadays, NAT used for many purposes
 - Design networks that are not visible outside
 - Security
 - Special cases:
 - Network masquerading
 - Native address translation
 - Port address translation [**PAT**]

NAT

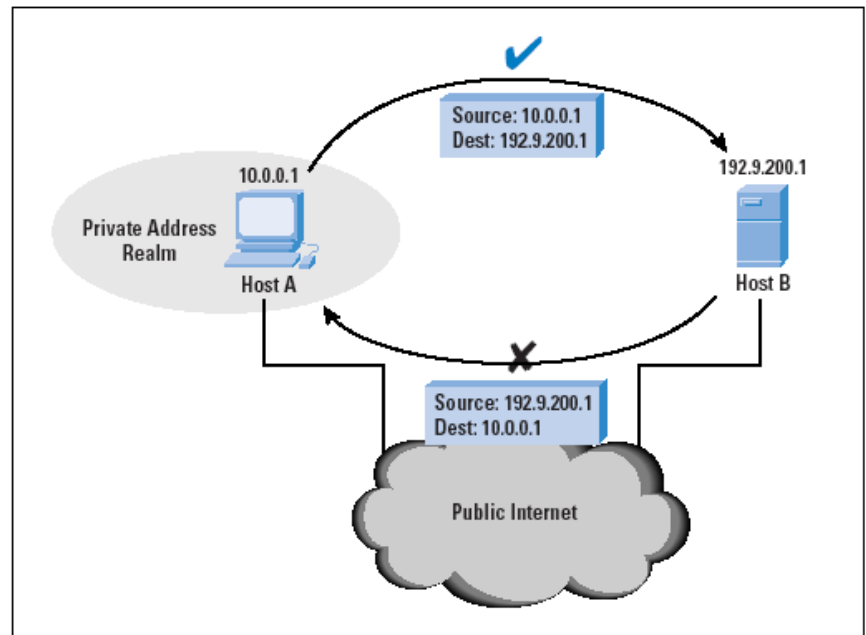
- Leaving private address space is not the problem...
-return path is the mess

Figure 2: Public/Private Communication



Basic mechanisms

- Pool of global IP addresses assigned to NAT
- Global addresses mapped to private ones when needed
 - Inbound packets directed to a non-mapped global address \Rightarrow dropped
- Packets sent from source or sent as replies from destination are **suitably manipulated**
 - Which IP header fields?
 - Is payload modified?



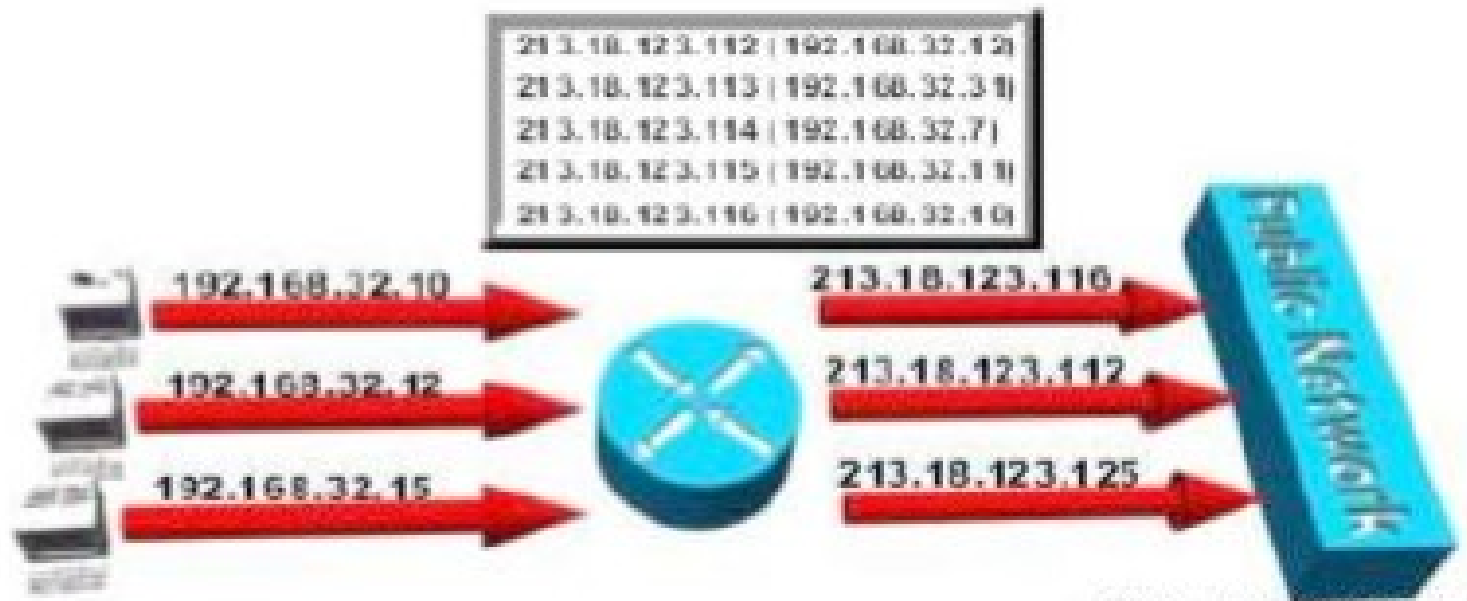
Static NAT

- Static association between global and private address
- Used when a node has to be accessible from outside
- NAT router in this case offers protection



DynamicNAT

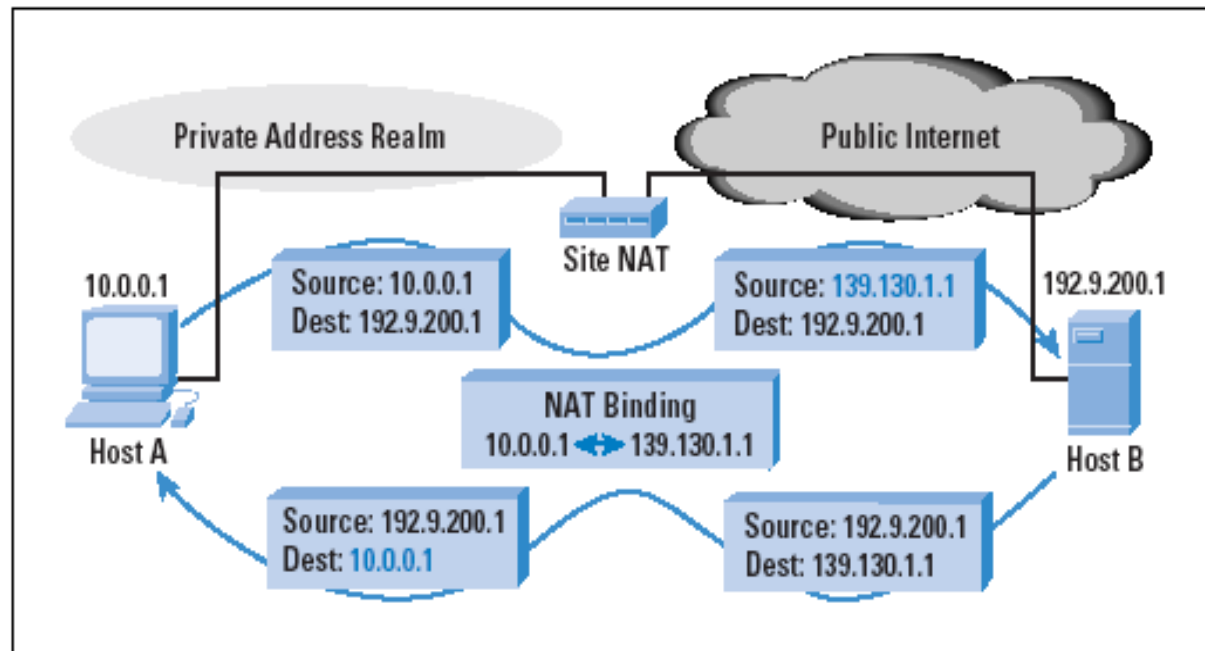
- Dynamic binding between global and local address
- NAT has a pool of global IP addresses
- Same private address can be associated to *different* global addresses over time



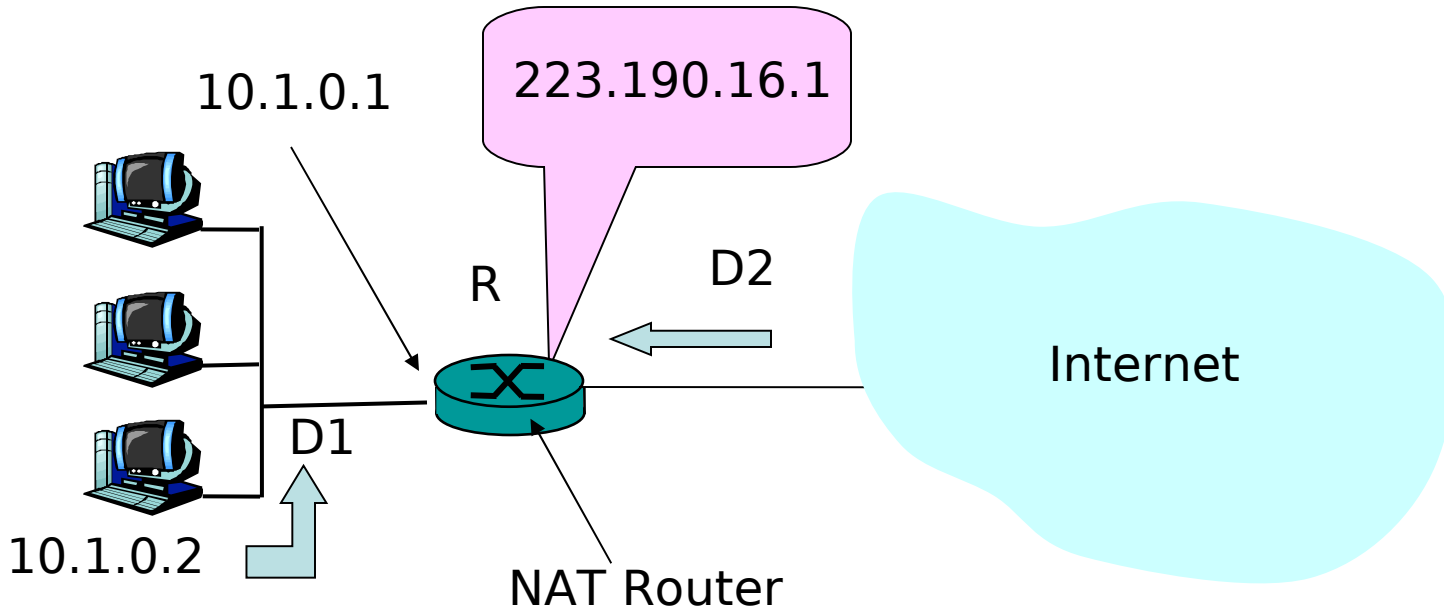
Dynamic NAT

- Dynamic binding
- Binding's expiry: timer-dependent

Figure 3: NAT Traversal



Dynamic NAT: example



“Translation” for datagrams to/from 10.1.0.2

- D1’s IP source: 10.1.0.2 \Rightarrow 223.190.16.1
- D2’s IP destination: 223.190.16.1 \Rightarrow R
- D: how does R understand that D2 is directed to 10.1.0.2?

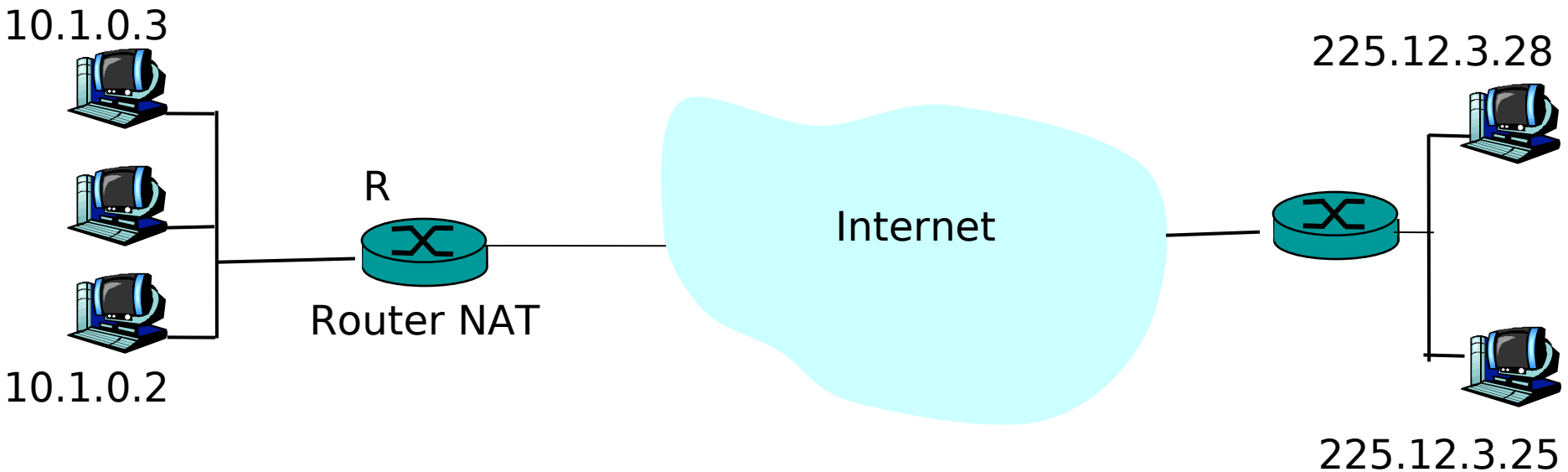
Implementation issues

- All Internet packets reaching R have same destination address (in the example: 223.190.16.1)
- How to map global destination address to corresponding private one within local network?
- E.g.: D2 is reply to D1, so internal destination for D2 is 10.1.0.2
- **Solutions**
 - **Tables**
 - **Multi-address NAT**
 - **NAT with port mapping**

NAT - translation tables

- Needed to route incoming datagram to correct private address
 - All incoming datagrams have *same* destination address
- Based on the following assumption:
 - If (internal) host H sends datagram to (Internet) host A then datagrams coming from H are destined to H
- Works if at most 1 internal host at a time can exchange datagrams with the same global host
 - Remark: no distinction based on transport protocol

NAT tables – example 1/2



Source Dest. host

225.12.3.28	10.1.0.2
225.12.3.25	10.1.0.3

R's translation table

- 10.1.0.2 communicates with 225.12.3.28
- 10.1.0.3 communicates with 225.12.3.25

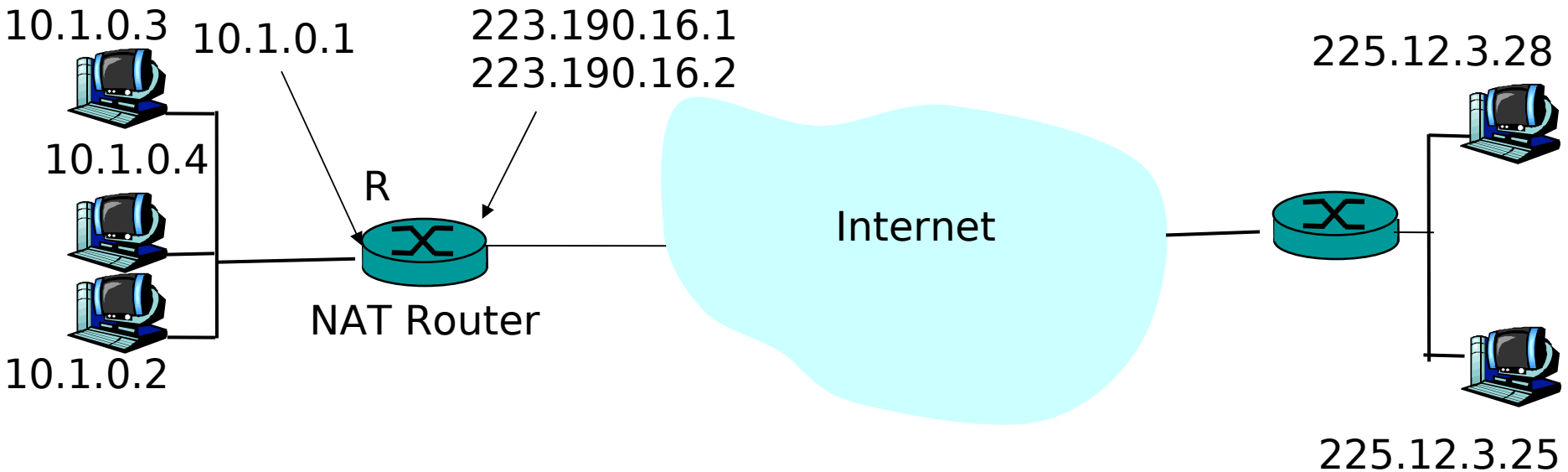
NAT tables – example 2/2

- NAT tables - initialization
 - Manual
 - Outbound datagram: first time that R receives datagram from 10.1.0.2 to 225.12.3.28 it creates entry (225.12.3.28, 10.1.0.2) in its NAT table - most common solution
 - Domain name based initialization
- Main drawback: at most 1 local host can access to the same Internet machine

