

# Internet infrastructure

Prof. dr. ir. André Mariën

# Topics:

- Protocols on top of IP:
  - ICMP: Internet Control Message Protocol
  - UDP: User Datagram Protocol
  - TCP: Transmission Control Protocol
- IP – Ethernet link
  - ARP: Address Resolution Protocol
- Network parameter configuration
  - DHCP: Dynamic Host Configuration Protocol
- Network address translation
  - NAT

# Internet Control Message Protocol: ICMP

- Protocol for managing and troubleshooting IP networks
  - Simple communication packets, self sufficient
  - Signaling problems
  - Aiding configuration
  - troubleshooting
- Exchanged between IP nodes
  - Intended for the nodes itself, not users nor services
- Packet essentials:
  - ICMP type and ICMP code
- ICMP types:
  - destination unreachable
  - time exceeded
  - echo request and reply (ping, traceroute)
  - router advertisement and solicitation
  - ...

# TCP & UDP addressing

- TCP and UDP add an address space: the Component address space
  - Port number = service identification
  - On top of IP: machine address space
    - IP number = network + node identification
- Connection: 6-tuple
  - Remember: node address is a tuple:
    - (LAN, component)
  - ( (LAN1,component1,service1), (LAN2,component2,service2))

# User Datagram Protocol: UDP

- Adds little to basic IP functionality
  - Send a packet from A to B
  - No retrial, no ...: any added value must be implemented on top of it
    - Best effort: packets may be dropped
    - No connection set-up
  - Core addition: intended for services on an IP node
    - Adds another level of addresses: ports
    - A service on a node uses an IP address and a port
- Packet essentials:
  - IP packet, data type UDP
  - IP data: (UDP source port, UDP destination port, data)
- Advantages
  - simplicity:
  - Speed: just send it (no handshakes, round trip delays, ...)
- Easy to fake addresses, insert packets, ...

# Transmission Control Protocol: TCP

- TCP
  - introduces “connection”
  - More than packets: protocols to set up, maintain, end connection
  - Protocol to increase reliability
  - Some security as side effect
    - a simple form of connection authentication
    - Requires correct numbers, one series per side
- TCP packet essentials:
  - (source port, destination port, sequence number, acknowledge number, data)
  - Other elements in the header to support features:
    - flags, checksums, lengths

# TCP flags and handshake

- TCP provides a “connection-oriented” service
  - No real connection
    - the connection exists only for sender and receiver: they can recognize packets belonging to the “connection”
    - Intermediate components (mostly) do not care if a packet is TCP/UDP/...: they work on individual packets
  - Connection set-up: three way handshake
    - Party 1: Start connection with “SYN” flag on packet
    - Party 2: Respond with “ACK” (on party 1 SYN) and (own) “SYN”
    - Party1: Respond with “ACK”

# Other flags

- FIN flag: want to end connection
- RST flag: abort connection
- Note: not all flag combinations make sense
- Christmas tree attack (all lights on): send packet with all flag on; crashed system's IP stack

# Simple flag-based filtering

- IP-port restrictions:
  - To block connections, simply look for “SYN” packets
  - Allow only “SYN” packets for authorized (IP,port) combinations
  - Result: no connection can be created for non-authorized (IP,port) combinations
  - Stateless “firewall”
- Killing connections
  - If a packet matches an attack signature, insert a “RST” packet in the connection, both ways
    - Replacing the suspect packet
    - Sequence numbers can be taken from the suspect packet

# Denial of Service

- Denial of service = DoS
- What: make a service unavailable
- Technical objective:
  - Direct: avoid that the service can be used
    - Web server: “block” access to its content
    - ISP network: block use of the network
  - Indirect: take out legitimate service to put stub in place
    - Stub is attacking users of the real service: false information
    - Avoid that the real service detects the stub/alternate systems

# Attacker objective

- Most important: financial gain
  - Extortion: pay or lose service
  - Undermine faith in competition
- Not to be ignored: political/activist
  - Wikileaks DoS of adversaries of wikileaks
  - Avoid coordination between opponents
    - Google China
    - Egypt twitter & facebook
    - DNS blocking of sites on simple demand
    - Internet kill switch

# DoS principle

- Key idea 1: find a lever
  - Attack resources are lower than defender resources
    - Start connecties but never complete (network level)
    - Requests draining back-end resources
      - Search all documents with the letter “e”
    - Complex session set-up server side before authentication
- Key idea 2: power of the masses
  - Distributed DoS
  - Multiple low-power systems coordinating

# SYN flooding – timeouts

- As soon as a “SYN” packet is received, some data is created to manage the session
  - This means resource allocation
  - Resources are fixed
  - Only a fixed amount of connections can be supported
- Sending “SYN” packets from spoofed (fake) IP-port combinations accumulates resources on the target
- Result: resource exhaustion, no legitimate user can create a connection anymore
- Timeouts can help to clean up, destination unreachable (on attempts to send the SYN-ACK) can help
- Throttling can help: number of SYN from specific (range of) IPs, number of SYNs per second etc.

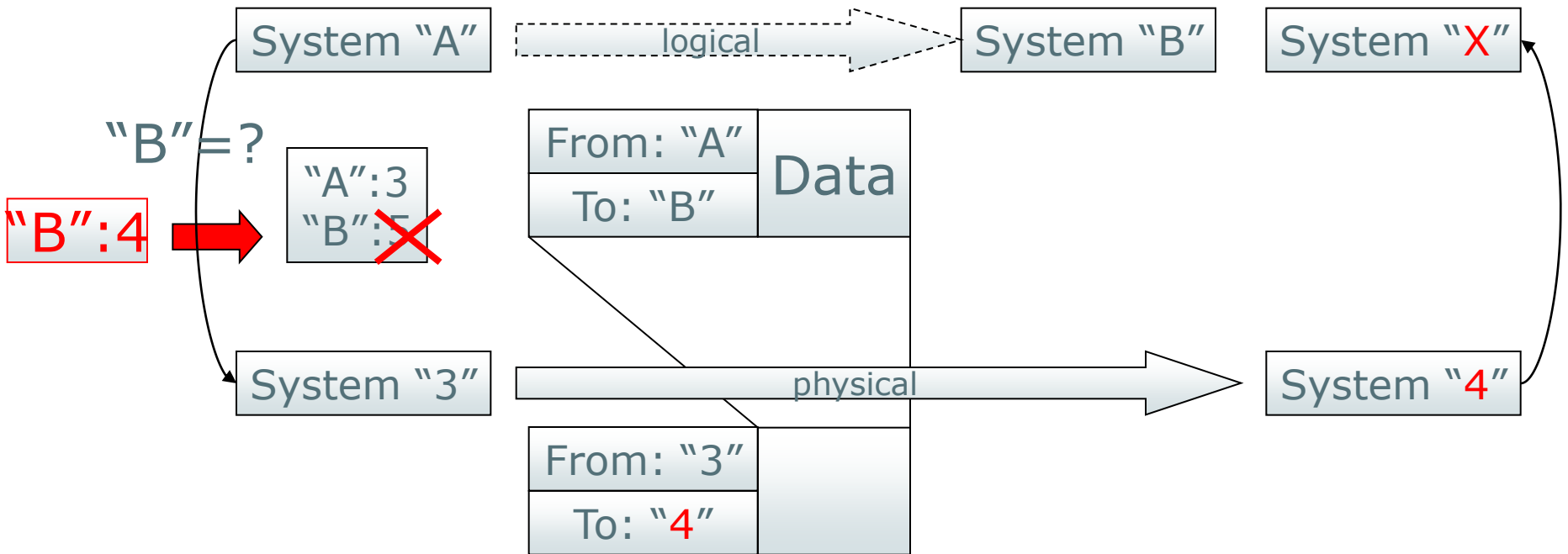
# Man-in-the-Middle

- In security discussions, a lot of times the problem of the “Man in the Middle” is described
  - A notorious class of attacks
- Variations:
  - All messages
    - from A to B
    - Both ways
  - Can be
    - Read (passive MitM)
    - Created, Deleted, Modified (active MitM)
  - By M
- Active MitM is a serious problem to make safe protocols
- Attacker either
  - Is a natural MitM (traffic flows by M by design)
  - Abuses protocols to become a MitM

# Man-in-the-Middle

- There are two more likely places where a MitM position is relatively easy
  - Near the sender
  - Near the receiver
- If you are in the same broadcast network (wired or wireless), clearly you have an easy job of passive MitM
- If you sit on the route of the packets, you have a good chance of being able to delete and insert packages
  - But how do you create such a situation?

# Man-in-the-Middle



**Result: Man-in-the-Middle. Though adversary!**

No attack on the logical protocol, but on the naming translation service  
Examples: DNS spoofing, ARP spoofing, BGP poisoning

# Man-in-the-Middle

- Most protocols are not good at protecting from MitM
  - The security measures are open and clear
    - MAC addresses, IP addresses: identification only
  - The layer-to-layer translators are vulnerable as well
    - DNS replays can be spoofed, poisoning can be done
    - DHCP provides means to change the targets configuration without notification

# Passive MitM on IP, UDP, TCP

- Any component where the traffic is visible is a potential MitM
- Ethernet : all components in the segment
  - Consequence: also for all layers on top (IP, UDP, TCP)
- Switched ethernet:
  - Switch
  - Initial packets with destination unknown (learning phase)

# Active MitM

- Sending packets with fake “from” address is easy
- Killing packets: on the switch
- Assuming address of off-line system
- Inserting legitimate packet based on snooped information
  
- Actual owner may/may not be able to see spoofing attempts and may/may not have detection in place

# Network scanning

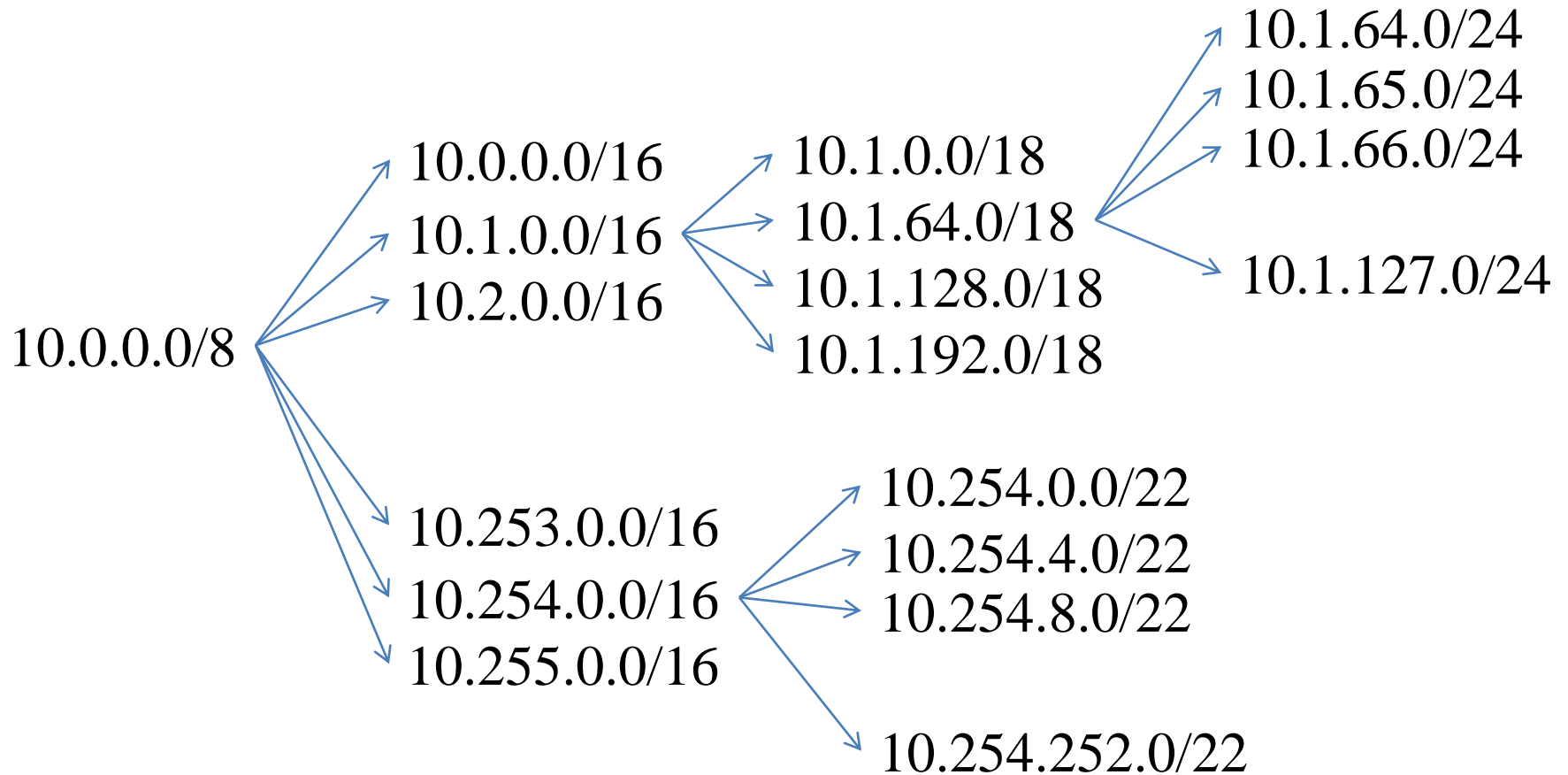
- Find out which nodes exist
  - ICMP echo request (ping utility)
  - Frequently available services
    - TCP connect to port 80,23,... (half-open: just see if there is a SYN/ACK)
    - SNMP request (see SNMP for details)
- Find out which services: attempt to connect
  - To common services (UDP/TCP)
  - To services with known issues
  - To all ports
- Level of visibility:
  - Trotteling
  - Selection
- Passive: sniff network and collect address and port information
- Note: only with proper authorization!

# Network scanning tools

- Note: many organizations do not allow even the presence of these tools on LAN connected computers, so do not even consider using them!
- Active:
  - Nessus, nmap
- Passive (mostly):
  - Wireshark, Snort, Netcat, Hping2, Tcpdump, Ettercap

# Routing

# Subnetting



# Routing

- Static routing: configuration
  - Specific routes
  - Default route
- Dynamic routing: routing protocols
  - Distance vector
  - Link state routing

# Routing styles

- Distance vector
  - Compute distance to locations based on neighbor information on distance to locations
  - Exchange distance information with your neighbors
  - Example: RIP
  - Used mainly internally in organizations
- Link state
  - Determine local view
  - Distribute
  - Compute global view
  - Examples: OSPF, BGP4
  - Used mainly as global routing algorithm

# Screening Router

- Security function
- Packet filtering
  - “from” and “to” addresses
  - “from” and “to” ports
  - protocol (UDP, TCP, ICMP)
- Stateless/stateful

# Routing MitM

- Routers are excellent places for MitM, both passive and active
- Routers close to point-of-interest are best
  - “home router” ideal for hacker
    - Hardly secured
    - No security measures like AV
    - Active MitM easy
- Rerouting invisible except eventual performance hit
- Can happen on AS routes
- Rerouting to black holes: “filtering”

# Routing-related incidents

POLITICS

## Internet Traffic from U.S. Government Websites Was Redirected Via Chinese Networks

By Joshua Rhett Miller

Published November 16, 2010 | FoxNews.com

Print Email Share Comments (693) Recommend 1K Text Size



AFP/Reuters

When 15 percent of the world's Internet traffic — including the Pentagon, Defense Secretary Robert Gates office, the Senate and several U.S. government agencies — was redirected last April onto computer routers in China, it also may have left the sites vulnerable to surveillance — or worse.

Nearly 15 percent of the world's Internet traffic — including data from the Pentagon, the office of Defense Secretary Robert Gates and other U.S. government websites — was briefly redirected through computer networks in China last April, according to a congressional commission report obtained by FoxNews.com.

It was not immediately clear whether the incident was deliberate, but the April 8 redirection could have enabled malicious activities and potentially caused an unintended "diversion of data" from many U.S. government, military and commercial websites, the U.S.-China Economic and Security



# Not the first incident

- April 1997: autonomous system 7007 (AS7007) announced routes to all of the Internet
- December 2004: thousands of networks in the US were misdirected to Turkey
- September 2005: AT&T, XO and Bell South networks were misdirected to Bolivia
- July 2007: Yahoo was unreachable for an hour due to a routing problem
- February 2008: Pakistan Telecom hijacked all traffic aimed at YouTube and took YouTube offline for two hours
- April 2010:
  - about 15% of the world's Internet *prefixes* was hijacked by a set of servers owned by China Telecom
  - Popular websites such as dell.com, cnn.com and amazon.de were “re-routed” through Chinese networks before reaching their destinations
  - for about 18 minutes

# Layer interaction: IP - ethernet

# Ethernet – IP

Ethernet:



IP:



Ethernet+IP:



# Layer interaction

- Communication: IP from – IP to
  - IP from: IP from own node (interface)
  - IP to: can be any address in the address space
- Actual transmission one layer down: ethernet
  - Need “MAC from” and “MAC to” addresses
- How to determine MACs given the IP addresses?
  - Central Database with all IP – MAC mappings?
  - Central service to request MAC given IP?
  - Hierarchical system?

# Layer interaction in simple, flat network

- Simple:
  - Single IP network
  - Single Ethernet network
  - One-to-one mapping
- Request/reply protocol
  - Any system can ask “please reply with your MAC if you have IP xxx’
  - System with IP xxx replies with the corresponding MAC
  - Distributed solution, no database
  - Answers may be cached
    - By the requestor
    - By any other system (all systems “see” the requests and replies)

# Interaction – refinements

- Protocol: based on ethernet level
  - Multicast request
- Query depends on IP network structure
  - Within local LAN, one ethernet broadcast domain
  - Destination in a LAN: broadcast request on the LAN will either
    - Yield answer: system on that LAN and running
    - No answer: system may exist but is not responsive
  - Destination not on LAN:
    - No point trying to send ethernet broadcast packet: destination is not in reach
    - Must pass through router on this LAN
    - Determine router IP, request MAC for that IP

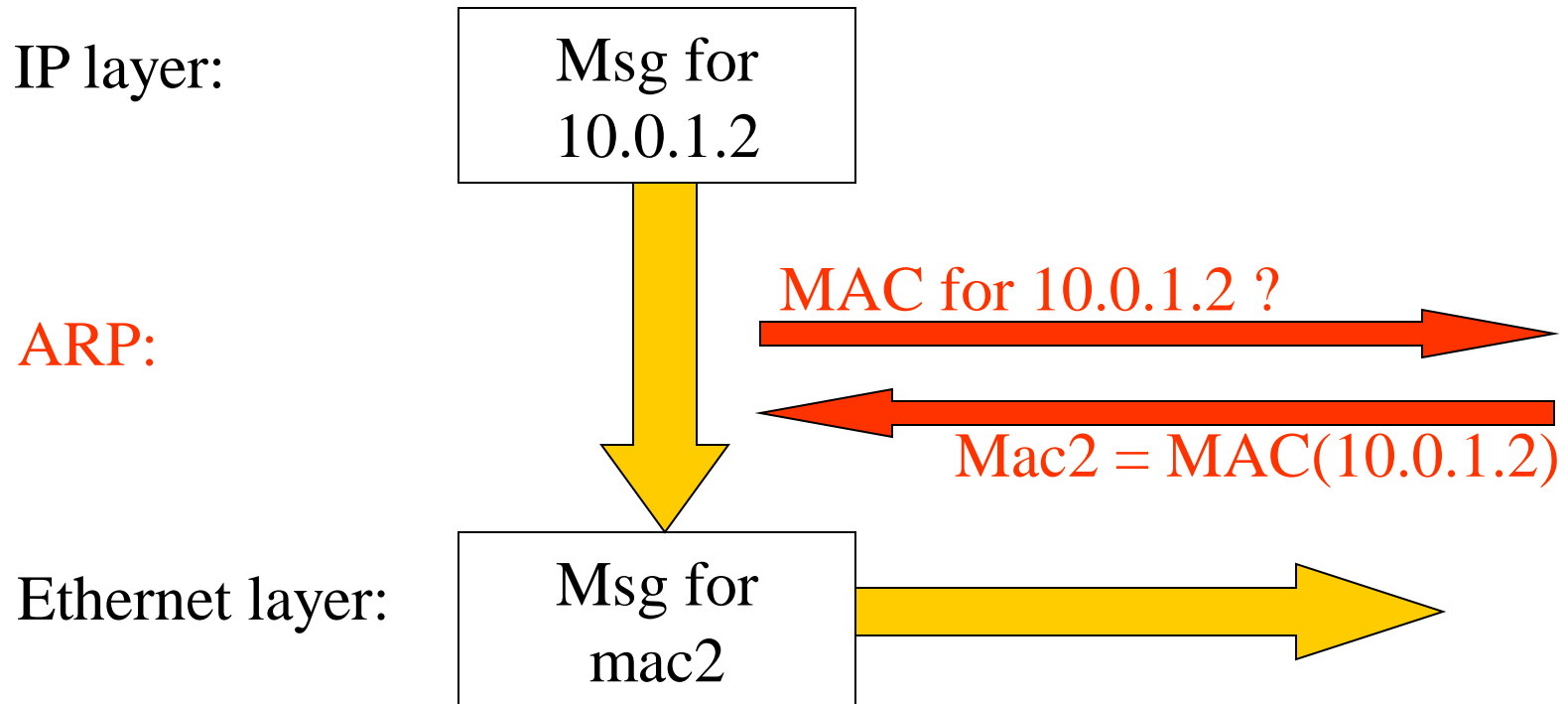
# Ethernet - IP layers interaction

- Need the association between MAC and IP
  - Query: retrieve MAC for given IP
- Address Resolution Protocol: ARP protocol
  - IP -> MAC table
  - if not in table, ask via layer 2 broadcast: which MAC for this IP?
- ARP/RARP
  - ARP: RFC 826 (STD 37)
    - Which MAC for IP address?
  - RARP: RFC 903 (STD 38)
    - Which IP for this MAC?
    - usage: configuration