NAT with multi-addressing

- Idea: use NAT tables with a pool of global IP addresses (instead of one)
 - NAT router's external interface has more than 1 associated global address
 - Global addresses assigned round-robin to outbound datagrams
 - K global addresses allow *at most* K internal hosts to exchange info with the same Internet host

E.g.: K=2



Source	Global dest. IP.	Dest. host
225.12.3.28	223.190.16.1	10.1.0.2
225.12.3.28	223.190.16.2	10.1.0.4
225.12.3.25	223.190.16.1	10.1.0.3

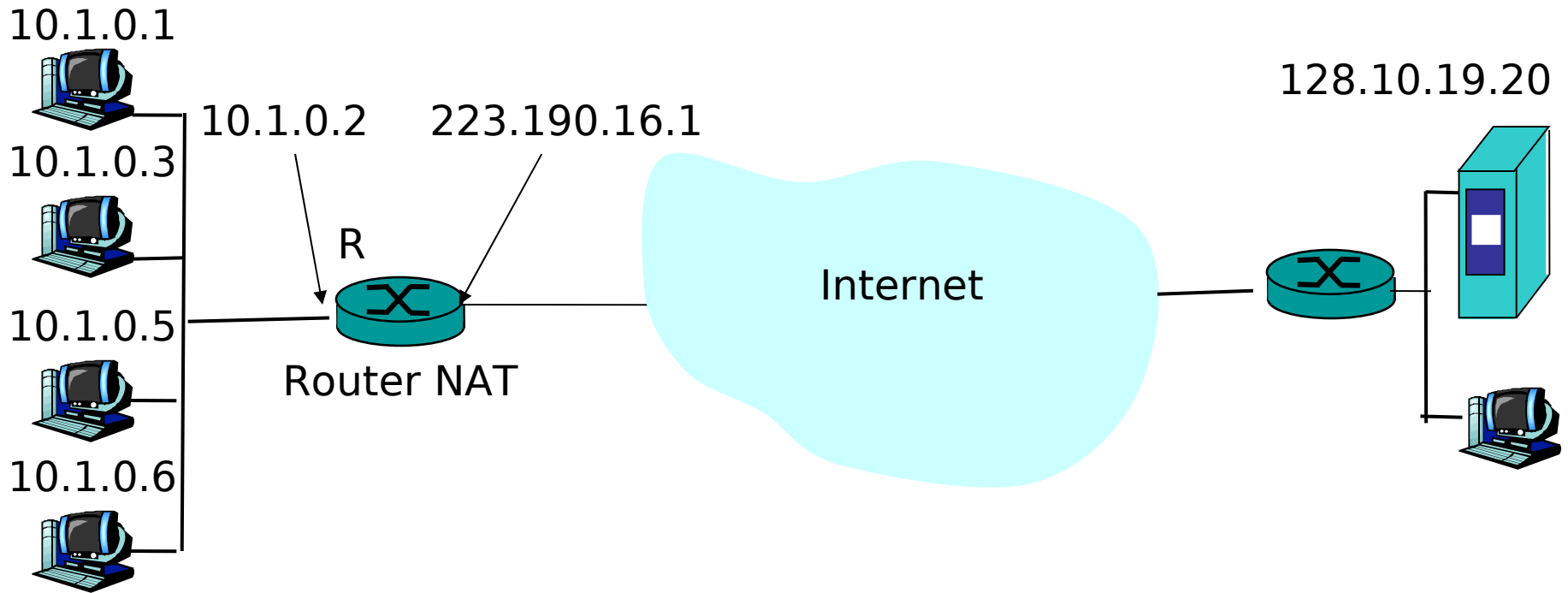
R's translation table

- 10.1.0.2 and 10.0.0.4 communicate with 225.12.3.28
- 10.1.0.3 communicates with 225.12.3.25

NAT with Port mapping

- NAT router distinguishes (internal) destinations of incoming datagrams also on the basis of transport protocol information
- Used when datagrams carry TCP or UDP segments
- It also uses port numbers
- Necessary to extend entry fields of NAT tables
- More processing complexity at NAT routers

Port mapping example 1/3



- 10.1.0.1 e 10.1.0.5 both have ongoing HTTP connection with Web server 128.10.19.20 on port 80
 - 2 TCP connections to 128.10.19.20 on port 80
- Other hosts have ongoing TCP connections to other machines not shown in the picture

Port mapping example 2/3

Local side		Protocol	Remote side		NAT info	
Address	Port		Address	Port	Port	Protocol
10.1.0.5	21023	TCP	128.10.19.20	80	14003	TCP
10.1.0.1	2386	TCP	128.10.19.20	80	14010	TCP
10.1.0.6	26600	UDP	207.200.75.12	21	14012	UDP
10.1.0.3	1274	UDP	128.10.19.20	80	14003	UDP

- NAT table
 - First 2 lines correspond to previous picture
 - Notice *explicit* management of transport protocol

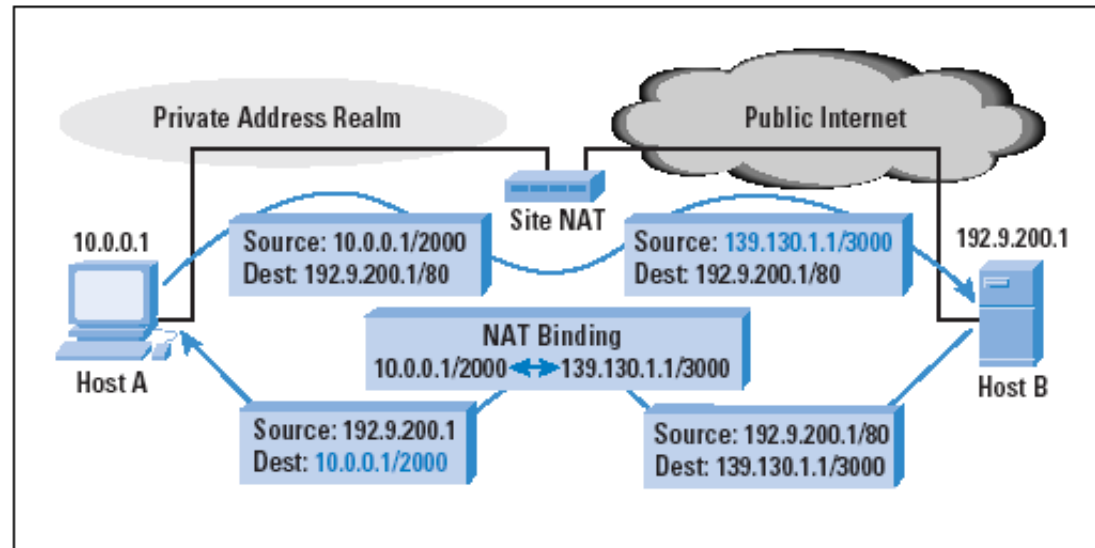
Port mapping example 3/3

- NAT maintains *distinct* NAT port for every information exchange [**binding**]
 - Two local hosts might choose same private port number
 - Host 128.10.19.20 can distinguish the two TCP connections corresponding to the first two lines of the previous table:
 - (**223.190.16.1, 14003**, 128.10.19.20, 80)
 - (**223.190.16.1, 14010**, 128.10.19.20, 80)

Port mapping

- Binding expires according to associated timer
- Up to 65536 sessions for every <external address, protocol> pair

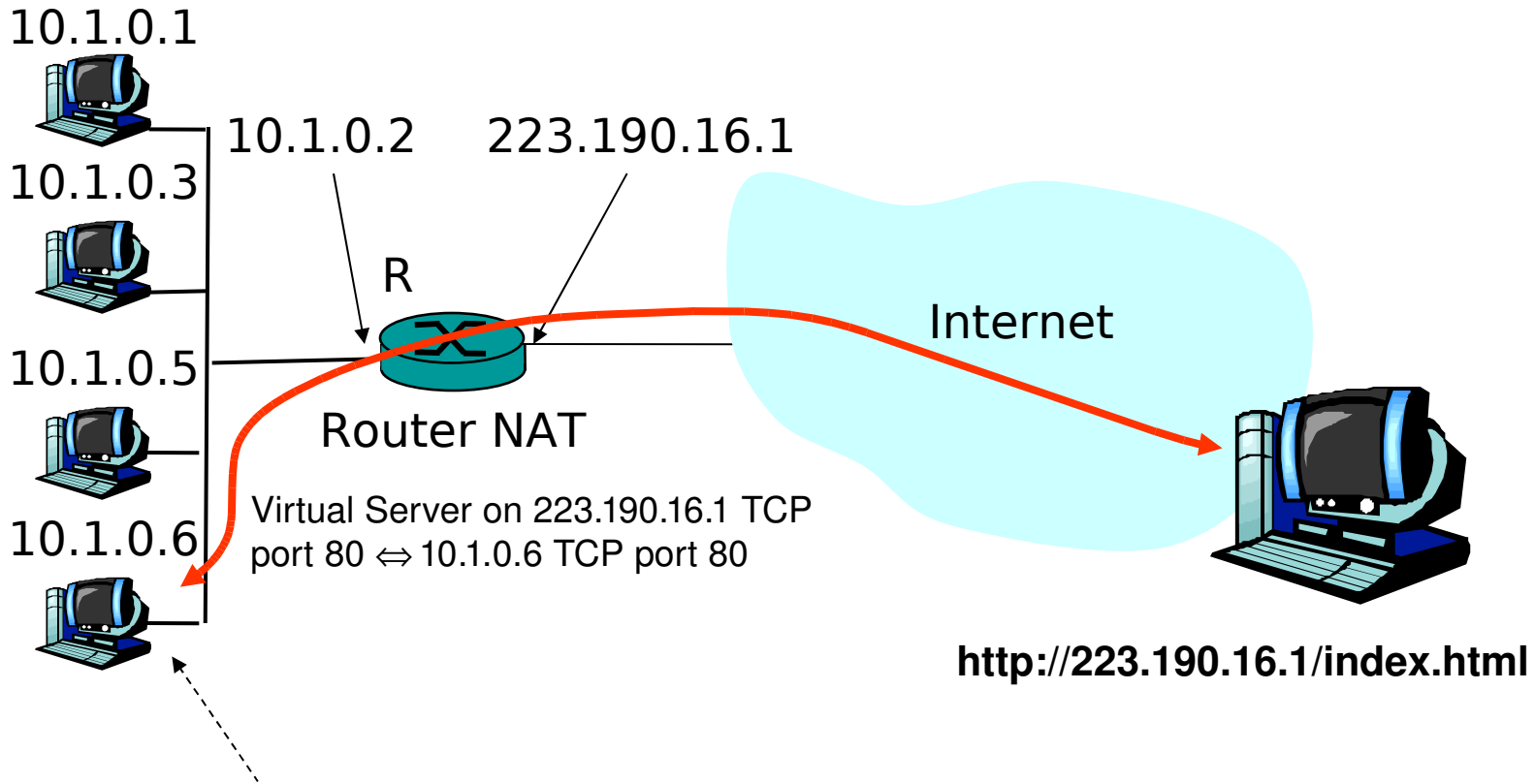
Figure 4: NATP Traversal



Virtual Server

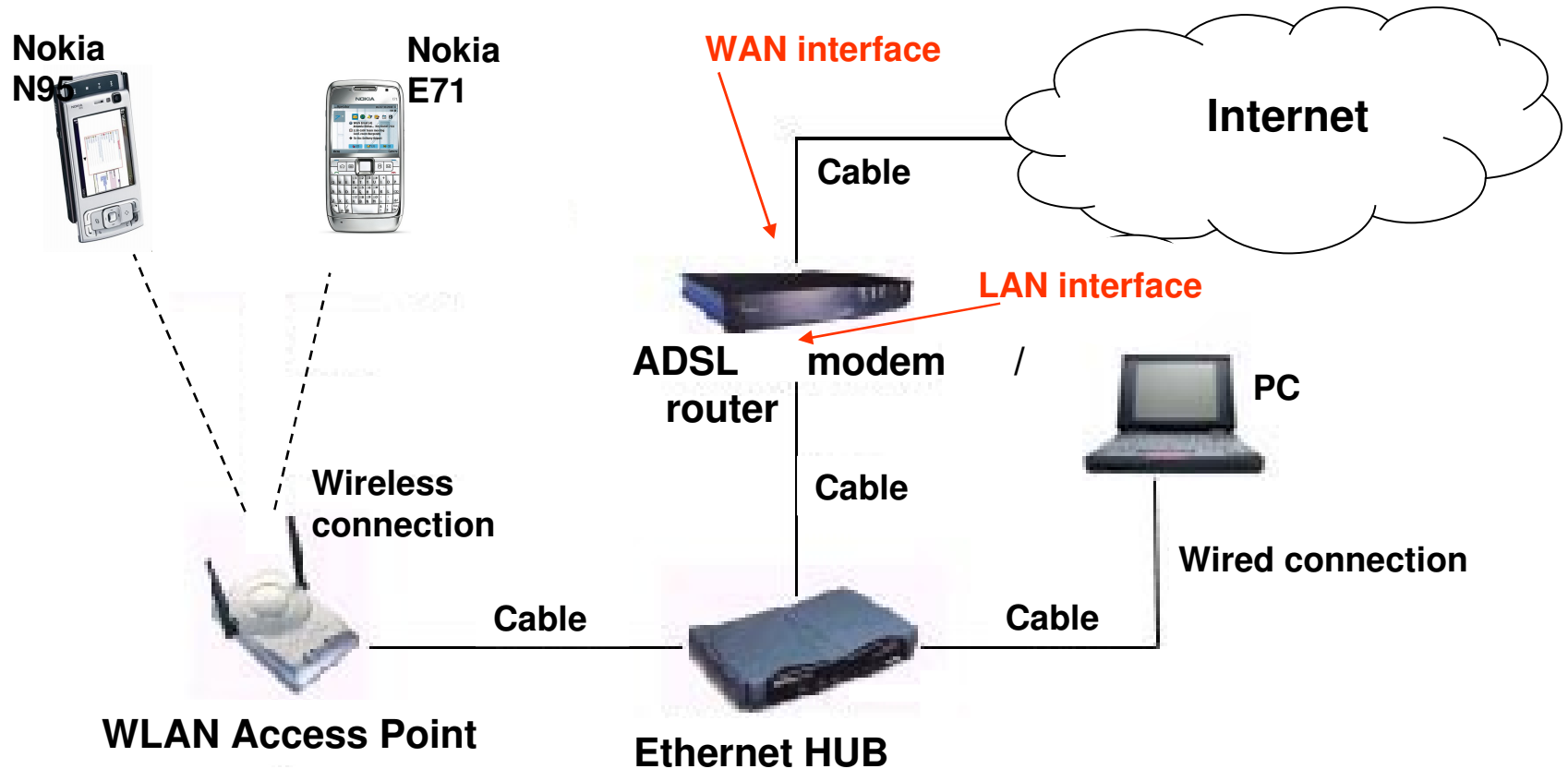
- For a given transport protocol [TCP and/or UDP]: static binding between <global_address,global_port> and <local_address, local_port>
- Allows active services on a host with private address to be reachable over the Internet
 - Examples: Web Server, UDP server,...
- Virtual Server vs dynamic Port mapping
 - Dynamic Port mapping activates binding only AFTER private host initiates transmission and binding is removed when associated timer expires (after binding not used for a time longer than timer)

Virtual Server: example



Web server physically executed on host host 10.1.0.6 and logically on 223.190.16.1

NAT example 1/5: network scenario



NAT example 2/5: router status

USRobotics ADVANCED SETUP

Home Logout

» SETUP WIZARD

SYSTEM

WAN

LAN

NAT

ROUTING

FIREWALL

UPnP

ADSL

DDNS

TOOLS

STATUS

Current Time: 10/22/2008 10:28:34 am

INTERNET

ADSL: CONNECTED

WAN IP: 151.49.111.3

Subnet Mask: 255.255.0.0

Gateway: 151.6.146.71

Primary DNS: 193.70.152.15

Secondary DNS: 193.70.152.25

GATEWAY

IP Address: 192.168.2.1

Subnet Mask: 255.255.255.0

DHCP Server: Enabled

Firewall: Disabled

UPnP: Enabled

INFORMATION

Numbers of DHCP Clients: 2

Runtime Code Version: 0.01.004 (May 15 2006 10:15:30)

Boot Code Version: 0.73.00

ADSL Modem Code Version: 06.00.04.00A

LAN MAC Address: 00-14-C1-21-E5-C6

WAN MAC Address: 00-14-C1-21-E5-C7

Hardware Version: 01

Serial Num: 1WCGE8GG4346

ATM PVC

VC1	
VPI/VC1	8/35
Encapsulation	VC MUX
Protocol	PPPoA
IP Address	151.49.111.3
Subnet Mask	255.255.0.0
Gateway	151.6.146.71
Primary DNS	193.70.152.15
Secondary DNS	193.70.152.25
<input type="button" value="Disconnect"/> <input type="button" value="Connect"/>	

VC2	
Disabled	

Completato

start | Esercitazioni | Microsoft PowerPoint ... | Mozilla Firefox | 20.29

NAT example 3/5: router NAT service

Mozilla Firefox

File Modifica Visualizza Cronologia Segnalibri Yahoo! Strumenti ?

http://192.168.2.1/index.stm

Google

Più visitati Come iniziare Ultime notizie

Y! Search Mail Answers Dating Mobile Sign in

USRobotics **ADVANCED SETUP**

Home Logout

» SETUP WIZARD

SYSTEM

WAN

LAN

NAT

» Address Mapping

» Virtual Server

» Special Application

» NAT Mapping Table

ROUTING

FIREWALL

UPnP

ADSL

DDNS

TOOLS

STATUS

NAT Settings

Network Address Translation (NAT) allows multiple users at your local site to access the Internet through a single public IP address or multiple public IP addresses. NAT can also prevent hacker attacks by mapping local addresses to public addresses for key services such as the Web or FTP.