# Interaction risks and abuse

- If the mapping is wrong:
  - The packets are send to MAC M instead of MAC B
  - System M can see all communication
  - System M can then forward to MAC B
    - Invisible on the IP level
  - System M can send packets to B with from 'Mac A'
    - Replies will go straight to A
    - B cannot detect half of the communication (requests from A) is intercepted
- Protocol to obtain mapping is not protected
- Caching persist the problem

# ARP



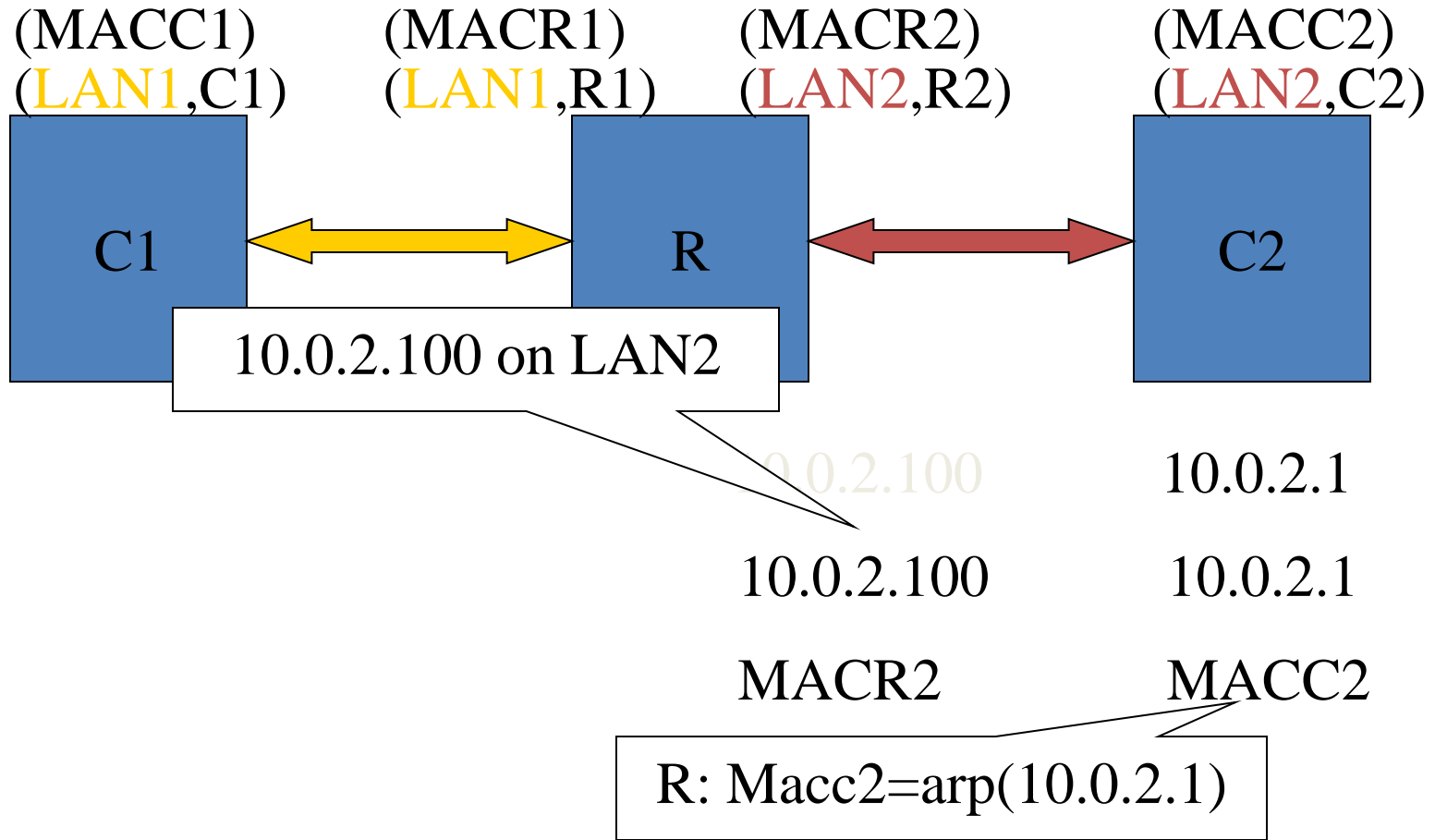
# ARP detail

- A needs to send packet to B
  - A knows IP<sub>b</sub>, needs MAC<sub>b</sub> to send
  - Broadcast: MAC<sub>a</sub>, IP<sub>a</sub> wants to know MAC<sub>b</sub> for IP<sub>b</sub>
  - All machines on the network update their caches: IP<sub>a</sub> = MAC<sub>a</sub>
  - Machine B with IP<sub>b</sub> responds to MAC<sub>b</sub> only: IP<sub>b</sub> = MAC<sub>b</sub>
- ARP cache
  - Database with associations between IP and MAC
  - Can have static entries
  - Entries expire (aging or LRU, for instance)
  - Can cope with configuration changes

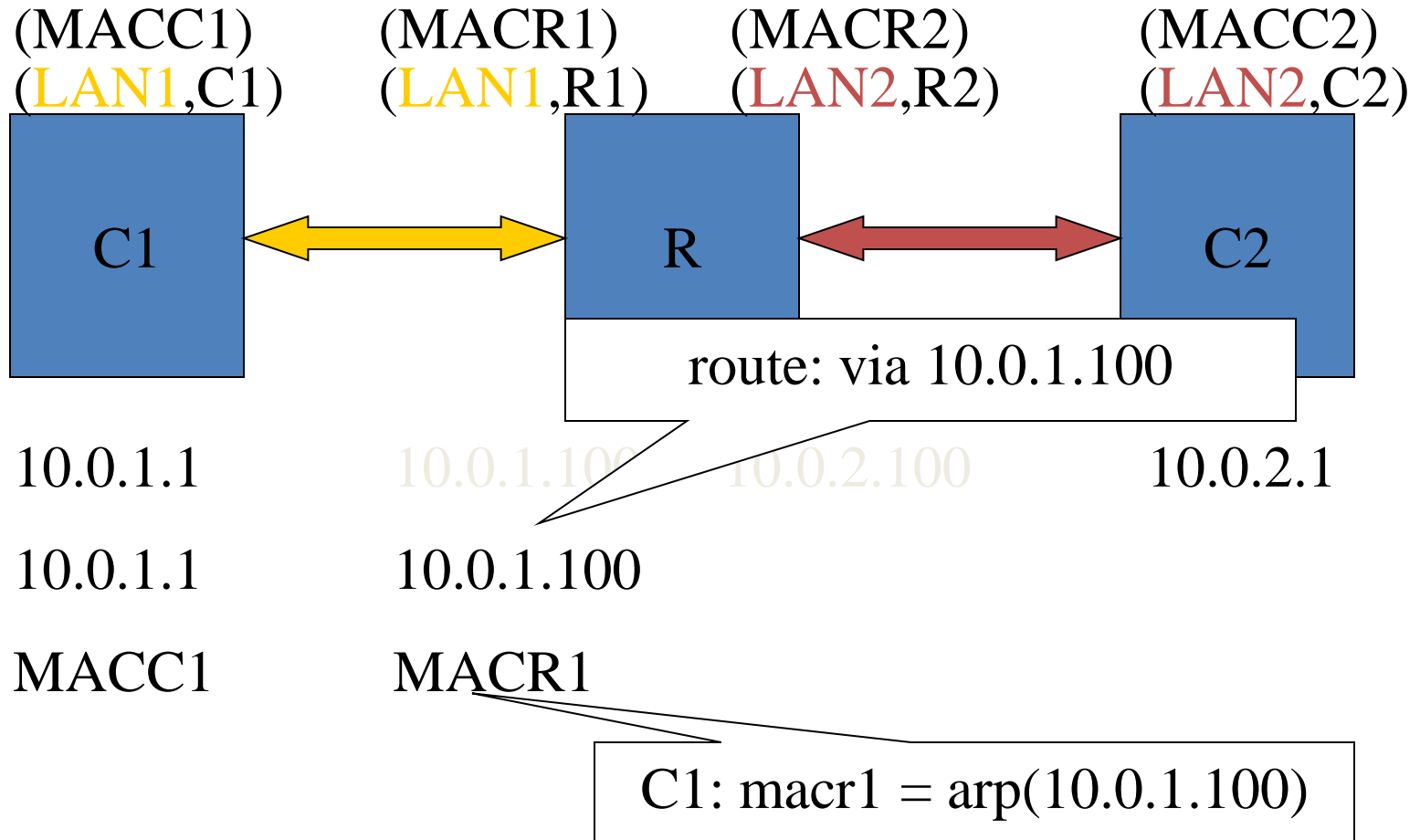
# ARP and routing

- How do ARP and routing cooperate?
  - A router handles packets not destined for its IP
  - What IP to use?
  - What ARP requests to issue?
  - Broadcast or unicast messages?
- Simple routing
  - packet with destination IPDest
  - in one of the linked subnets:
    - use ARP on that subnet to obtain MACdest
    - send to MACDest via corresponding interface
  - otherwise:
    - pick up the router IP: IPRouter
    - find MAC of router: MACRouter
    - send to MACRouter

# Host addressing principle



# Host addressing principle (cont.)



# Host addressing principle (cont.)

- component C1 wants to send a packet to component C2
- C1 sits on LAN1, C2 on LAN2
- C1 sees from the address of C2 it needs to use the router R
- C1 obtains MAC 'MACR1' for router R on LAN1 (ARP)
- C1 sends IP packet for C2 to MACR1
- router sees packet on its LAN1 interface for C2
- router sees C2 sits on LAN2
- router uses LAN2 interface to obtain MAC 'MACC2' for C2 (ARP)
- router sends IP packet for C2 to MACC2 on LAN2 interface

Network Address Translation

**NAT**

# Help! We run out of IP addresses!

- IP addresses are 32 bit
- More and more devices get IP addresses
- IP addresses were handed out in blocks (256, 4096, and even bigger chunks)
- Often: many unused, but assigned
- Uniqueness only required for some communication, inside islands: may clash with other islands

# Reduce pressure on IP addresses

- Islands could use any set of IPs
- To avoid collision risks and complexity: reserve addresses for islands (RFC 1918)
- Island-to-internet communication: provide bridging
- Explicit:
  - Staged connectivity: from A to bridge, from bridge to B
- Implicit:
  - (semi-)transparent A to bridge to B

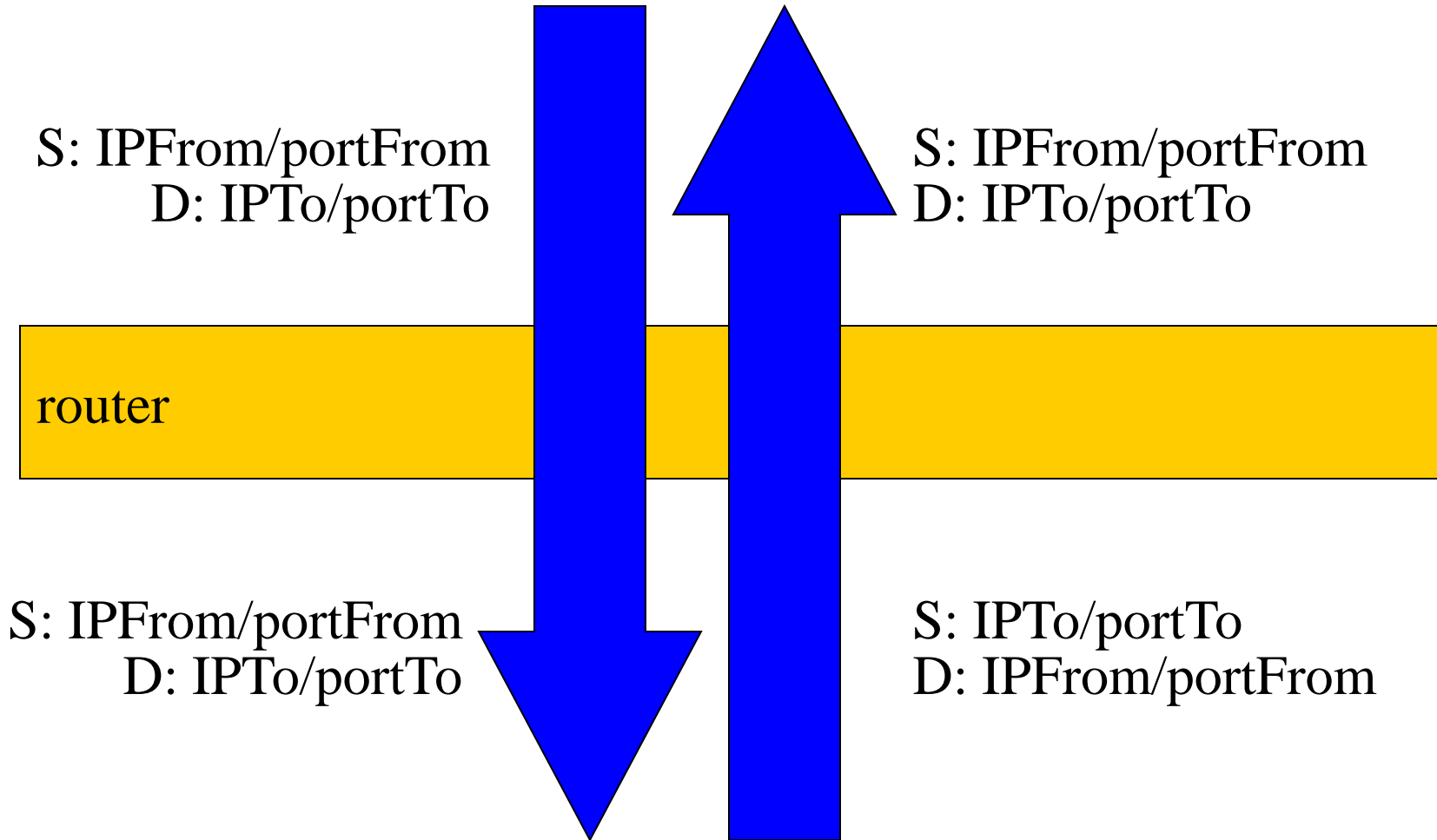
# NAT: when to use

- Security: hides internal addresses
- Cost: requires less official addresses
- Transparency for internal components
- Internal use of “special” addresses (RFC1918)
  - 10.0.0.0 - 10.255.255.255
    - Conventional use: inside big LANs
  - 172.16.0.0 - 172.31.255.255
    - Conventional use: no configuration
  - 192.168.0.0 - 192.168.255.255
    - Conventional use: DMZ

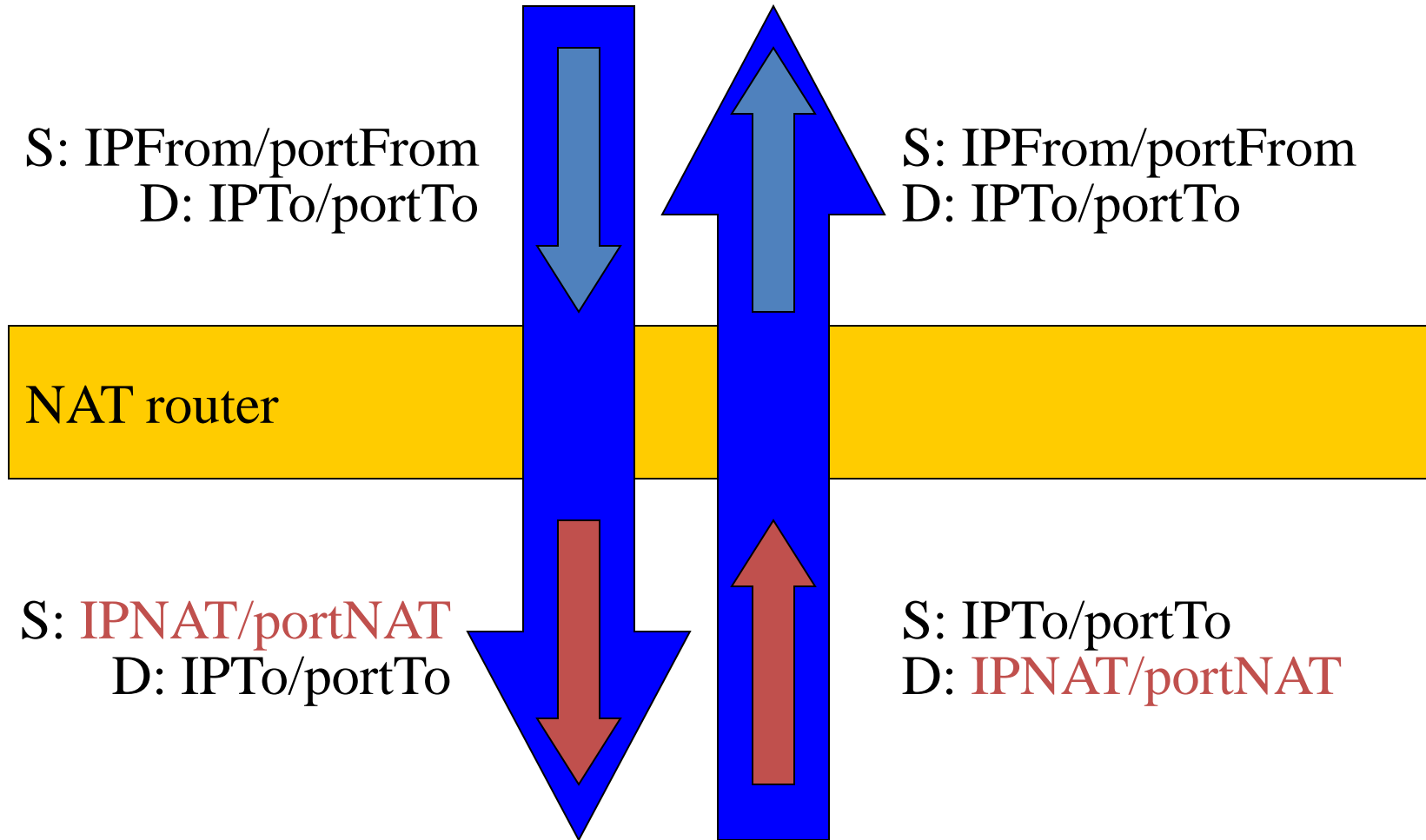
# NAT: Network Address Translation

- NAT breaks connection in two:
  - “from” - NAT component
  - NAT component - “to”
- The packets are modified: NAT changes either
  - source IP/port into NAT-IP/NAT-port
  - destination IP/port into NAT-IP/NAT-port

# No NAT



# NAT



# NAT: two uses

- Translate outgoing connections
  - Internal systems connecting to the outside
  - Change internal addresses into external address at the border
- Translate incoming connections
  - External systems connecting to an internal server
  - Change external address to the internal one

# NAT – outgoing/incoming

- NAT: outgoing
  - NAT component
    - sees an outgoing connection
    - selects a unique IP/Port combination
    - Systematically translates packets
  - Note: UDP is not that simple...
- NAT: incoming
  - NAT component
    - Sees incoming connection
    - Needs to know which internal address to translate to
    - Must be configured
    - Systematically translated packets
  - Note: think about ICMP too

# NAT – NAP**T**

- NAT: network address translation
- NAP**T**: network address and port translation
- NAP**T** is the norm, NAT is used as a simpler term

# NAT problems

- NAT has a series of known problems
- See RFC3027: Protocol Complications with the IP Network Address Translator
  - Higher level uses IP information
    - Authentication: must remain constant
    - Connect back via new connection
    - Port information
  - IPSec: protect IP header (see later)

# NAT ICMP problems

- RFC 3022: basic functions of NAT and NATP
- RFC 5508 (BCP 148): NAT Behavioral Requirements for ICMP
- To understand, need more detail of ICMP
  - Some ICMP packets contain identifier (request/reply style: ids must match)
  - Others, the error reporting, contain the “bad” packet header and some data

# NAT – ICMP

- For packets with an ID, the NAT box can assure external unique IDs and translate back
- For packets with the bad payload, this payload contains the connection data as seen externally, so they can be used to “reverse NAT” the message

# Proposed behavior of NAT for ICMP

- MUST support
  - Destination Unreachable Message
  - Time Exceeded Message
  - Echo Request/Reply Messages
- MAY support:
  - Redirect Message
  - Timestamp and Timestamp Reply Messages
  - Source Route Options,
  - Address Mask Request/Reply Message
  - Parameter Problem Message
  - Router Advertisement and Solicitations
- SHOULD NOT support:
  - Source Quench Message
  - Information Request/reply

# In practice...

- ICMP Message traversal behavior on a NAT device may be overridden by local administrative policies (like security)
- For instance:
  - prohibit forwarding of ICMP Error messages across a NAT device
  - prohibit ICMP Query based applications across a NAT device