Enable or disable NAT: Enable Disable

SAVE SETTINGS

Completato

start Esercitazioni Microsoft PowerPoint ... Mozilla Firefox 20:25

NAT example 4/5: router NAT mapping table

The screenshot shows a Mozilla Firefox browser window displaying the USRobotics router configuration page. The page title is "NAT Mapping Table" and it is part of the "ADVANCED SETUP" section. The browser's address bar shows the URL "http://192.168.2.1/index.stm". The page content includes a navigation menu on the left with options like "SETUP WIZARD", "SYSTEM", "WAN", "LAN", "NAT", "ROUTING", "FIREWALL", "UPnP", "ADSL", "DDNS", "TOOLS", and "STATUS". The "NAT" section is expanded, showing "NAT Mapping Table" as the selected option. The main content area displays a table of NAT mappings and a "Refresh" button.

USRobotics ADVANCED SETUP

[Home](#) [Logout](#)

>> SETUP WIZARD **NAT Mapping Table**

The NAT Mapping Table displays the current NAT address mappings.

Index	Protocol	Local IP	Local Port	Pseudo IP	Pseudo Port	Peer IP	Peer Port
1	TCP	192.168.2.2	4318	151.49.111.3	4318	63.245.209.121	80
2	TCP	192.168.2.2	4375	151.49.111.3	4375	74.125.97.39	80

Page: 1/1

Completato

start | Esercitazioni | Microsoft PowerPoint ... | Mozilla Firefox | 20.26

NAT example 5/5: router virtual servers

USRobotics ADVANCED SETUP

Home Logout

Virtual Server

You can configure the router as a virtual server. This will allow remote users to access the Web or FTP at your local site via public IP addresses. These remote users will be automatically redirected to local servers that are configured with private IP addresses. In other words, depending on the requested service (TCP/UDP port number), the router redirects the external service request to the appropriate server (located at another internal IP address). This page supports port ranges, multiple ports, and combinations of both.

For example:

- Port Ranges: ex. 100-150
- Multiple Ports: ex. 25,110,80
- Combination: ex. 25-100,80

No.	LAN IP Address	Protocol Type	LAN Port	Public Port	Enable	Add	Clean
1	192.168.2.5	TCP&UDP	8080	8080	<input checked="" type="checkbox"/>	Add	Clean
2	192.168.2.6	TCP	7000	8000	<input checked="" type="checkbox"/>	Add	Clean
3	192.168.2.6	UDP	7000	8000	<input checked="" type="checkbox"/>	Add	Clean
4	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
5	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
6	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
7	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
8	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
9	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
10	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean
11	192.168.2.	TCP			<input type="checkbox"/>	Add	Clean

Completato Un download in corso (42 minuti al termine)

NAT behaviour

- Many differences among proposed solutions
 - Even between different solutions by *same* vendor
- Some applications require info about “client IP address” [see example further]
 - What happens with NAT? Often it is necessary to understand which binding scheme is used
 - Specific protocol to this purpose: STUN [Simple traversal of UDP over NATs];
 - STUN classifies binding strategies as follows:
 - Symmetric
 - Full-cone
 - Address restricted-cone
 - Port-restricted-cone
 - In practice: NAT devices can actually combine different strategies :-)

Symmetric NAT

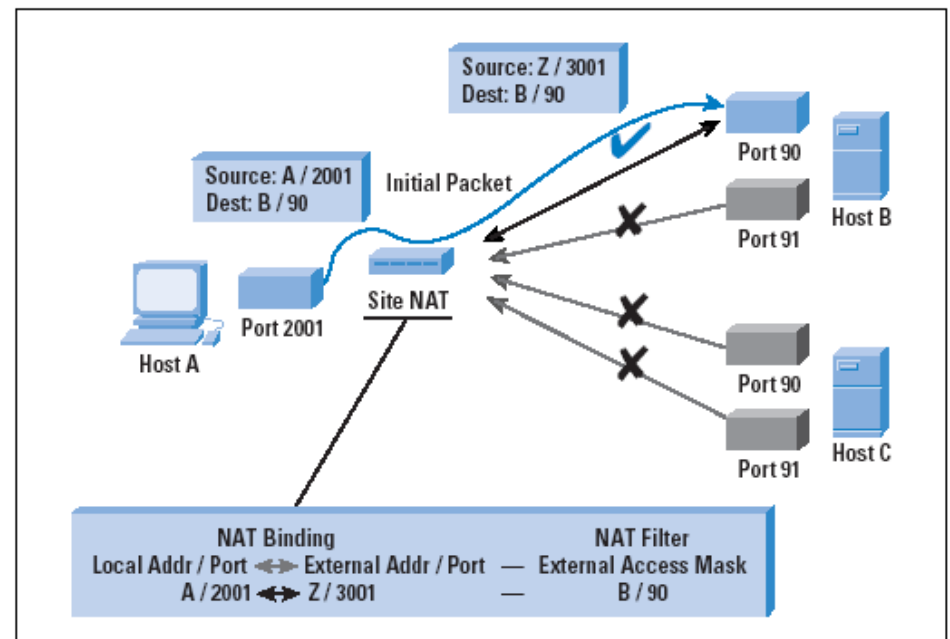
- Mapping determined by destination address/port
- Incoming (external) packets filtered according to source address/port
- Only filtered packets are allowed

Figure 5: Symmetric NAT

Each request from the same internal IP address and port to a specific destination IP address and port is mapped to a unique external source IP address and port.

If the same internal host sends a packet even with the same source address and port but to a different destination, a different mapping is used.

Only an external host that receives a packet from an internal host can send a packet back



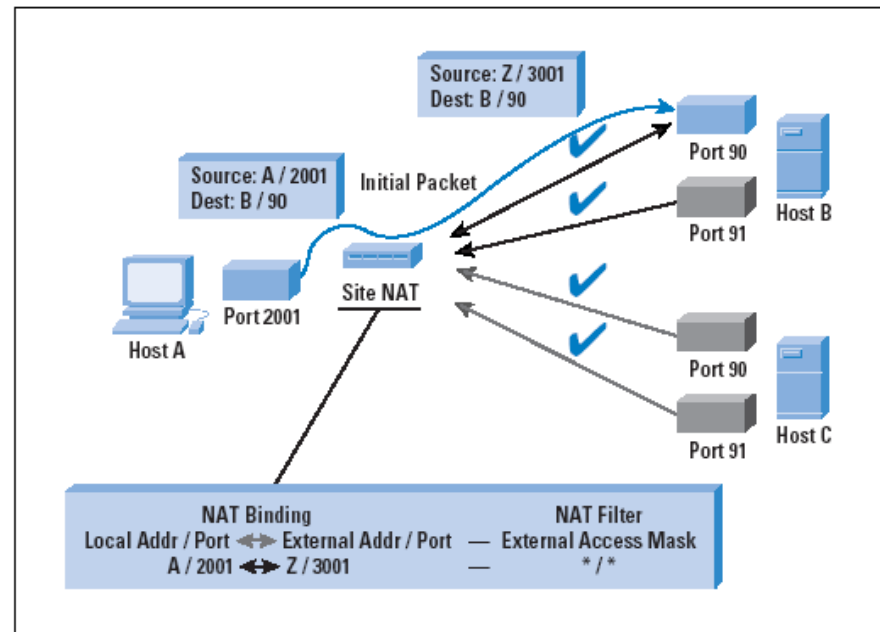
Full-cone NAT

- Also known as NAT 1:1
- Incoming (external) packets are allowed whenever destination address/port pair is currently used

Figure 6: Full Cone NAT

Once an internal address (iAddr:port1) is mapped to an external address (eAddr:port2), any packets from iAddr:port1 will be sent through eAddr:port2.

Any external host can send packets to iAddr:port1 by sending packets to eAddr:port2.



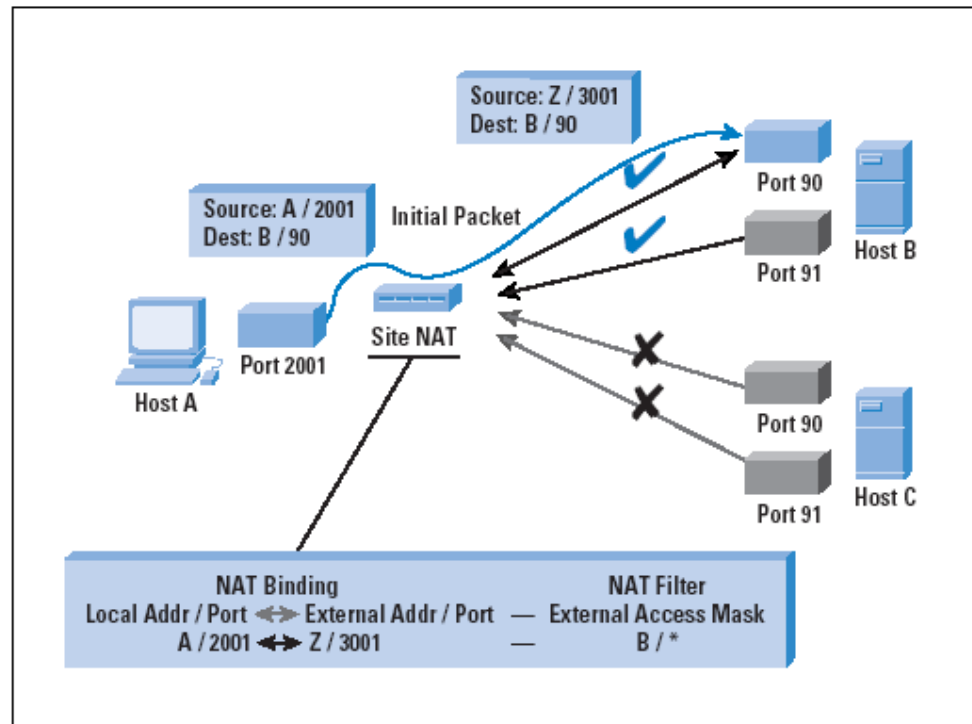
Address-restricted-cone or Restricted-cone

- Incoming (external) packets allowed whenever destination address/port is currently used and source address is associated to the pair in NAT binding

Figure 7: Restricted-Cone NAT

Once an internal address (iAddr:port1) is mapped to an external address (eAddr:port2), any packets from iAddr:port1 will be sent through eAddr:port2.

An external host (hostAddr:any) can send packets to iAddr:port1 by sending packets to eAddr:port2 only if iAddr:port1 had previously sent a packet to hostAddr:any; "any" means that port number doesn't matter



Port-restricted-cone

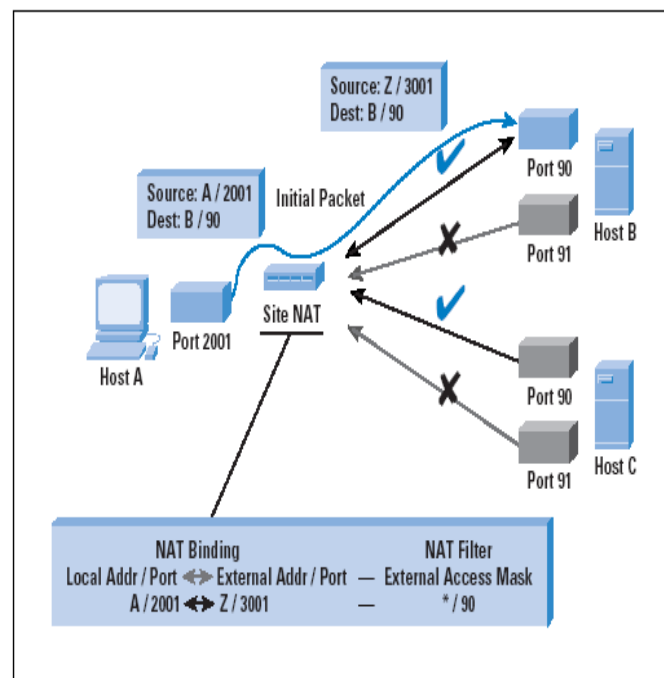
- Incoming (external) packets allowed whenever destination address/port is currently in use and source address is the one associated to the pair under consideration in NAT binding

Like a restricted cone NAT, but the restriction includes port numbers.

Once an internal address (iAddr:port1) is mapped to an external address (eAddr:port2), any packets from iAddr:port1 will be sent through eAddr:port2.

An external host (hostAddr:port3) can send packets to iAddr:port1 by sending packets to eAddr:port2 only if iAddr:port1 had previously sent a packet to hostAddr:port3.

Figure 8: Port-Restricted-Cone NAT



Any external host can send packets to internal_address / internal_port by sending packets to external_addr / external_port

Interaction with other protocols

- NAT interacts with upper (layer) protocols and ICMP
- NAT has to:
 - Modify IP headers
 - Modify TCP and UDP port numbers
 - Recompute checksums (because headers change)
 - Explicitely manage ICMP messages (e.g., ping)
- Which impact at the application level?
 - E.g., FTP, SIP,...

Interactions with other protocols/cont.