# **NODE CONFIGURATION**

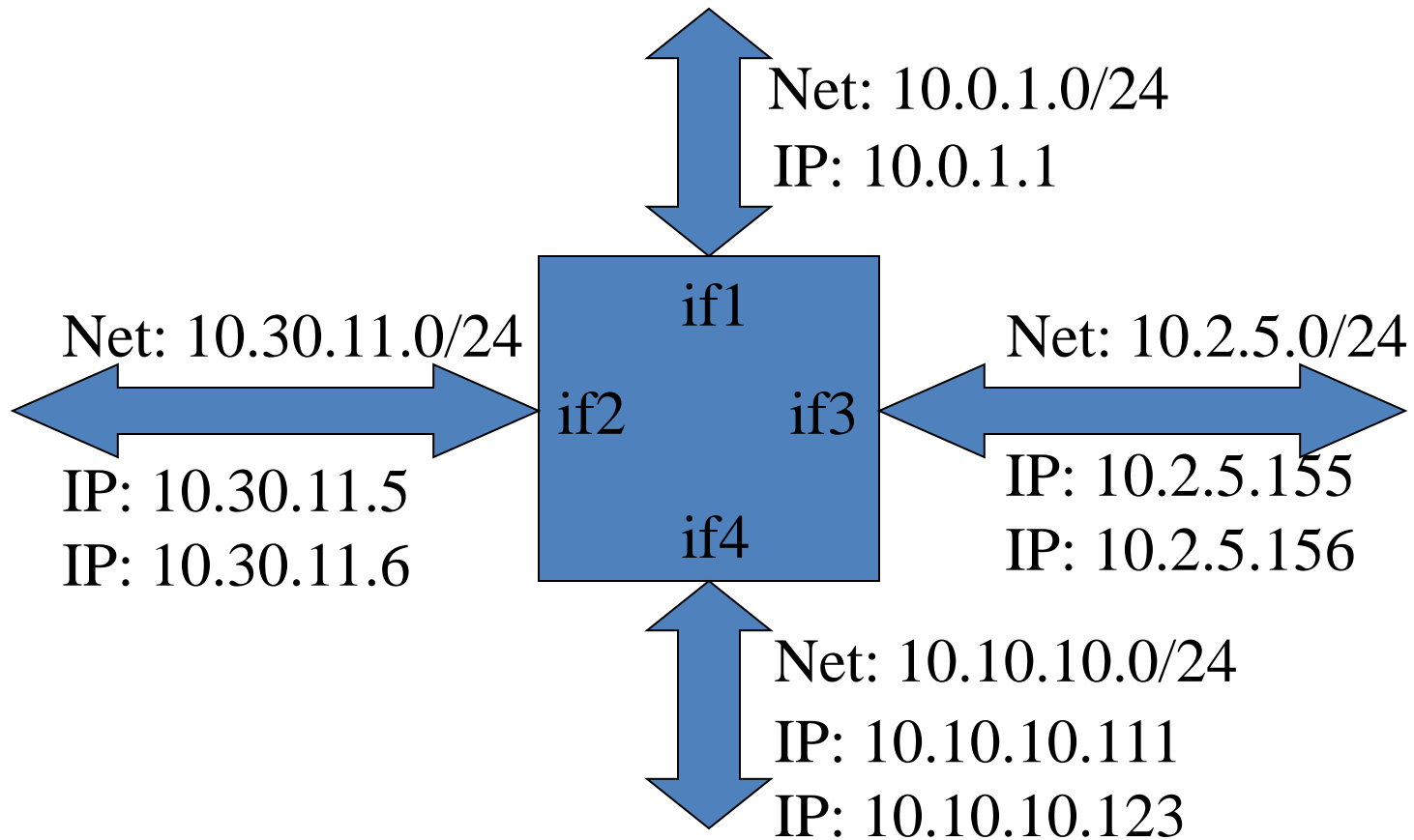
# Node configuration

- What must be configured?
  - Interfaces, ARP, routing ... lots of stuff
- Configuring interfaces
  - node with multiple network interfaces (Ethernet cards, serial links, X.25 connections, ...)
  - interface namespace
  - IP - interface mapping: connected subnet definition
    - one (or more) IP address per interface
    - subnet mask per interface

# Multiple IP per interface

- multi-homed system: on same interface three IP addresses
  - IP 212.7.34.1
  - IP 212.7.34.2
  - IP 212.7.34.3
- software can distinguish the IP address used to connect

# Interfaces, netmasks, IPs



# Multiple interfaces

- Multiple interfaces more and more the norm
- Example:
  - Ethernet connection
  - Wireless connection
  - Bluetooth connection
  - Infrared connection
  - Modem connection

# Interfaces: special situations

- An interface can be linked to multiple networks
  - Unrelated
  - Share underlying network
  - Multiple IPs per network
- An interface can have no IP address (stealth set-up)

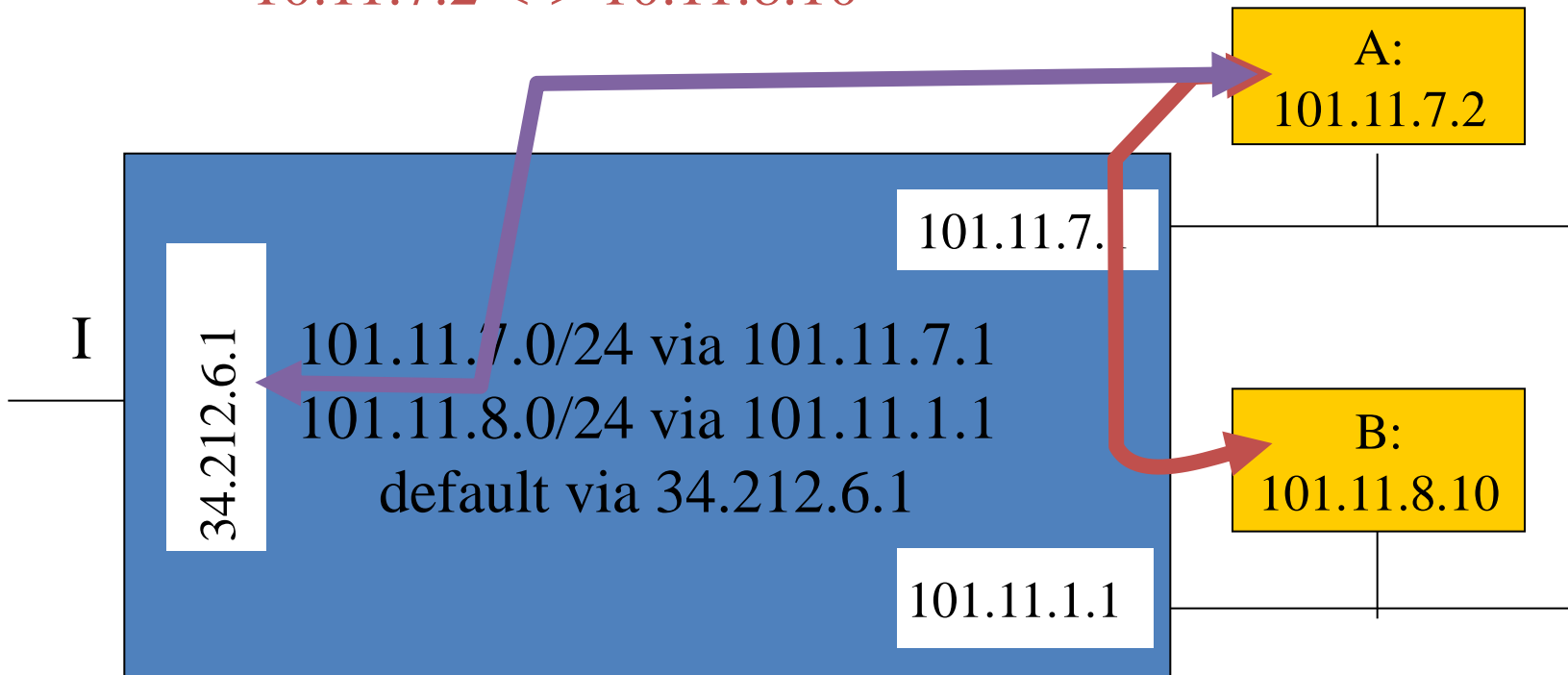
# Configuring ARP

- Multiple Ethernet interface implies multiple subnets
- The host needs to know on which interface which MAC reside

# Multiple interfaces

101.11.7.2 <-> 125.71.50.66

10.11.7.2 <-> 10.11.8.10



# Exercise

- RFC 1027 Gateway - switch
- RFC structure
  - Status of this memo
  - Acknowledgement
  - Introduction
  - Design
    - Basic method
    - Routing
    - Multiple gateways

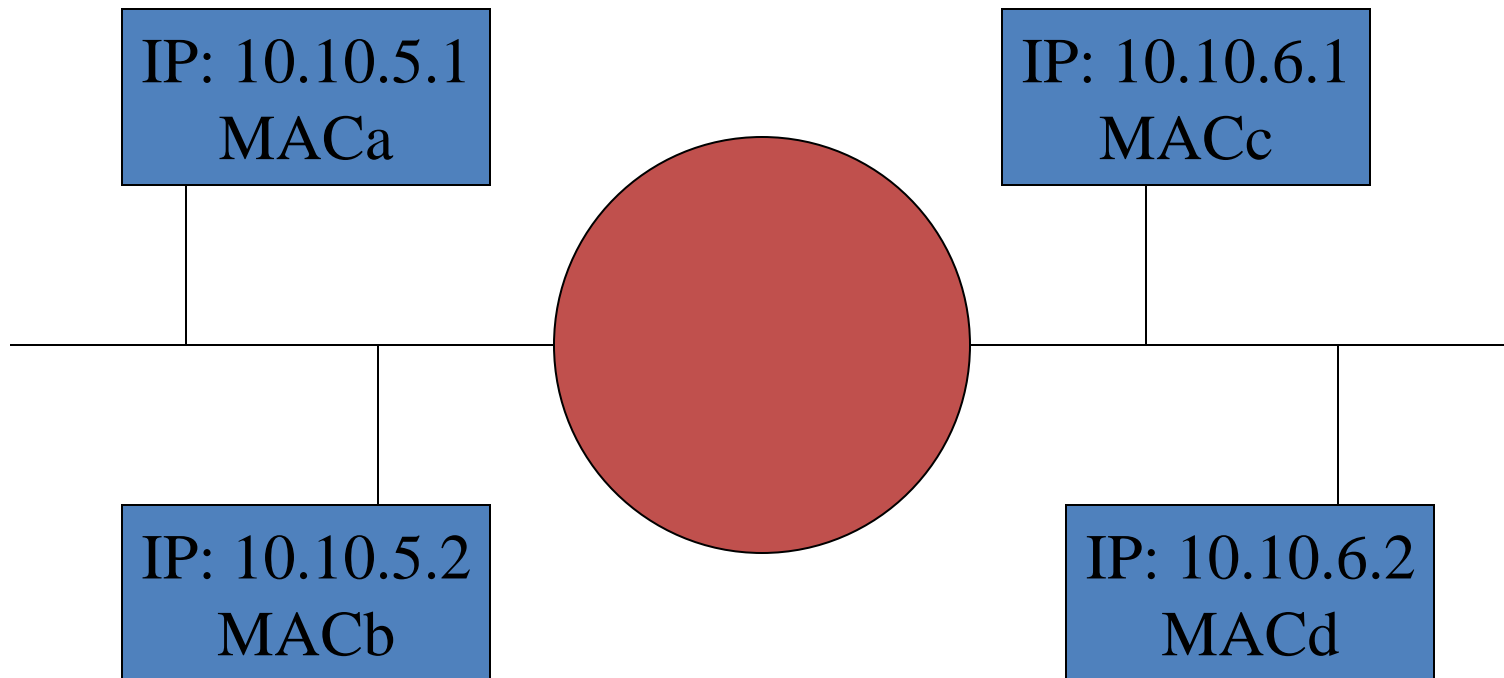
# Fragments from RFC

- “Assuming that subnet numbers are made to correspond to physical networks”
- “all ARP subnet handling is done in the ARP subnet gateways”
- “when an ARP request is seen, the ARP subnet gateway can determine whether it knows a route to the target host by looking in the ordinary routing table”

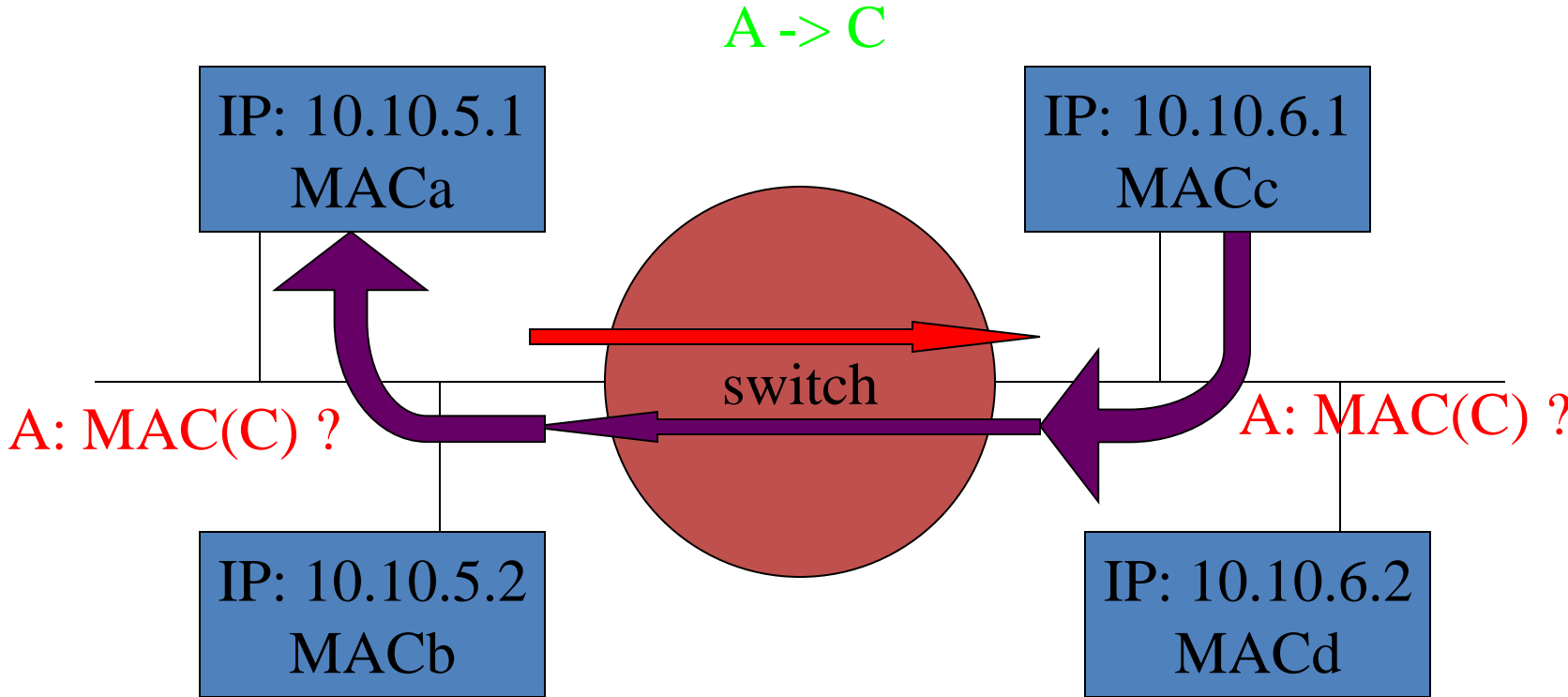
# Transparent separation of subnet

- Two options:
  - RFC 1027: transparent gateway
  - Switch
- RFC 1027
  - Proxy assumes MACs of all IPs right vis-a-vis left, and all MACs of all IPs left vis-a-vis right
  - Question: which IPs are ‘left’, which are ‘right’?
    - Based on subnet
      - Systems need not know this
    - Based on configuration