- Some applications need “client IP address” info [see example further]
 - What happens with NAT? which $\langle \text{IP}, \text{port} \rangle$ info does client communicate to application’s server side?
 - If private info communicated --> client could be unreachable
 - If global info communicated [NAT] it needs to know it

Interaction with other protocols

VoIP / SIP Example 1/13

- scenario:
 - VoIP / SIP softphone [\Leftrightarrow VoIP client] registers for the services of a SIP *Registrar* server
 - SIP / UDP / IP
 - Softphone uses UDP port 25992 for SIP signalling
 - Registrar SIP uses UDP port 5061 for SIP signalling
 - SIP registration message [REGISTER] contains header *Contact:* which provides to the *registrar* client's reachability info in the case of incoming calls: < client_ip_address, client_UDP_port >
 - Upon a phone call, client must communicate to SIP Proxy which <ip_address, UDP_port > it is going to use for RTP audio session

Interaction with other protocols

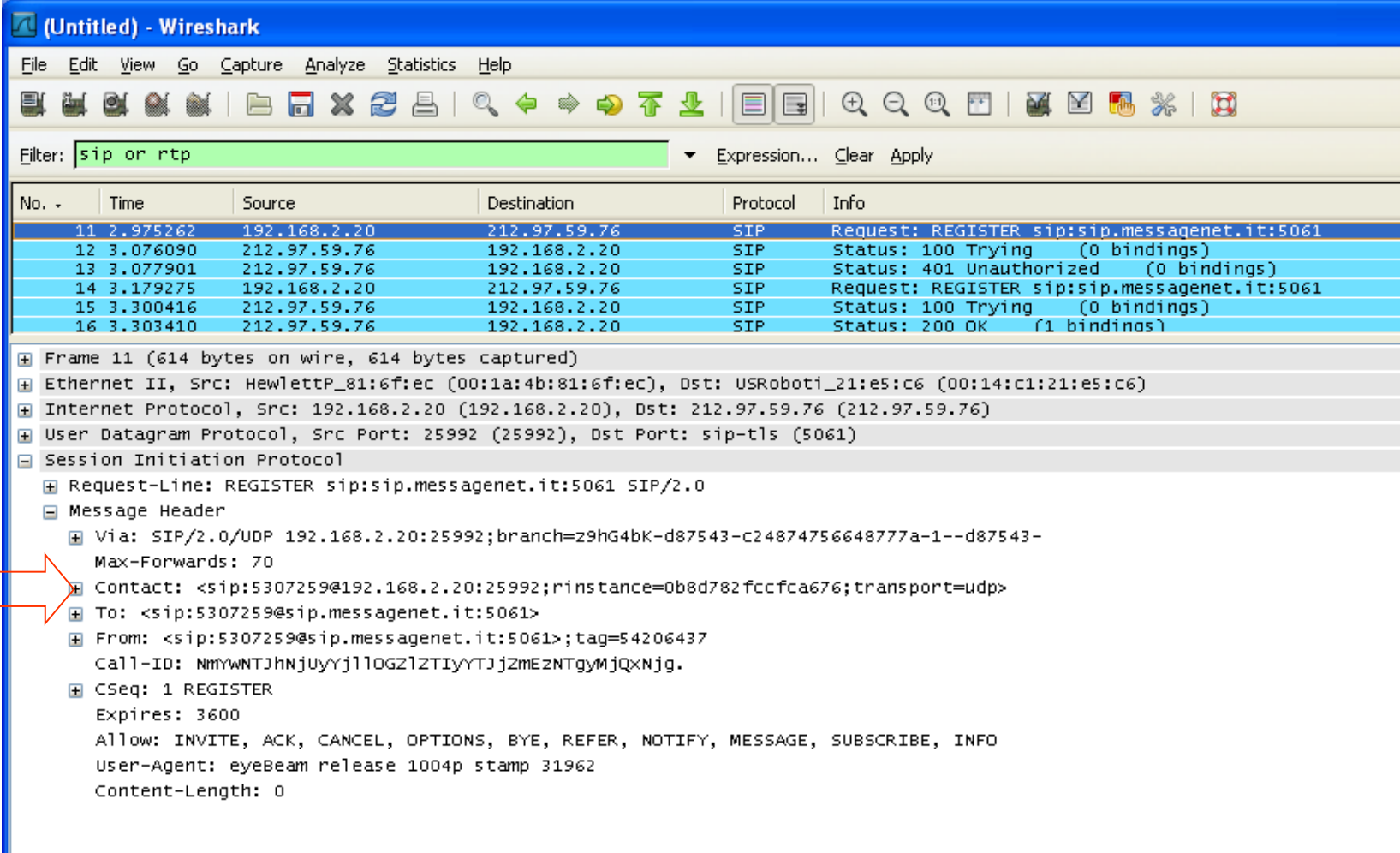
VoIP / SIP Example 2/13

- I 2 aspetti evidenziati rientrano nelle problematiche di gestione del **NAT traversal**:
- 2 Alternative:
 - Gestione lato server:
 - Il client comunica al registrar ed al proxy il proprio indirizzo privato ed i numeri di porta UDP locali
 - il registrar ignora le info di raggiungibilità comunicate dal client e le recupera invece dal pacchetto UDP ricevuto [external_address, external_port]
 - Il proxy deve aspettare il primo pacchetto RTP inviato dal client per conoscere su quale coppia <IP,UDP_port> il NAT ha mappato l'indirizzo IP e la porta UDP per il traffico RTP del client
 - Gestione lato client:
 - Il client utilizza il protocollo STUN [o equivalente] per conoscere su quali coppie <IP,UDP_port> sono mappate:
 - » la sua porta SIP locale; questo mapping deve restare invariato nel tempo ⇒ **necessario refresh del binding in assenza di traffico**
 - » la porta UDP per la sessione RTP; questo mapping è in generale diverso nel tempo

Interaction with other protocols

VoIP / SIP Example 3/13

Server side management:



The image shows a Wireshark capture of SIP traffic. The filter is set to 'sip or rtp'. The packet list shows a sequence of SIP messages: a REGISTER request, a 100 Trying status, a 401 Unauthorized status, another REGISTER request, a 100 Trying status, and finally a 200 OK status. The packet details for the first REGISTER request (Frame 11) are expanded, showing the SIP header fields: Request-Line, Message Header, Via, Contact, To, From, Call-ID, CSeq, Expires, Allow, User-Agent, and Content-Length.

No. -	Time	Source	Destination	Protocol	Info
11	2.975262	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
12	3.076090	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
13	3.077901	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
14	3.179275	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
15	3.300416	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
16	3.303410	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)

Frame 11 (614 bytes on wire, 614 bytes captured)

- Ethernet II, Src: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec), Dst: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6)
- Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 212.97.59.76 (212.97.59.76)
- User Datagram Protocol, Src Port: 25992 (25992), Dst Port: sip-tls (5061)
- Session Initiation Protocol
 - Request-Line: REGISTER sip:sip.messagenet.it:5061 SIP/2.0
 - Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:25992;branch=z9hG4bK-d87543-c24874756648777a-1--d87543--Max-Forwards: 70
 - Contact: <sip:5307259@192.168.2.20:25992;rinstance=0b8d782fccfca676;transport=udp>
 - To: <sip:5307259@sip.messagenet.it:5061>
 - From: <sip:5307259@sip.messagenet.it:5061>;tag=54206437
 - Call-ID: NmyWNTJhNjuyYjll0GZlZTIyYTJjZmEzNTgyMjQxNjg.
 - CSeq: 1 REGISTER
 - Expires: 3600
 - Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
 - User-Agent: eyeBeam release 1004p stamp 31962
 - Content-Length: 0

Interaction with other protocols

VoIP / SIP Example 4/13

Server side management:

The image shows a Wireshark capture of SIP traffic. The filter is set to 'sip or rtp'. The packet list shows a sequence of messages: a REGISTER request, a 100 Trying response, a 401 Unauthorized response, another REGISTER request, another 100 Trying response, and finally a 200 OK response. The details pane for the selected 200 OK packet shows the following fields:

- Frame 12 (610 bytes on wire, 610 bytes captured)
- Ethernet II, Src: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6), Dst: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec)
- Internet Protocol, Src: 212.97.59.76 (212.97.59.76), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: sip-tls (5061), Dst Port: 25992 (25992)
- Session Initiation Protocol
 - Status-Line: SIP/2.0 100 Trying
 - Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:25992;branch=z9hG4bK-d87543-c24874756648777a-1--d87543-;rport=25992;received=151.49.83.75
 - To: <sip:5307259@sip.messagenet.it:5061>
 - From: <sip:5307259@sip.messagenet.it:5061>;tag=54206437
 - Call-ID: NmywNTJhNjuyYjllOGZlZTIyYTJjZmEzNTgyMjQxNjg.
 - CSeq: 1 REGISTER
 - Server: OpenSer (1.1.0-notls (i386/linux))
 - Content-Length: 0
 - Warning: 392 212.97.59.76:5061 "Noisy feedback tells: pid=15603 req_src_ip=151.49.83.75 req_src_port=25992 in_uri=sip:si

Interaction with other protocols

VoIP / SIP Example 5/13

Server side management:

(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: sip or rtp

No. -	Time	Source	Destination	Protocol	Info
67	15.232212	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messagenet.it:5061, with session description
68	15.364931	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
69	15.373992	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description
71	15.397321	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B04D4E0, Seq=5188, Time=1130300, Mark
72	15.403205	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6869, Time=480
73	15.417495	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B04D4E0, Seq=5189, Time=1130460

Ethernet II, Src: Hewlett-Packard (00:1d:60:81:01:01), Dst: USB0001_21:es:L6 (00:14:c1:21:es:L6)

Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 212.97.59.76 (212.97.59.76)

User Datagram Protocol, Src Port: 25992 (25992), Dst Port: sip-tls (5061)

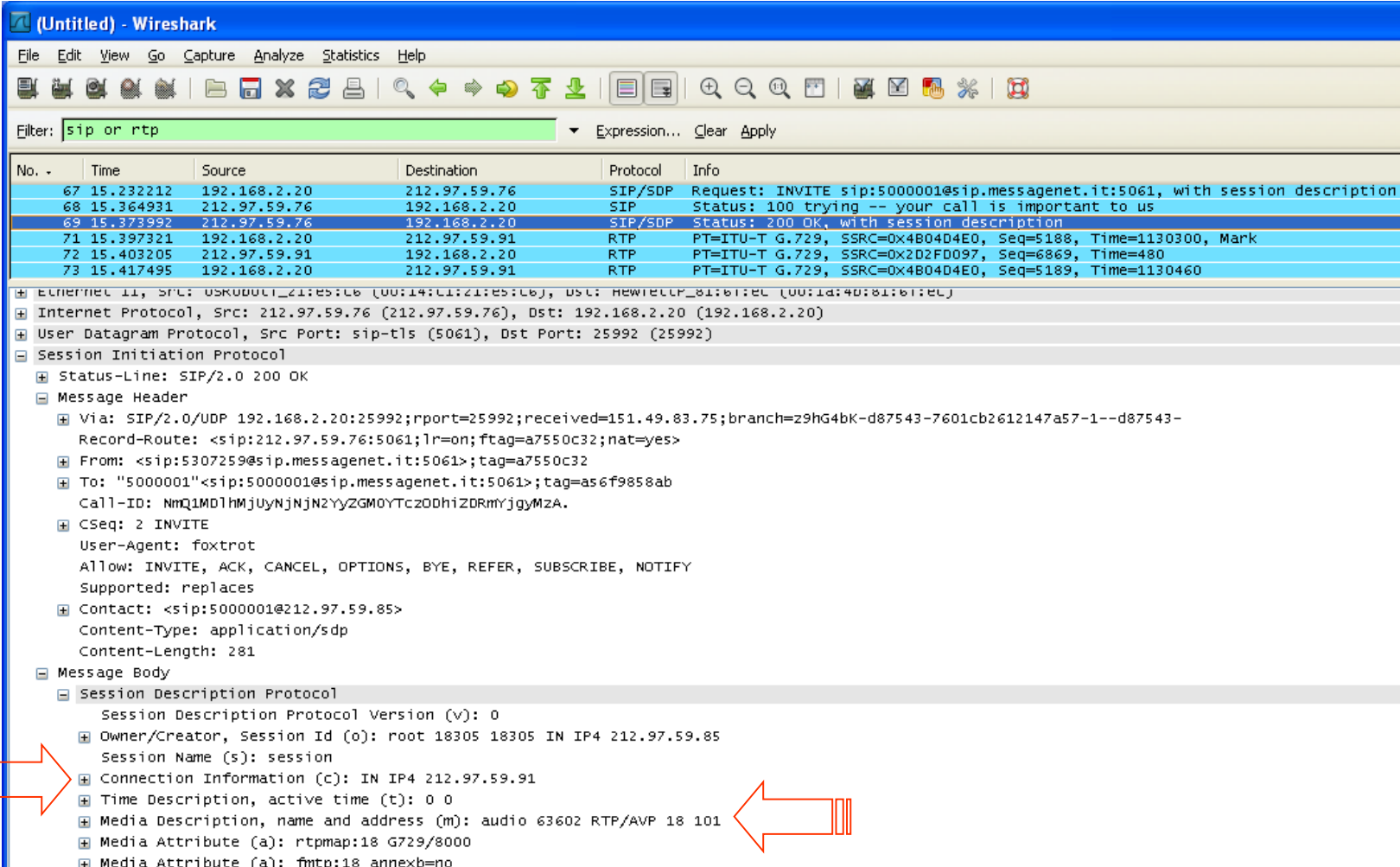
Session Initiation Protocol

- Request-Line: INVITE sip:5000001@sip.messagenet.it:5061 SIP/2.0
- Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:25992;branch=z9hG4bK-d87543-7601cb2612147a57-1--d87543-Max-Forwards: 70
 - Contact: <sip:5307259@192.168.2.20:25992;transport=udp>
 - To: "5000001"<sip:5000001@sip.messagenet.it:5061>
 - From: <sip:5307259@sip.messagenet.it:5061>;tag=a7550c32
 - Call-ID: NmQ1MD1hmJuyNjNjN2YyZGM0YTczODhiZDRmYjgyMZA.
 - CSeq: 2 INVITE
 - Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
 - Content-Type: application/sdp
 - Proxy-Authorization: Digest username="5307259", realm="sip.messagenet.it", nonce="492ad320a6fbd3bf6e2344853aaf7acd2a46547e", uri="sip:5000001@sip.r
 - User-Agent: eyeBeam release 1004p stamp 31962
 - Content-Length: 259
- Message Body
 - Session Description Protocol
 - Session Description Protocol Version (v): 0
 - Owner/Creator, Session Id (o): - 8 2 IN IP4 192.168.2.20
 - Session Name (s): CounterPath eyeBeam 1.5
 - Connection Information (c): IN IP4 192.168.2.20
 - Time Description, active time (t): 0 0
 - Media Description, name and address (m): audio 43740 RTP/AVP 18 101
 - Media Attribute (a): fmtp:18 annexb=no

Interaction with other protocols

VoIP / SIP Example 6/13

Server side management:



The image shows a Wireshark capture of network traffic. The filter is set to 'sip or rtp'. The packet list shows several packets related to a SIP call and RTP media. The packet details pane is expanded to show the Session Description Protocol (SDP) information for the selected packet (No. 73).

No.	Time	Source	Destination	Protocol	Info
67	15.232212	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messagenet.it:5061, with session description
68	15.364931	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
69	15.373992	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description
71	15.397321	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B04D4E0, Seq=5188, Time=1130300, Mark
72	15.403205	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6869, Time=480
73	15.417495	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B04D4E0, Seq=5189, Time=1130460

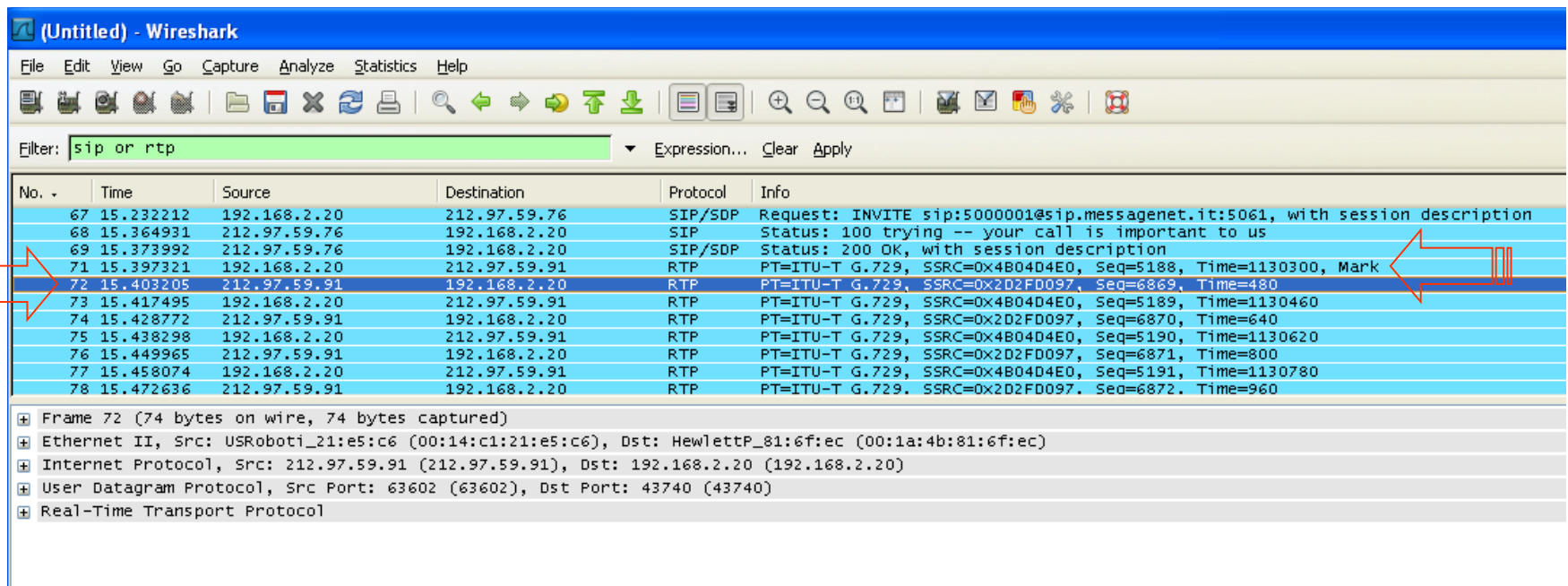
Packet 73 details:

- Ethernet II, Src: USK0001_21:es:c6 (00:14:c1:21:es:c6), Dst: Hewlett_81:61:ec (00:14:40:81:61:ec)
- Internet Protocol, Src: 212.97.59.76 (212.97.59.76), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: sip-tls (5061), Dst Port: 25992 (25992)
- Session Initiation Protocol
 - Status-Line: SIP/2.0 200 OK
 - Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:25992;rport=25992;received=151.49.83.75;branch=29HG4bk-d87543-7601cb2612147a57-1--d87543-Record-Route: <sip:212.97.59.76:5061;lr=on;ftag=a7550c32;nat=yes>
 - From: <sip:5307259@sip.messagenet.it:5061>;tag=a7550c32
 - To: "5000001"<sip:5000001@sip.messagenet.it:5061>;tag=as6f9858ab
 - Call-ID: NmQ1MD1hMjUyNjNjN2YyZGM0YTczODhiZDRmYjgyMzA.
 - CSeq: 2 INVITE
 - User-Agent: foxtrot
 - Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, SUBSCRIBE, NOTIFY
 - Supported: replaces
 - Contact: <sip:5000001@212.97.59.85>
 - Content-Type: application/sdp
 - Content-Length: 281
 - Message Body
 - Session Description Protocol
 - Session Description Protocol Version (v): 0
 - Owner/Creator, Session Id (o): root 18305 18305 IN IP4 212.97.59.85
 - Session Name (s): session
 - Connection Information (c): IN IP4 212.97.59.91
 - Time Description, active time (t): 0 0
 - Media Description, name and address (m): audio 63602 RTP/AVP 18 101
 - Media Attribute (a): rtpmap:18 G729/8000
 - Media Attribute (a): fmtp:18 annexb=no

Interaction with other protocols

VoIP / SIP Example 7/13

Server side management:



The image shows a Wireshark capture of a VoIP session. The filter is set to 'sip or rtp'. The capture shows a sequence of packets: SIP INVITE, SIP 100 trying, SIP 200 OK, and several RTP packets. Frame 72 is highlighted in blue and has two red arrows pointing to it from the left and right sides. The packet list table is as follows:

No. -	Time	Source	Destination	Protocol	Info
67	15.232212	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messagenet.it:5061, with session description
68	15.364931	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
69	15.373992	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description
71	15.397321	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B0404E0, Seq=5188, Time=1130300, Mark
72	15.403205	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6869, Time=480
73	15.417495	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B0404E0, Seq=5189, Time=1130460
74	15.428772	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6870, Time=640
75	15.438298	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B0404E0, Seq=5190, Time=1130620
76	15.449965	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6871, Time=800
77	15.458074	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x4B0404E0, Seq=5191, Time=1130780
78	15.472636	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x2D2FD097, Seq=6872, Time=960

Below the packet list, the details of Frame 72 are shown:

- Frame 72 (74 bytes on wire, 74 bytes captured)
- Ethernet II, Src: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6), Dst: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec)
- Internet Protocol, Src: 212.97.59.91 (212.97.59.91), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: 63602 (63602), Dst Port: 43740 (43740)
- Real-Time Transport Protocol

Interaction with other protocols

VoIP / SIP Example 8/13

Client side management:

The screenshot shows a Wireshark capture of network traffic. The filter is set to 'stun or sip or rtp'. The packet list shows a sequence of STUN and SIP messages. The packet details pane is expanded to show the structure of a STUN Binding Request.

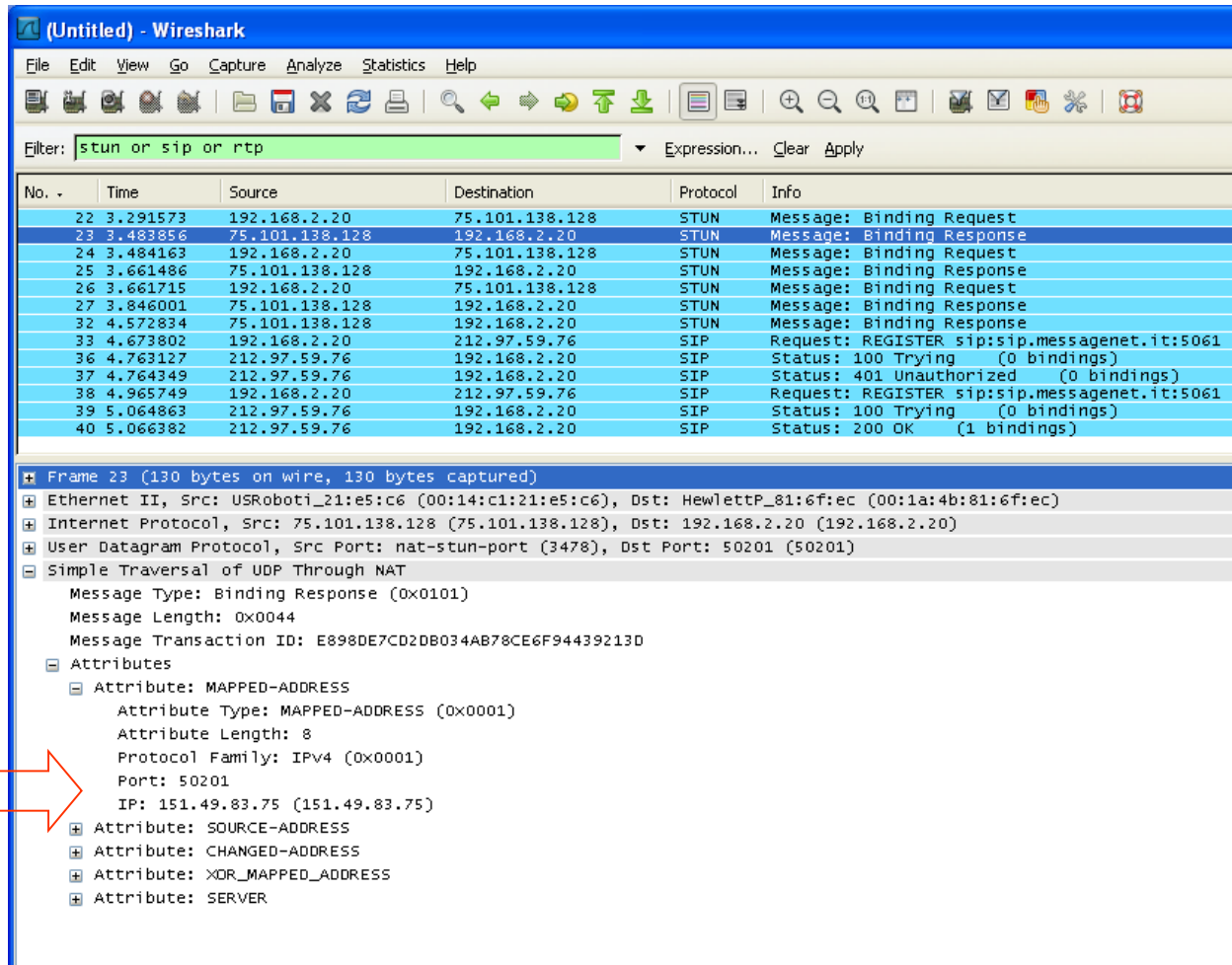
No. -	Time	Source	Destination	Protocol	Info
22	3.291573	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
23	3.483856	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
24	3.484163	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
25	3.661486	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
26	3.661715	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
27	3.846001	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
32	4.572834	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
33	4.673802	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
36	4.763127	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
37	4.764349	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
38	4.965749	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
39	5.064863	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
40	5.066382	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)

Frame 22 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec), Dst: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6)
Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 75.101.138.128 (75.101.138.128)
User Datagram Protocol, Src Port: 50201 (50201), Dst Port: nat-stun-port (3478)
Simple Traversal of UDP Through NAT
Message Type: Binding Request (0x0001)
Message Length: 0x0000
Message Transaction ID: E898DE7CD2DB034AB78CE6F94439213D

Interaction with other protocols

VoIP / SIP Example 9/13

Client side management:



Wireshark capture showing STUN and SIP messages. The filter is set to `stun or sip or rtp`. The packet list shows a sequence of STUN Binding Requests and Responses, followed by SIP REGISTER requests and responses.

No.	Time	Source	Destination	Protocol	Info
22	3.291573	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
23	3.483856	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
24	3.484163	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
25	3.661486	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
26	3.661715	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
27	3.846001	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
32	4.572834	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
33	4.673802	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.message.net.it:5061
36	4.763127	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
37	4.764349	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
38	4.965749	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.message.net.it:5061
39	5.064863	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
40	5.066382	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)

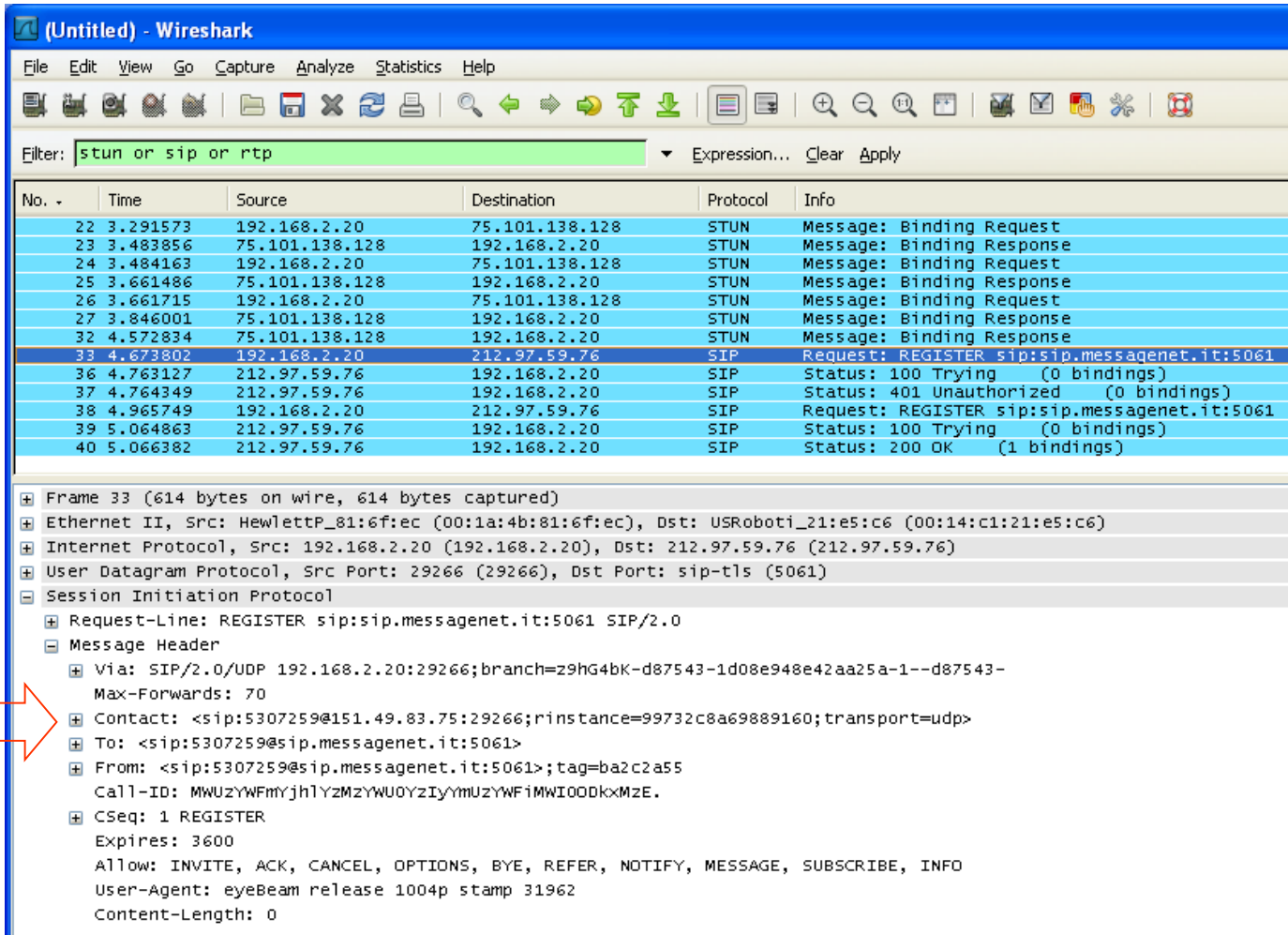
Frame 23 (130 bytes on wire, 130 bytes captured)

- Ethernet II, Src: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6), Dst: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec)
- Internet Protocol, Src: 75.101.138.128 (75.101.138.128), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: nat-stun-port (3478), Dst Port: 50201 (50201)
- Simple Traversal of UDP Through NAT
 - Message Type: Binding Response (0x0101)
 - Message Length: 0x0044
 - Message Transaction ID: E898DE7CD2DB034AB78CE6F94439213D
 - Attributes
 - Attribute: MAPPED-ADDRESS
 - Attribute Type: MAPPED-ADDRESS (0x0001)
 - Attribute Length: 8
 - Protocol Family: IPv4 (0x0001)
 - Port: 50201
 - IP: 151.49.83.75 (151.49.83.75)
 - Attribute: SOURCE-ADDRESS
 - Attribute: CHANGED-ADDRESS
 - Attribute: XOR_MAPPED_ADDRESS
 - Attribute: SERVER

Interaction with other protocols

VoIP / SIP Example 10/13

Client side management:



(Untitled) - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `stun or sip or rtp` Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
22	3.291573	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
23	3.483856	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
24	3.484163	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
25	3.661486	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
26	3.661715	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
27	3.846001	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
32	4.572834	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
33	4.673802	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
36	4.763127	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
37	4.764349	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
38	4.965749	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
39	5.064863	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
40	5.066382	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)

Frame 33 (614 bytes on wire, 614 bytes captured)

- Ethernet II, Src: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec), Dst: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6)
- Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 212.97.59.76 (212.97.59.76)
- User Datagram Protocol, Src Port: 29266 (29266), Dst Port: sip-tls (5061)
- Session Initiation Protocol
 - Request-Line: REGISTER sip:sip.messagenet.it:5061 SIP/2.0
 - Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:29266;branch=z9hG4bK-d87543-1d08e948e42aa25a-1--d87543-
Max-Forwards: 70
 - Contact: <sip:5307259@151.49.83.75:29266;rinstance=99732c8a69889160;transport=udp>
 - To: <sip:5307259@sip.messagenet.it:5061>
 - From: <sip:5307259@sip.messagenet.it:5061>;tag=ba2c2a55
Call-ID: MWUzYWFmYjhlYzZmZyYU0yZiYmUzYWFmIWI0ODkxMzE.
 - CSeq: 1 REGISTER
Expires: 3600
Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
User-Agent: eyeBeam release 1004p stamp 31962
Content-Length: 0

Interaction with other protocols

VoIP / SIP Example 11/13

Client side management:

Intel(R) 82562GT 10/100 Network Connection (Microsoft's Packet Scheduler) : Capturing - Wireshark

Filter: `stun or sip or rtp`

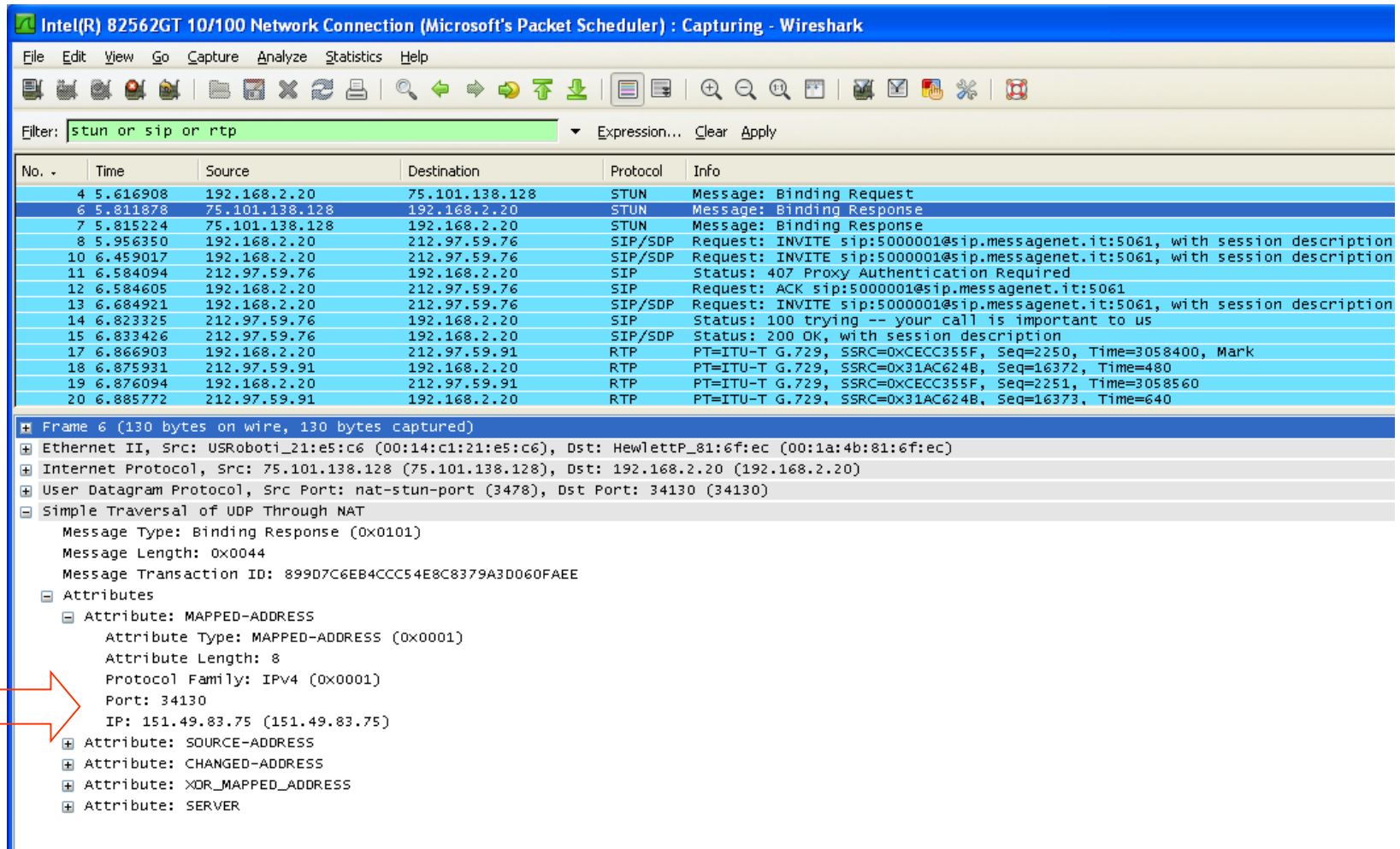
No. -	Time	Source	Destination	Protocol	Info
4	5.616908	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
6	5.811878	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
7	5.815224	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
8	5.956350	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.message.net.it:5061, with session description
10	6.459017	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.message.net.it:5061, with session description
11	6.584094	212.97.59.76	192.168.2.20	SIP	Status: 407 Proxy Authentication Required
12	6.584605	192.168.2.20	212.97.59.76	SIP	Request: ACK sip:5000001@sip.message.net.it:5061
13	6.684921	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.message.net.it:5061, with session description
14	6.823325	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
15	6.833426	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description
17	6.866903	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=2250, Time=3058400, Mark
18	6.875931	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=16372, Time=480
19	6.876094	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=2251, Time=3058560
20	6.885772	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=16373, Time=640