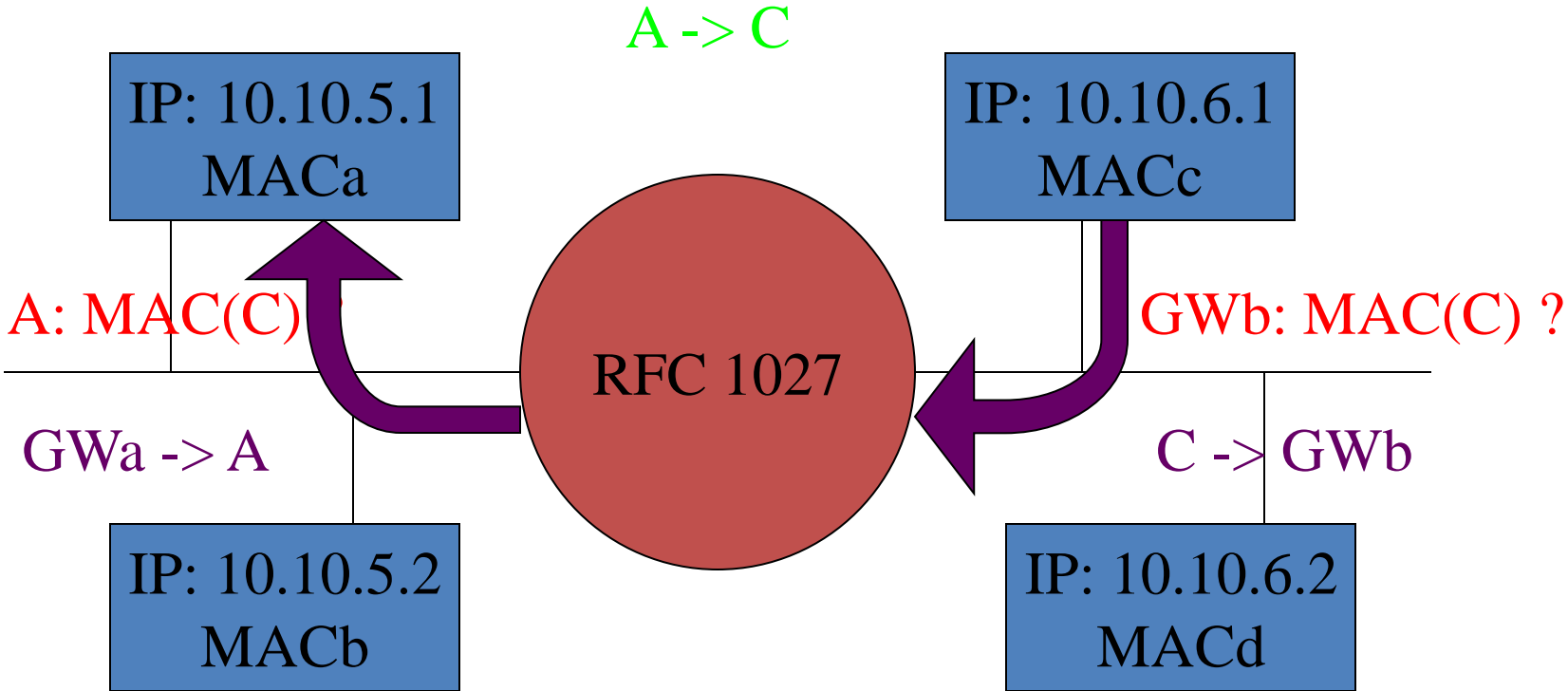
# Diagram 1



# Diagram 2



# Diagram 3



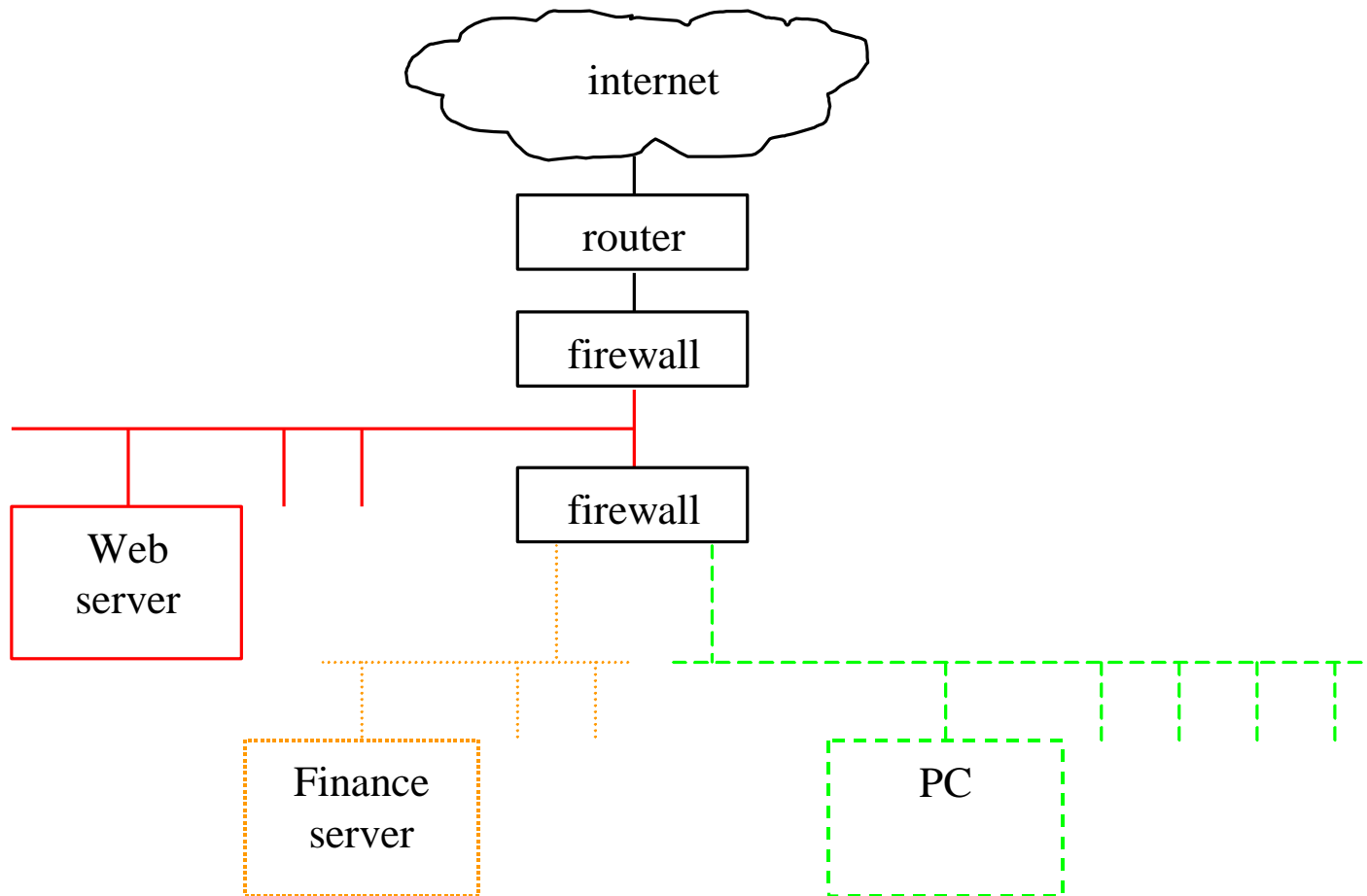
# Exercise

- RFC 925
- Multi-LAN address resolution
- Introduces “magic box” connected to two or more LANs
- Maintains IA:HA pairs
- Problems
  - Bad cache entries
  - Infinite transmission loops

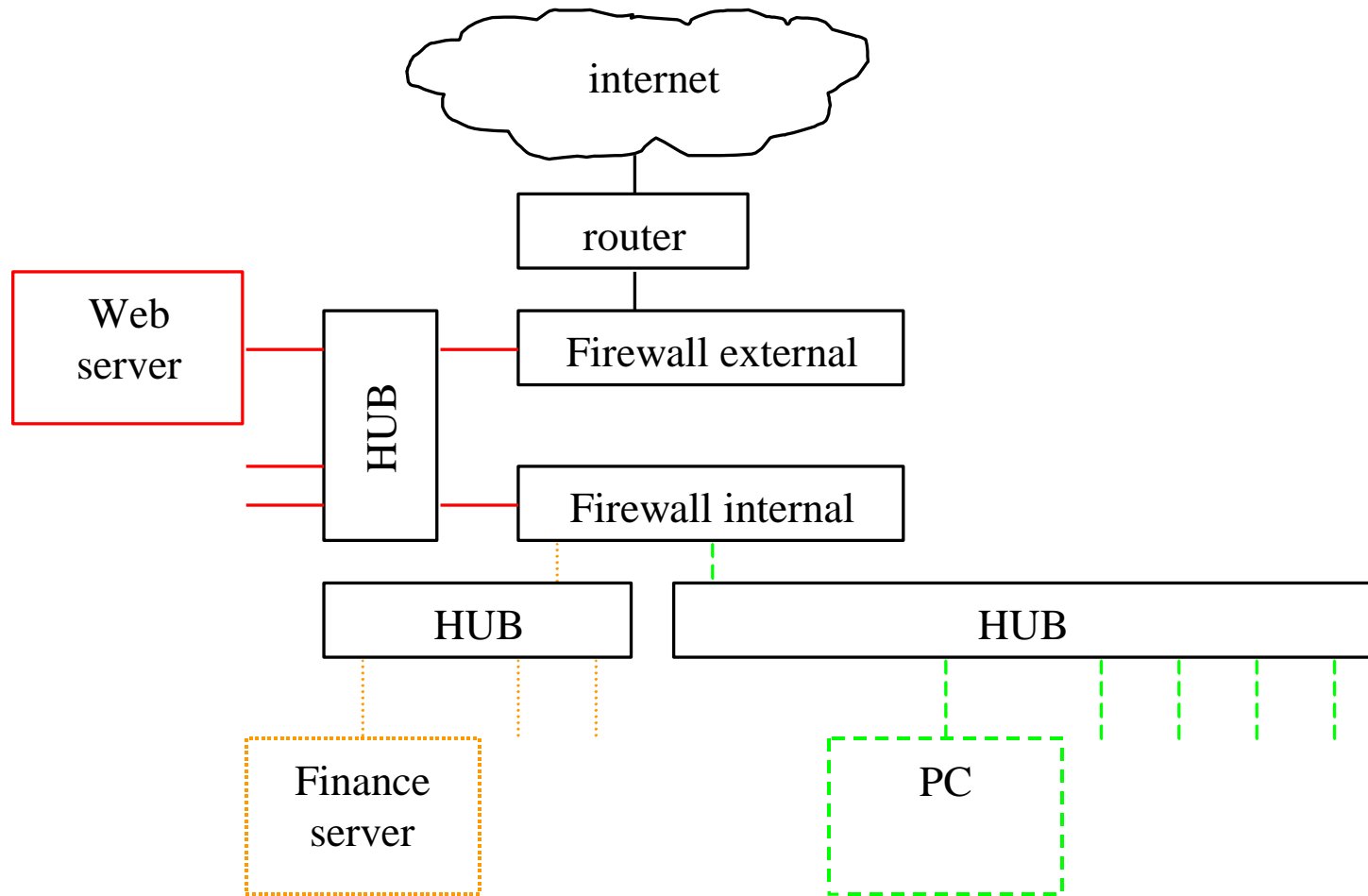
# Exercise

## Configuration

# Logical scheme

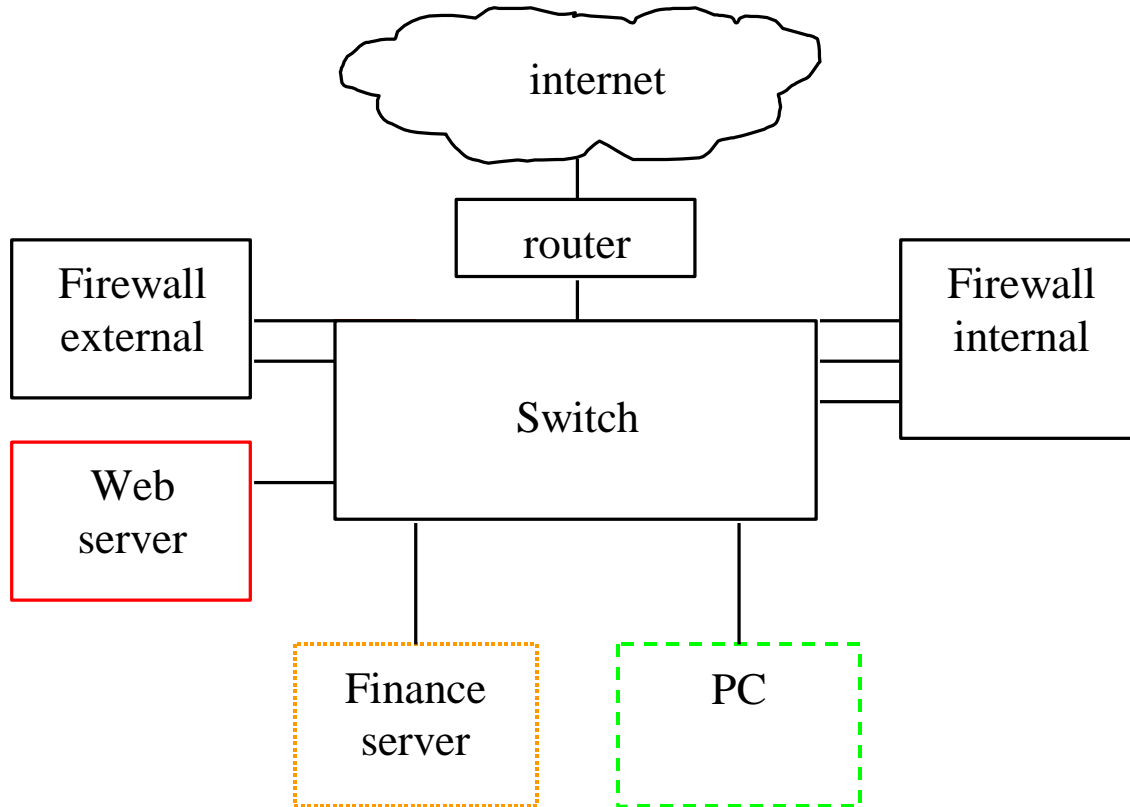


# Physical, HUBs



# Physical, Switch

VLANs – LANs – IP – routing ?



DHCP

# **AUTO CONFIGURATION**

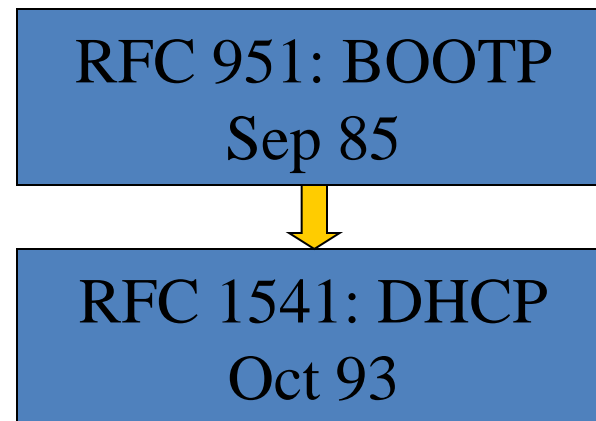
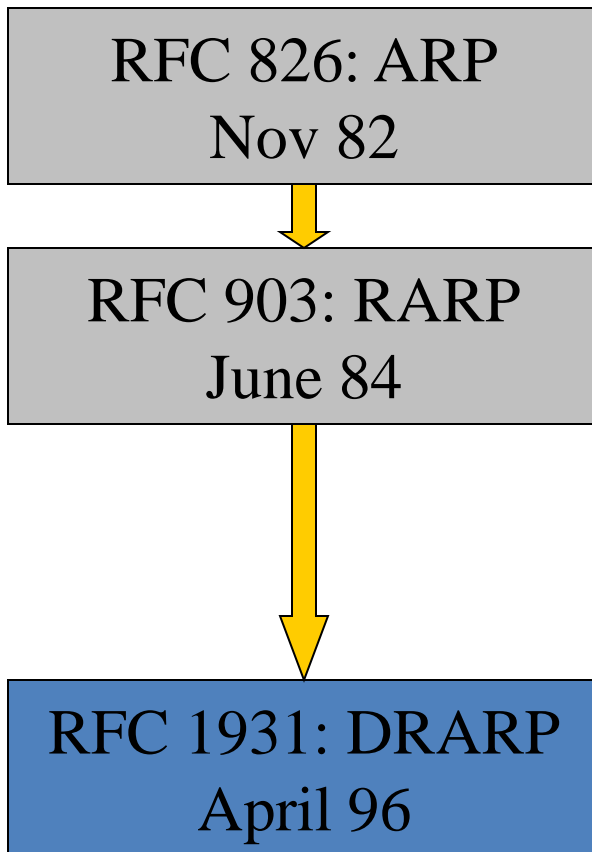
# Auto Configuration: problem