Frame 4 (62 bytes on wire, 62 bytes captured)
Ethernet II, Src: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec), Dst: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6)
Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 75.101.138.128 (75.101.138.128)
User Datagram Protocol, Src Port: 34130 (34130), Dst Port: nat-stun-port (3478)
Simple Traversal of UDP Through NAT
Message Type: Binding Request (0x0001)
Message Length: 0x0000
Message Transaction ID: 899D7C6EB4CCC54E8C8379A3D060FAEE

Interaction with other protocols

VoIP / SIP Example 12/13

Client side management:



Intel(R) 82562GT 10/100 Network Connection (Microsoft's Packet Scheduler) : Capturing - Wireshark

File Edit View Go Capture Analyze Statistics Help

Filter: `stun or sip or rtp` Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
4	5.616908	192.168.2.20	75.101.138.128	STUN	Message: Binding Request
6	5.811678	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
7	5.815224	75.101.138.128	192.168.2.20	STUN	Message: Binding Response
8	5.956350	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messenger.it:5061, with session description
10	6.459017	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messenger.it:5061, with session description
11	6.584094	212.97.59.76	192.168.2.20	SIP	Status: 407 Proxy Authentication Required
12	6.584605	192.168.2.20	212.97.59.76	SIP	Request: ACK sip:5000001@sip.messenger.it:5061
13	6.684921	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messenger.it:5061, with session description
14	6.823325	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
15	6.833426	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description
17	6.866903	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0xCECC355F, Seq=2250, Time=3058400, Mark
18	6.875931	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=16372, Time=480
19	6.876094	192.168.2.20	212.97.59.91	RTP	PT=ITU-T G.729, SSRC=0xCECC355F, Seq=2251, Time=3058560
20	6.885772	212.97.59.91	192.168.2.20	RTP	PT=ITU-T G.729, SSRC=0x31AC624B, Seq=16373, Time=640

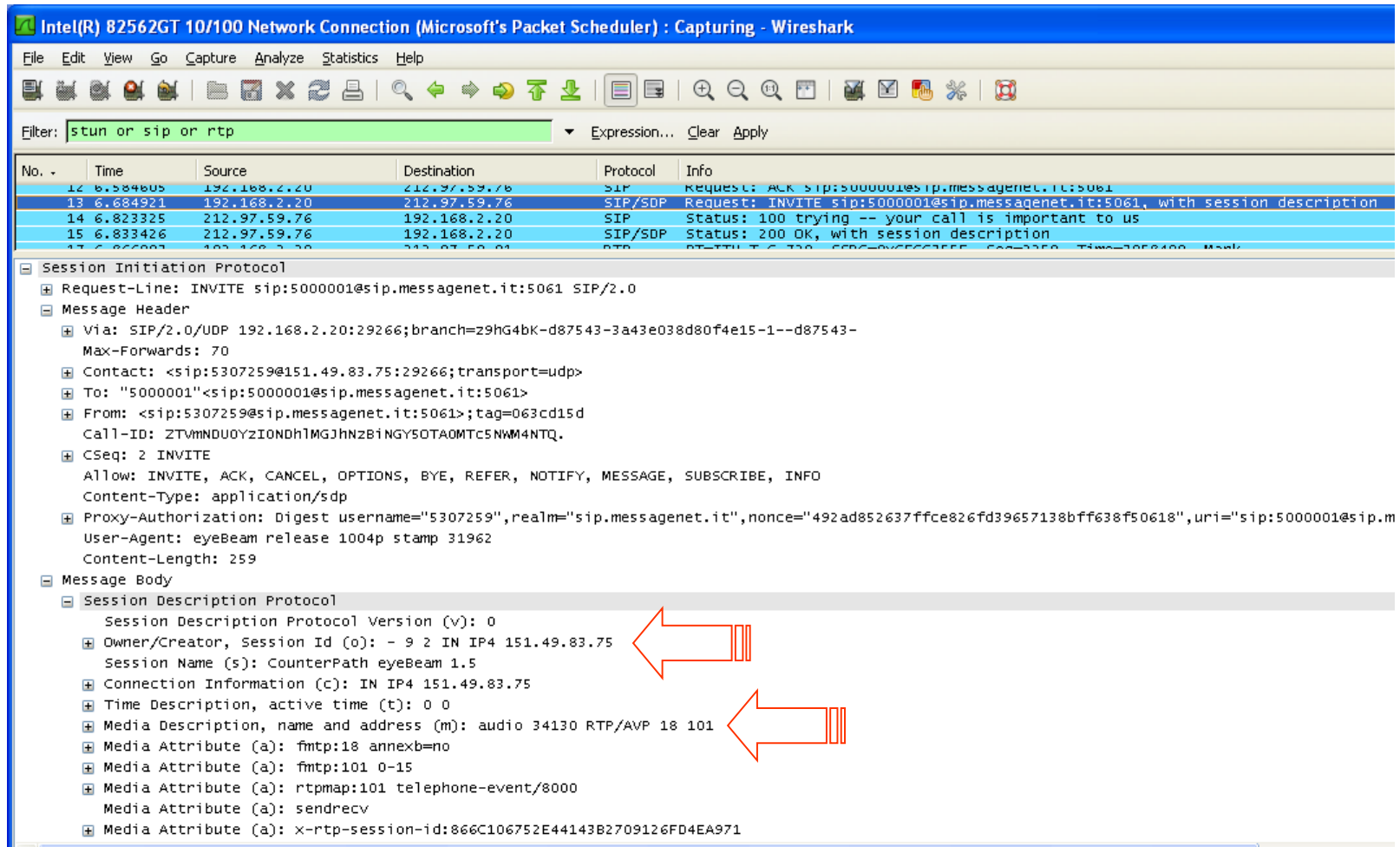
Frame 6 (130 bytes on wire, 130 bytes captured)

- Ethernet II, Src: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6), Dst: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec)
- Internet Protocol, Src: 75.101.138.128 (75.101.138.128), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: nat-stun-port (3478), Dst Port: 34130 (34130)
- Simple Traversal of UDP Through NAT
 - Message Type: Binding Response (0x0101)
 - Message Length: 0x0044
 - Message Transaction ID: 899D7C6EB4CC54E8C8379A3D060FAEE
 - Attributes
 - Attribute: MAPPED-ADDRESS
 - Attribute Type: MAPPED-ADDRESS (0x0001)
 - Attribute Length: 8
 - Protocol Family: IPv4 (0x0001)
 - Port: 34130
 - IP: 151.49.83.75 (151.49.83.75)
 - Attribute: SOURCE-ADDRESS
 - Attribute: CHANGED-ADDRESS
 - Attribute: XOR_MAPPED_ADDRESS
 - Attribute: SERVER

Interaction with other protocols

VoIP / SIP Example 13/13

Client side management:



The image shows a Wireshark capture of network traffic on an Intel(R) 82562GT 10/100 Network Connection. The filter is set to 'stun or sip or rtp'. The capture shows several packets related to SIP and SDP. The packet list pane shows the following packets:

No. -	Time	Source	Destination	Protocol	Info
12	6.584605	192.168.2.20	212.97.59.76	SIP	Request: ACK sip:5000001@sip.messengeret.it:5061
13	6.684921	192.168.2.20	212.97.59.76	SIP/SDP	Request: INVITE sip:5000001@sip.messengeret.it:5061, with session description
14	6.823325	212.97.59.76	192.168.2.20	SIP	Status: 100 trying -- your call is important to us
15	6.833426	212.97.59.76	192.168.2.20	SIP/SDP	Status: 200 OK, with session description

The packet details pane shows the following information for the selected packet (No. 13):

- Request-Line: INVITE sip:5000001@sip.messengeret.it:5061 SIP/2.0
- Message Header
 - Via: SIP/2.0/UDP 192.168.2.20:29266;branch=z9hG4bK-d87543-3a43e038d80f4e15-1--d87543-Max-Forwards: 70
 - Contact: <sip:5307259@151.49.83.75:29266;transport=udp>
 - To: "5000001"<sip:5000001@sip.messengeret.it:5061>
 - From: <sip:5307259@sip.messengeret.it:5061>;tag=063cd15d
 - Call-ID: ZTvmNDU0Y2I0NDhlMGJhbnZBINGY5OTA0MTC5NWM4NTQ.
 - CSeq: 2 INVITE
 - Allow: INVITE, ACK, CANCEL, OPTIONS, BYE, REFER, NOTIFY, MESSAGE, SUBSCRIBE, INFO
 - Content-Type: application/sdp
 - Proxy-Authorization: Digest username="5307259",realm="sip.messengeret.it",nonce="492ad852637ffce826fd39657138bff638f50618",uri="sip:5000001@sip.messengeret.it"
 - User-Agent: eyeBeam release 1004p stamp 31962
 - Content-Length: 259
- Message Body
 - Session Description Protocol
 - Session Description Protocol Version (v): 0
 - Owner/Creator, Session Id (o): - 9 2 IN IP4 151.49.83.75
 - Session Name (s): CounterPath eyeBeam 1.5
 - Connection Information (c): IN IP4 151.49.83.75
 - Time Description, active time (t): 0 0
 - Media Description, name and address (m): audio 34130 RTP/AVP 18 101
 - Media Attribute (a): fmtp:18 annexb=no
 - Media Attribute (a): fmtp:101 0-15
 - Media Attribute (a): rtpmap:101 telephone-event/8000
 - Media Attribute (a): sendrecv
 - Media Attribute (a): x-rtp-session-id:866C106752E44143B2709126FD4EA971

Two red arrows point to the Session ID and Media Description fields in the details pane.

Interaction with other protocols:

binding refresh techniques

- Some server side applications requires client to be reachable for a certain time interval (even hours) over the <IP address, port> pair that it communicated
- What happens with NAT?
 - NAT binding typically has limited duration
 - In the absence of traffic, binding is removed as timer expires and <internal_address, internal_port> no longer reachable by sending packets to <external_address, external_port>
- approach: send keepalive packets to maintain binding
 - Packets can be sent by server or client
 - Typical interval between consecutive keepalive packets: 30secs.

Interaction with other protocols:

Binding refresh techniques: Example 1/2

Wireshark interface showing a capture filter for `udp.port==5061`. The packet list shows several SIP and UDP packets. The packet details for the selected packet (No. 2042) are shown below:

No.	Time	Source	Destination	Protocol	Info
2042	8.374188	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
2043	8.463687	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
2044	8.466743	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
2047	8.567442	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
2052	8.657615	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
2053	8.659848	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)
6125	32.821300	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150
12429	62.952034	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150
25261	123.867398	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150

Packet details for Frame 2042 (614 bytes on wire, 614 bytes captured):

- Ethernet II, Src: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec), Dst: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6)
- Internet Protocol, Src: 192.168.2.20 (192.168.2.20), Dst: 212.97.59.76 (212.97.59.76)
- User Datagram Protocol, Src Port: 62150 (62150), Dst Port: sip-tls (5061)
- Session Initiation Protocol

Interaction with other protocols:

Binding refresh techniques: Example 2/2

The image shows a Wireshark network traffic capture window. The title bar reads "(Untitled) - Wireshark". The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, and Help. The toolbar contains various icons for file operations, navigation, and analysis. The filter bar shows "Filter: udp.port==5061". The packet list pane displays a table of captured packets:

No. -	Time	Source	Destination	Protocol	Info
2042	8.374188	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
2043	8.463687	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
2044	8.466743	212.97.59.76	192.168.2.20	SIP	Status: 401 Unauthorized (0 bindings)
2047	8.567442	192.168.2.20	212.97.59.76	SIP	Request: REGISTER sip:sip.messagenet.it:5061
2052	8.657615	212.97.59.76	192.168.2.20	SIP	Status: 100 Trying (0 bindings)
2053	8.659848	212.97.59.76	192.168.2.20	SIP	Status: 200 OK (1 bindings)
6125	32.821300	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150
12429	62.952034	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150
25261	123.867398	212.97.59.76	192.168.2.20	UDP	Source port: sip-tls Destination port: 62150

The packet details pane for packet 6125 is expanded, showing the following layers:

- Frame 6125 (60 bytes on wire, 60 bytes captured)
- Ethernet II, Src: USRoboti_21:e5:c6 (00:14:c1:21:e5:c6), Dst: HewlettP_81:6f:ec (00:1a:4b:81:6f:ec)
- Internet Protocol, Src: 212.97.59.76 (212.97.59.76), Dst: 192.168.2.20 (192.168.2.20)
- User Datagram Protocol, Src Port: sip-tls (5061), Dst Port: 62150 (62150)
- Data (4 bytes)
 - Data: 00000000

A red arrow points to the "Data" layer in the packet details pane.

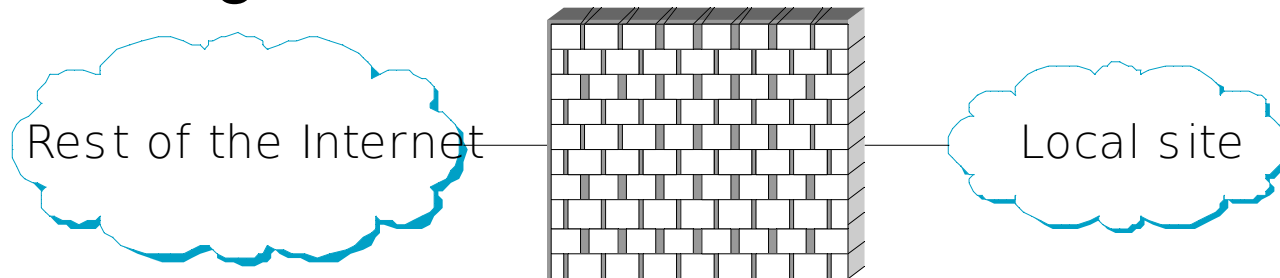
Further aspects

- Binding
 - NAT binding behaviour
 - Port binding behaviour
 - Binding timer refresh
 - Filtering
- Interaction with other protocols
 - STUN
- Basic reference:
 - http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html

Firewalls

Firewall

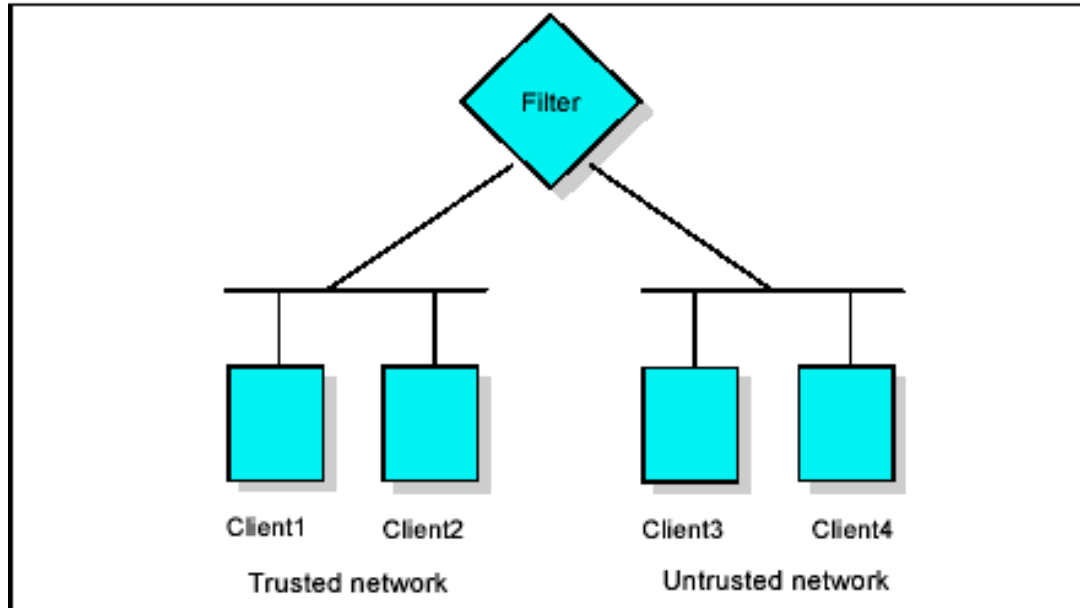
- Router with augmented functionalities, at the border between a private network and the Internet
- Forwards/filters traversing packets
- Allows implementation of centralized security policies
- Often integrated within NAT



Firewalls - components

- Packet-filtering router
- Application level gateway (proxy)
- Circuit level gateway
 - Often, but not necessarily, all present at the same time

Packet filtering router



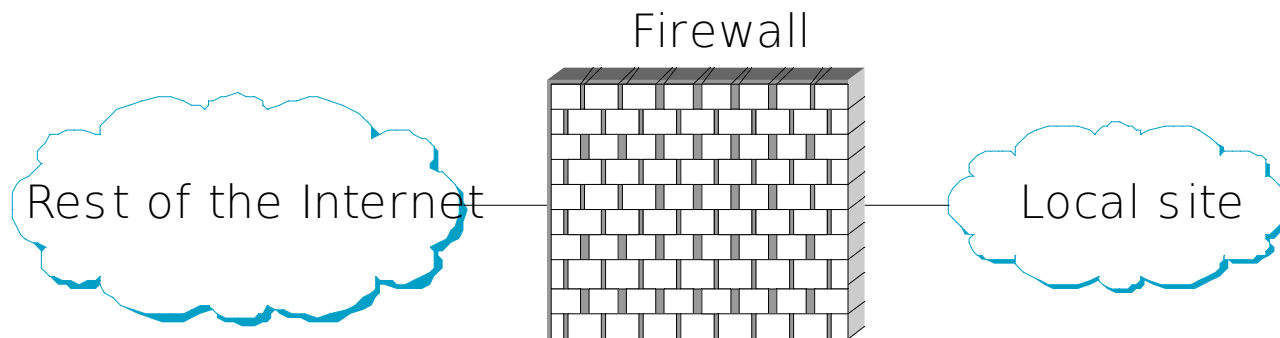
Picture from IBM© Redbook on TCP/IP

- Filtering based on IP datagrams' headers
 - IP source/destination
 - UDP/TCP Source/destination port
 - ICMP message type
 - Payload (UDP, TCP, IP tunnel...)

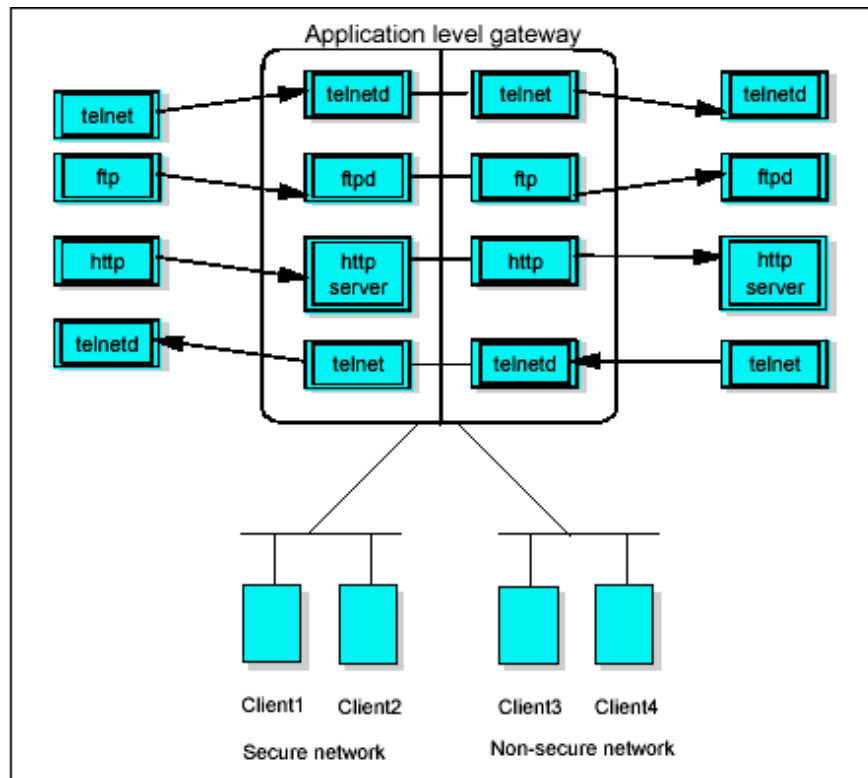
Example:

IP addresses and port numbers

- Filter-Based Solution
 - Firewall has filtering table
 - Typically, every entry is a 4-tuple (SourceIP, SourcePort, DestIP, DestPort)
 - (192.12.13/24, *, 128.7.6.5, 80)
 - (*, *, 128.7.6.5, 80)
 - default: forward or drop



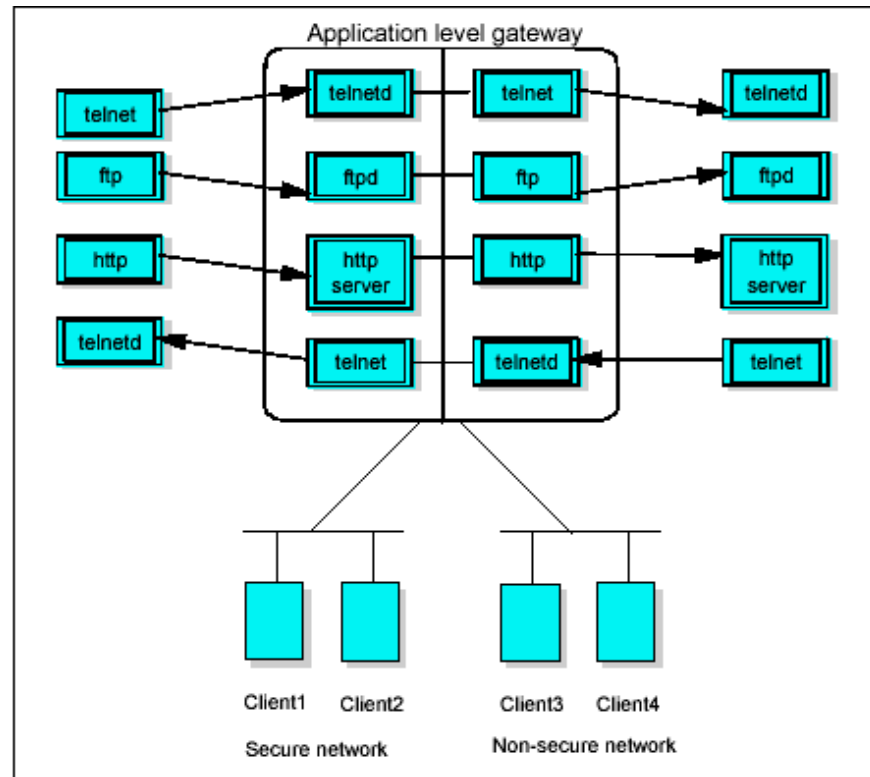
Application Gateway



Picture from IBM© Redbook on TCP/IP

- One proxy for every application
- Proxy behaves as a server to *local* client and as client towards remote server

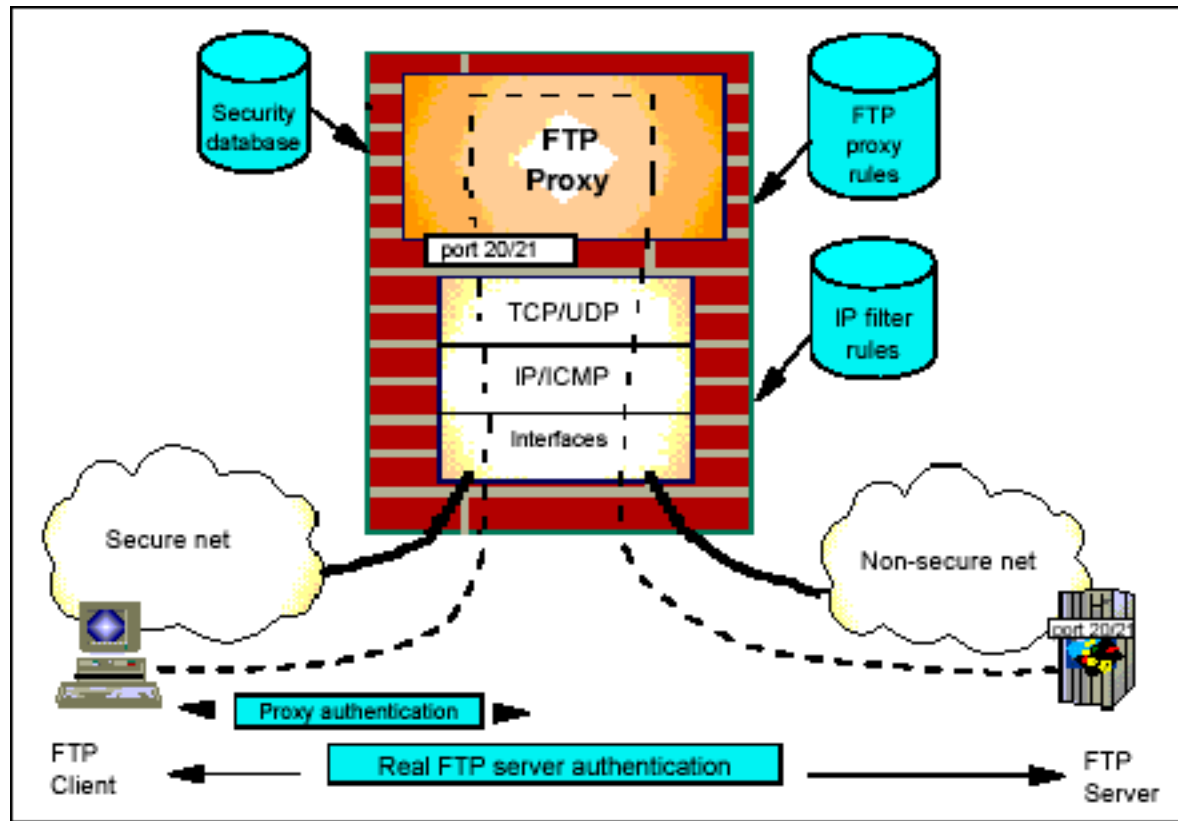
Application Gateway/cont.



Picture from IBM© Redbook on TCP/IP

- Possible to disable IP traffic between internal network and the Internet but ...
- ... only applications implemented at proxy can interact with external nodes

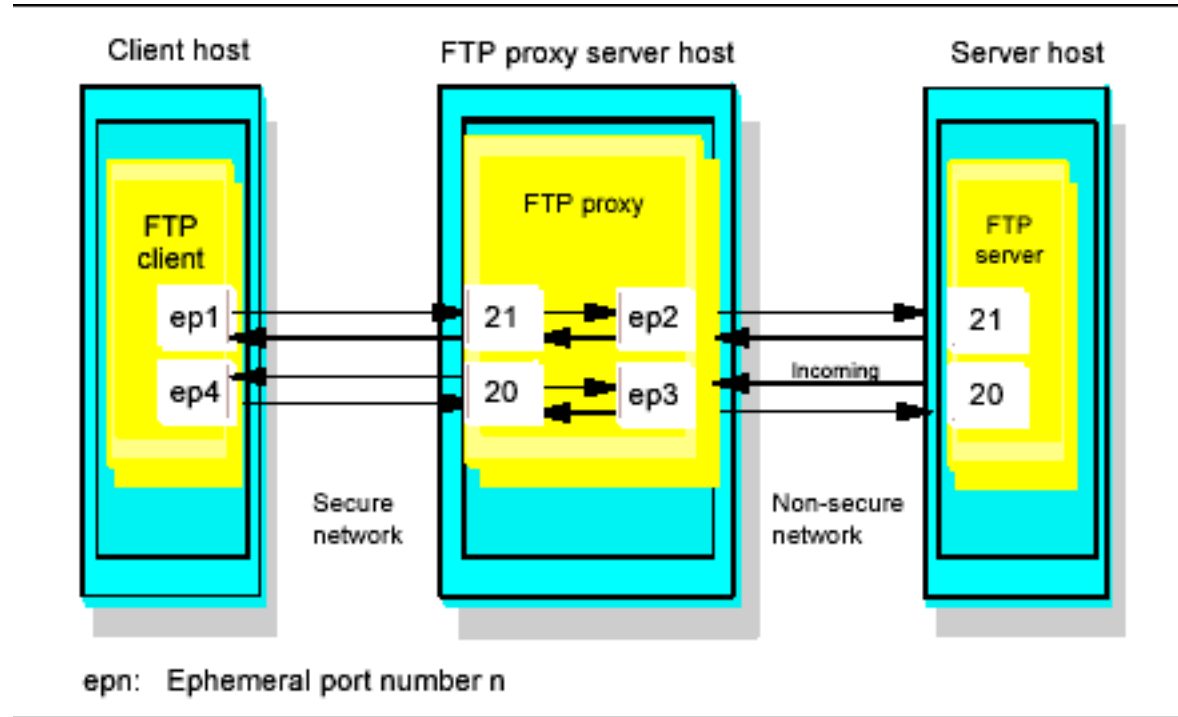
Example - FTP



Picture from IBM© Redbook on TCP/IP

- Necessary to modify FTP Sw
 - Necessary to implement proxy traversal

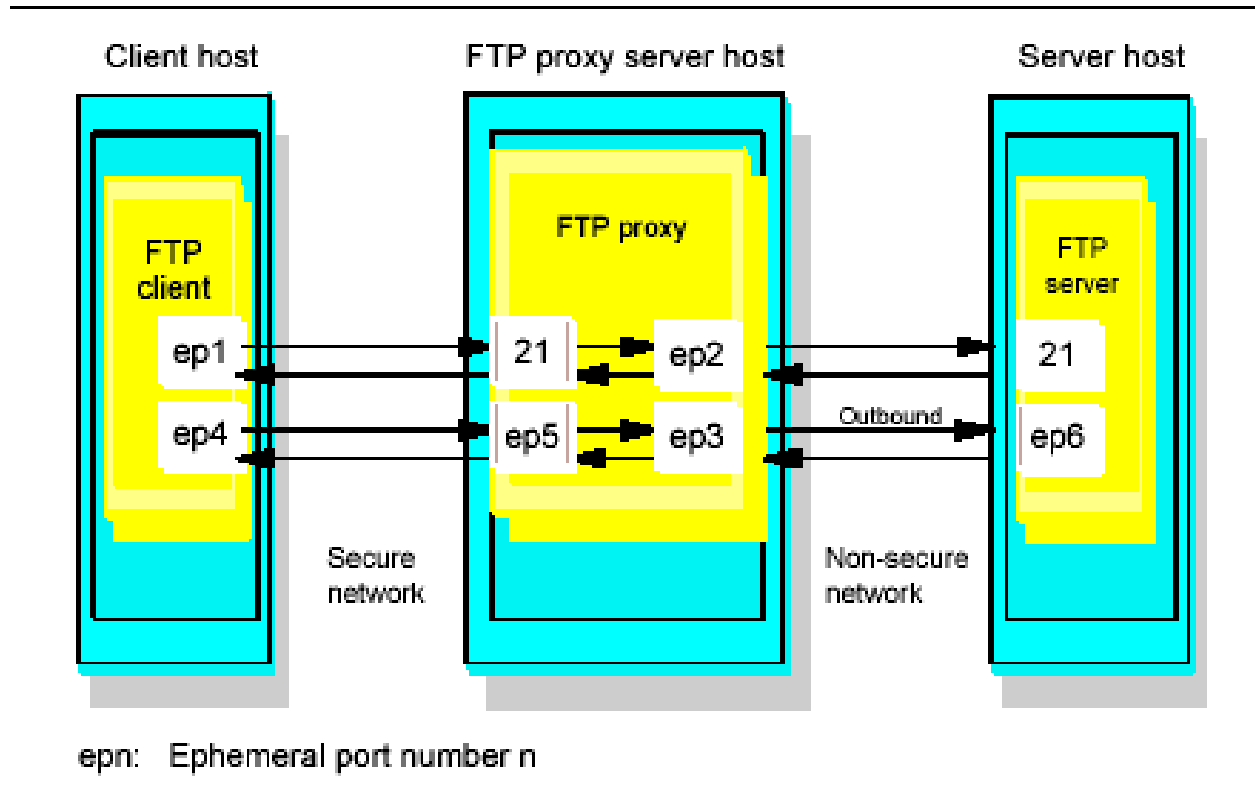
FTP/cont.



Picture from IBM© Redbook on TCP/IP

- Active mode operation
 - Possible attacks - which ones?

A possible fix...



Picture from IBM© Redbook on TCP/IP

- Passive mode or Firewall-friendly FTP

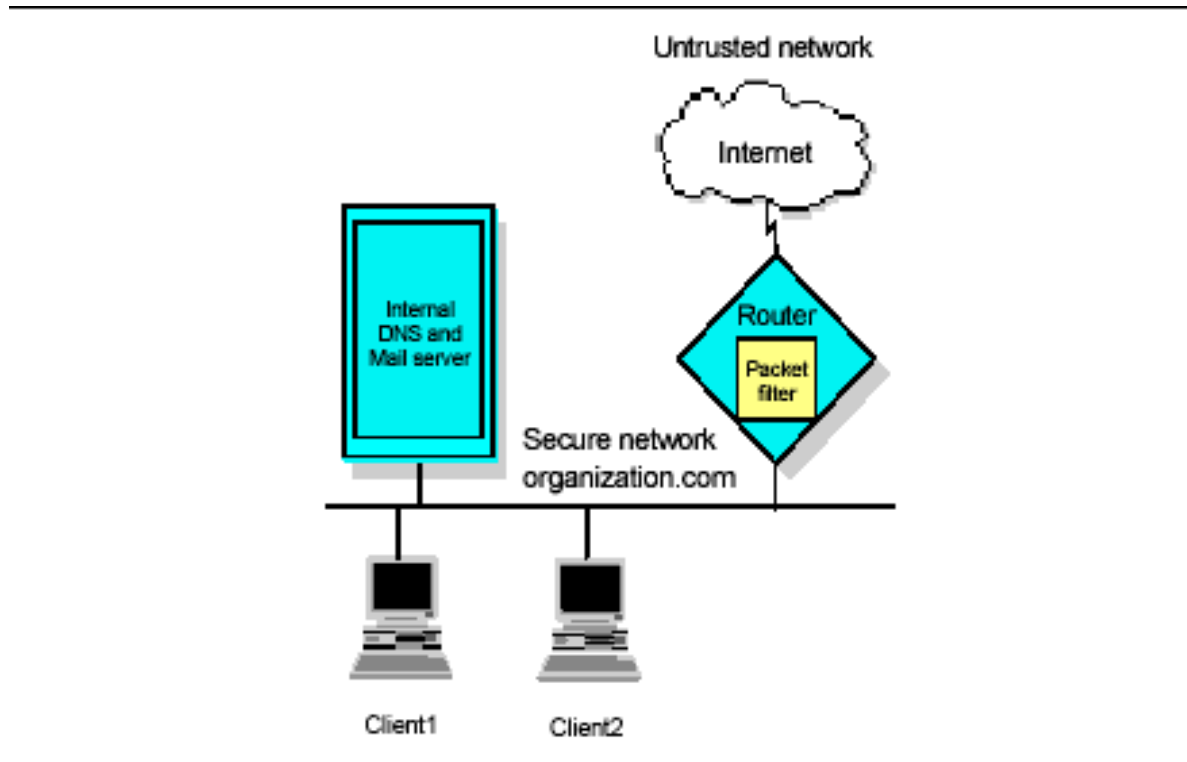
Circuit Level Gateway

- In practice: a transport-layer proxy (UDP or TCP)
- Allows all applications that satisfy desired requirements at the transport layer
 - Example: allow all and only TCP traffic; for UDP allow only DNS traffic
 - All TCP-based applications work without changes, as well as DNS
- No packet filtering

Firewalls

Firewall architectures

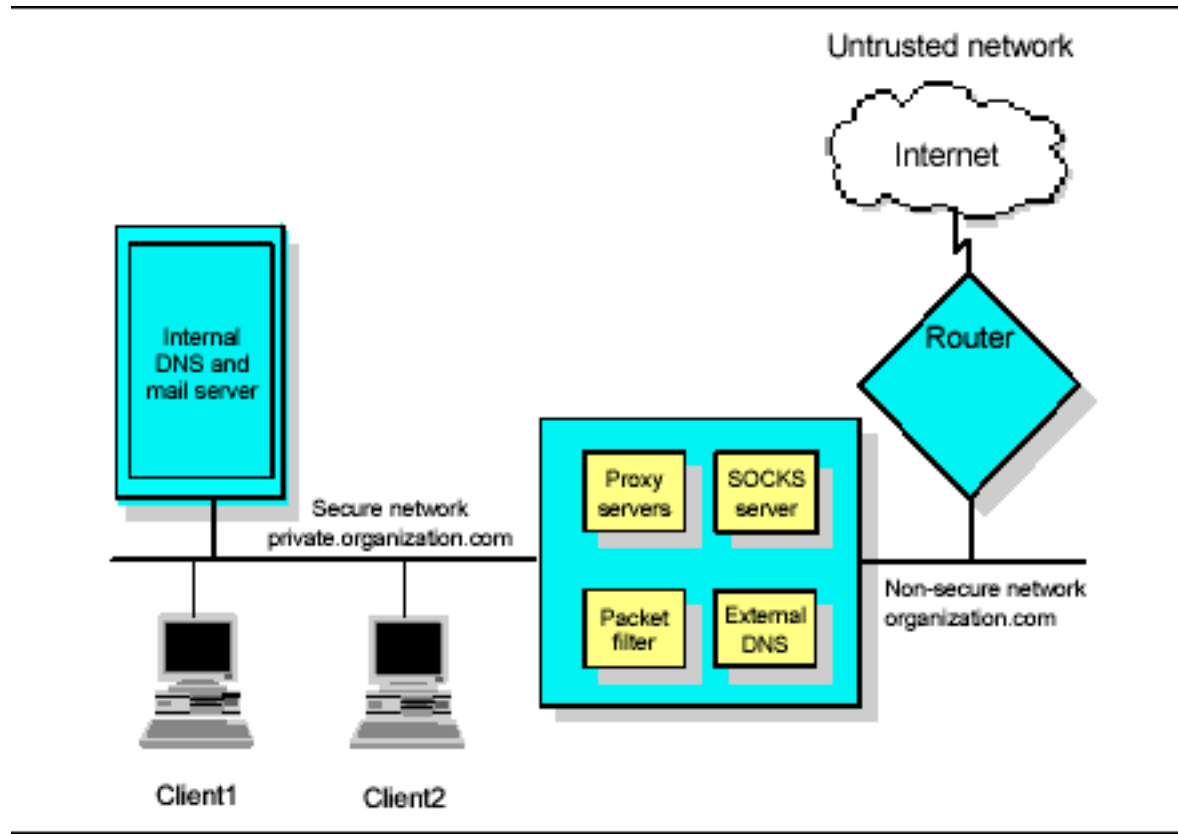
Packet filtering



Picture from IBM© Redbook on TCP/IP

- Cheap
 - Simply a router performing packet filtering
- **Rules: usually, what is not explicitly allowed is forbidden**

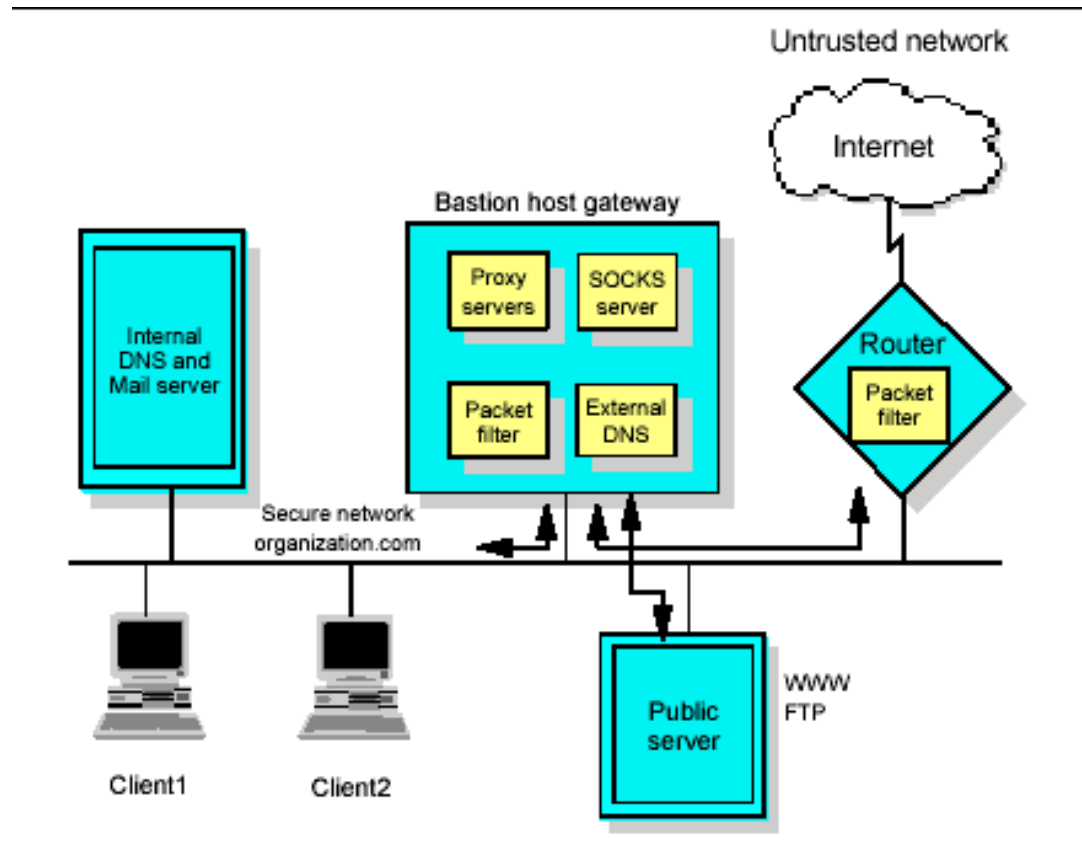
Dual-homed



Picture from IBM© Redbook on TCP/IP

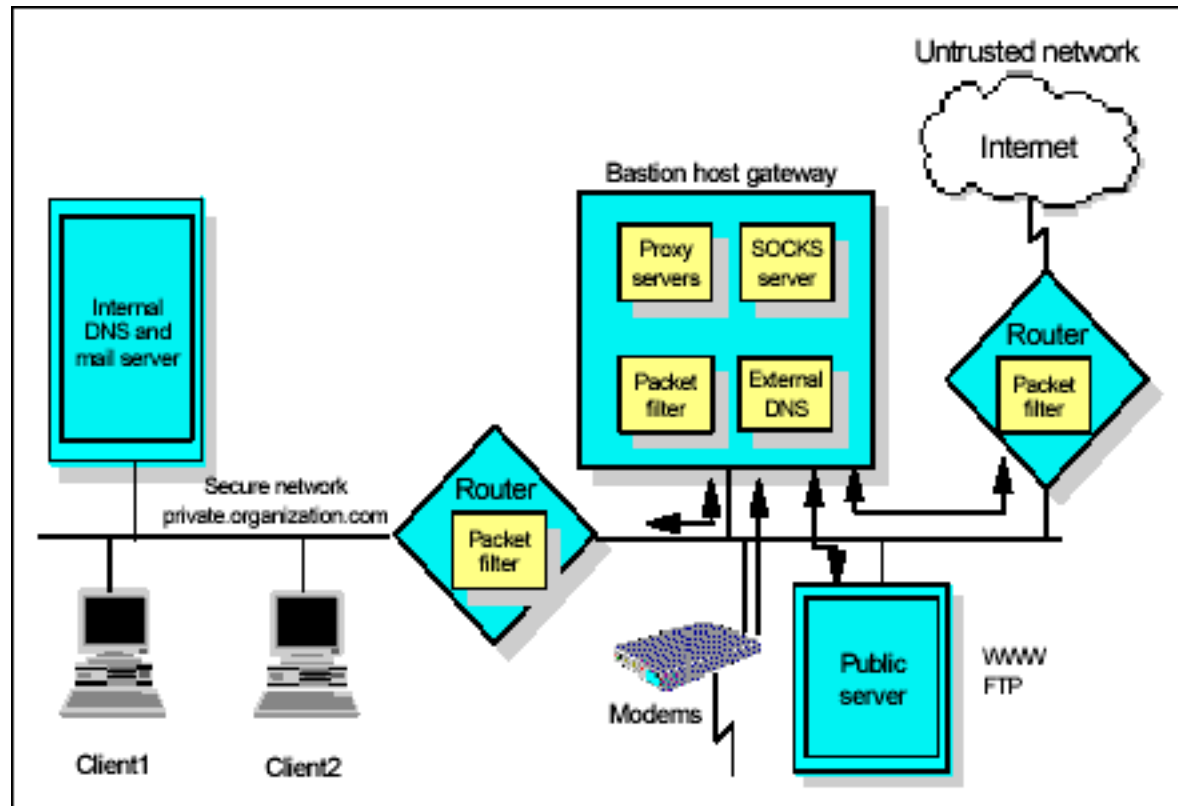
- At least 2 network interfaces (and 2 IP addresses)
- **Everything which is not allowed is forbidden**
- No direct IP traffic between local network and the Internet possible
- E.g.: proxy firewall

Screened host



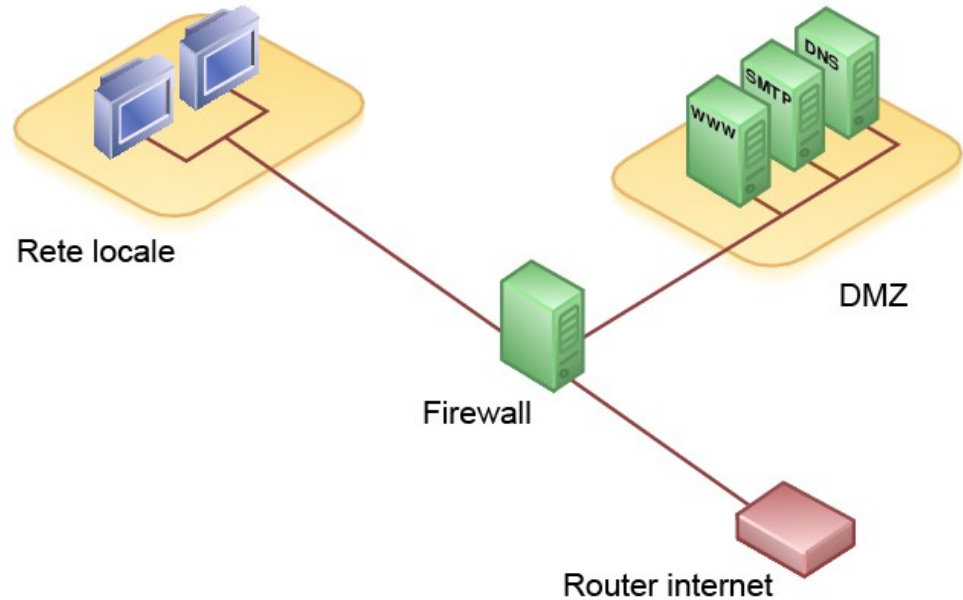
- Application GW (bastion) and packet filter
- Server on same subnet as bastion GW
- Access to server from outside *only* through bastion GW
 - How do you implement this policy?
 - How do you screen server from internal traffic?

DeMilitarized Zone



- Both internal and external traffic through Bastion GW
- Maximum level of security

DeMilitarized Zone



- DMZ can be created by defining distinct policies on one or more firewalls

References

- Tunneling
 - GRE RFCs 2784 and 2890: <http://tools.ietf.org/html/rfc2784> (rfc2890)
 - Overview of Virtual Private Networks: <http://www.vpnc.org/vpn-technologies.html>
- NAT
 - http://www.tcpipguide.com/free/t_IPNetworkAddressTranslationNATProtocol.htm
 - NAT and STUN
 - http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_7-3/anatomy.html
- Firewalls: IBM's TCP/IP Tutorial and Technical Overview. Available on-line
 - <http://www.redbooks.ibm.com/redbooks/pdfs/gg243376.pdf>