- IP configuration: settings for many parameters
- values for those parameters:
  - There are no obvious defaults
  - need to be specific for infrastructure
- Mobile solutions (portables)
  - Reconfigure per connection point
  - “Technically challenged”

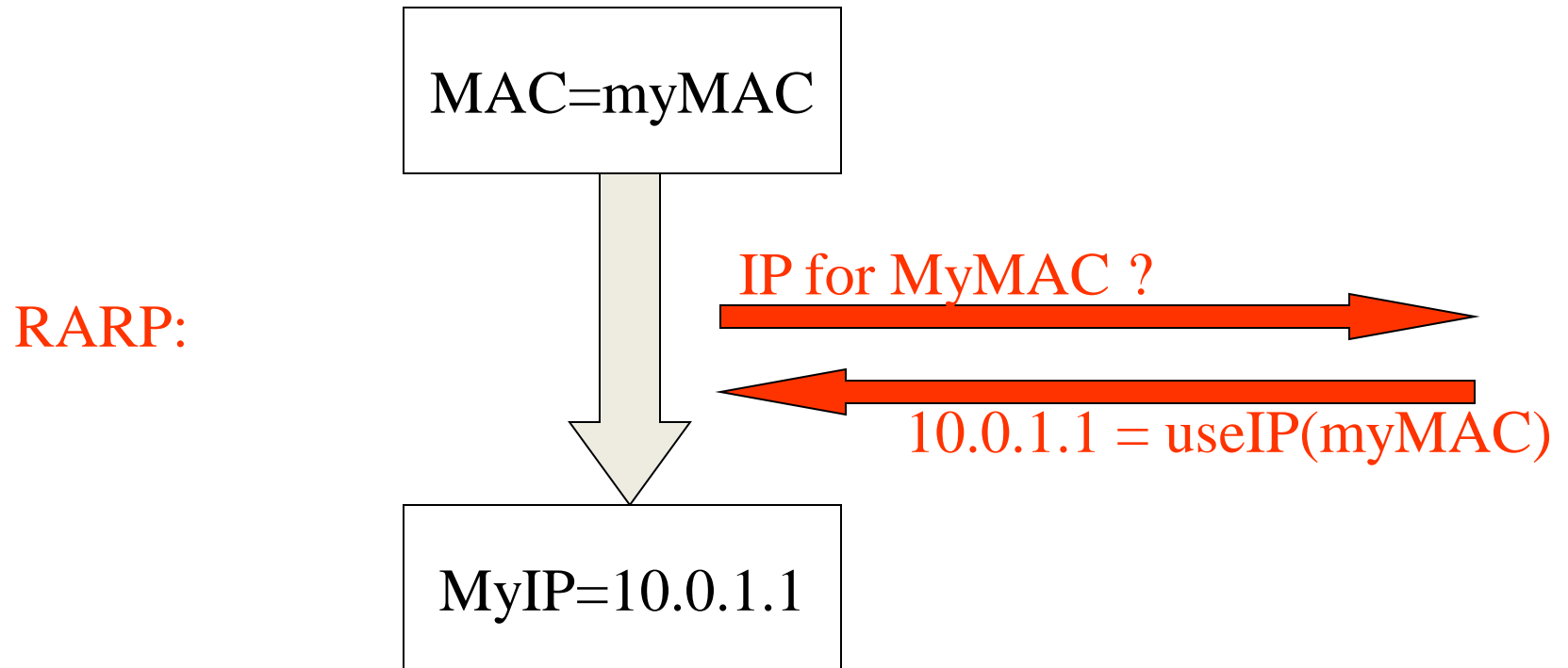
# Auto Configuration: “extinct” alternatives

- Dynamic RARP (DRARP)
- Trivial File Transfer Protocol (TFTP)
- Internet Control Message Protocol (ICMP)
  - additional routers: "ICMP redirect"
  - subnet mask: "ICMP mask request"
- Locate routers through the ICMP router discovery mechanism
- BOOTP

# Auto Configuration: RFCs



# RARP



# BOOTP

- Defined in RFC 951
- Uses IP (network layer), hence routable
- Compare: (D)RARP: link layer
- Now obsolete: replaced by DHCP, RFC 1541

# DHCP

- RFC 2131: Dynamic Host Configuration Protocol
- Two components:
  - a protocol for delivering host-specific configuration parameters from a DHCP server to a host
  - a mechanism for allocation of network addresses to hosts

# DHCP: client - server

- Follows client-server model
- Uses designated DHCP servers
  - allocate network addresses
  - deliver configuration parameters to dynamically configured hosts
- DHCP uses UDP as its transport protocol.
  - client to server: port 67
  - server to client: port 68

# DHCP: IP to host mapping

- Automatic: permanent assignment
  - usage: desktops
- Dynamic: limited period of time (lease period)
  - the host may explicitly relinquishes the address
  - usage: laptops, ipads, notebooks, ...
- Manual: IP address is assigned by the network administrator
  - usage: servers

# DHCP - BOOTP

- Relations
  - BOOTP inheritance:
    - DHCP can use BOOTP relay: RFC 951
    - DHCP uses BOOTP message format defined in RFC 951
    - DHCP must provide service to existing BOOTP clients
- DHCP - BOOTP: primary differences
  - DHCP clients
    - can be assigned a network address for a fixed lease
    - serial reassignment of network addresses to different clients
    - can acquire all of the IP configuration parameters
  - DHCP: explicit client identifier (may be the hardware address, or a DNS name)

# DHCP: the basics

- Client: requests an address for some period of time (lease period)
- Server: allocation mechanism guarantees
  - address not re-allocated within lease period
  - returns same IP address to the same client
- Clients may
  - refresh lease
  - release the address
  - ask infinite lease

# DHCP: Client-Server Protocol

- uses BOOTP message format defined in RFC 951
- two types of messages:
  - client->server: BOOTREQUEST
  - server->client: BOOTREPLY

# Messages: client to server

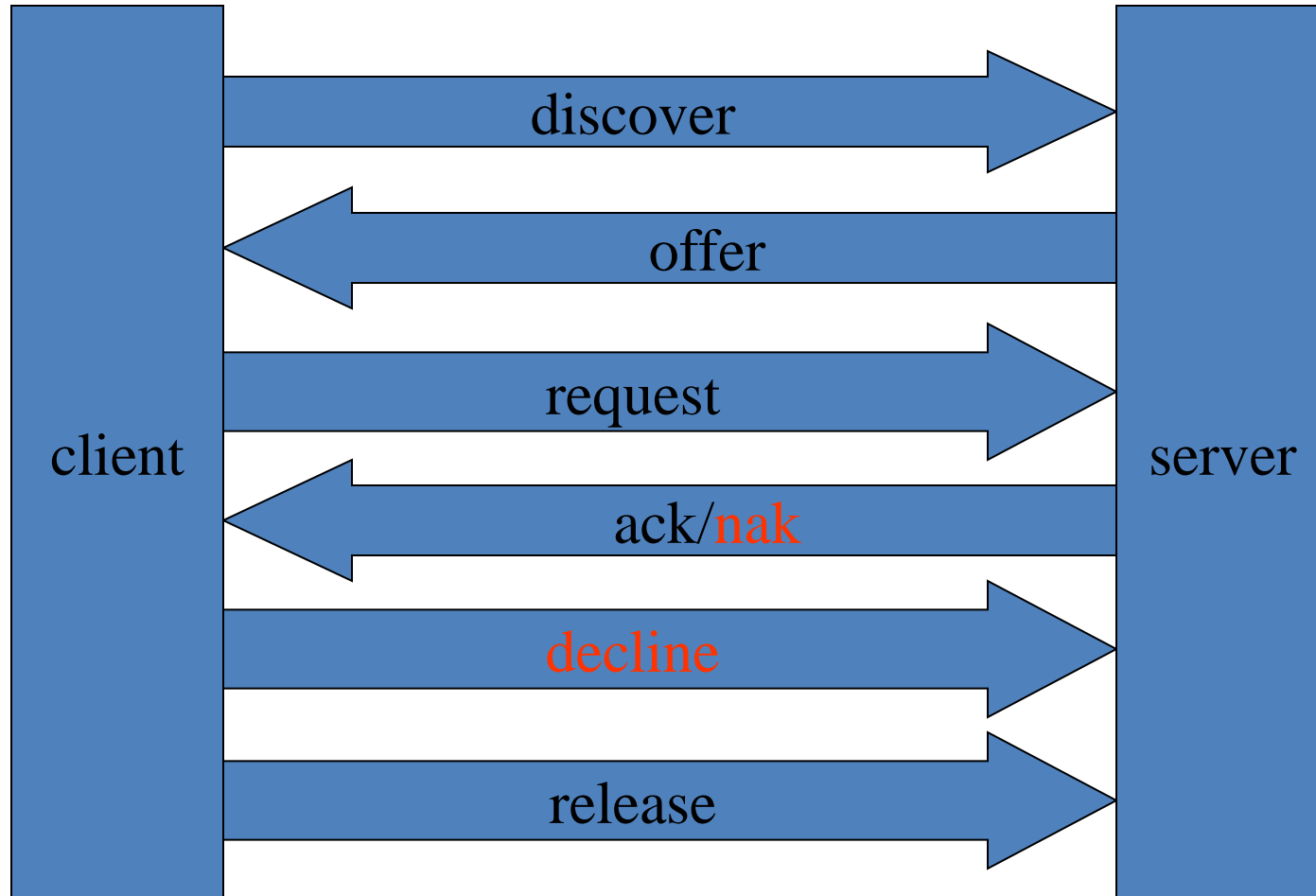
## Client to server:

- DHCPDISCOVER: broadcast to locate available servers
- DHCPREQUEST: broadcast requesting offered parameters from one server (implicitly declining offers from all others)
- DHCPDECLINE: received configuration invalid
- DHCPRELEASE: relinquishing IP address (also: canceling remaining lease)

## Server to client

- DHCPOFFER: in response to DHCPDISCOVER with offer of configuration parameters.
- DHCPACK: configuration parameters, including committed network address.
- DHCPNAK: refusing request for configuration parameters (e.g., requested network address already allocated).

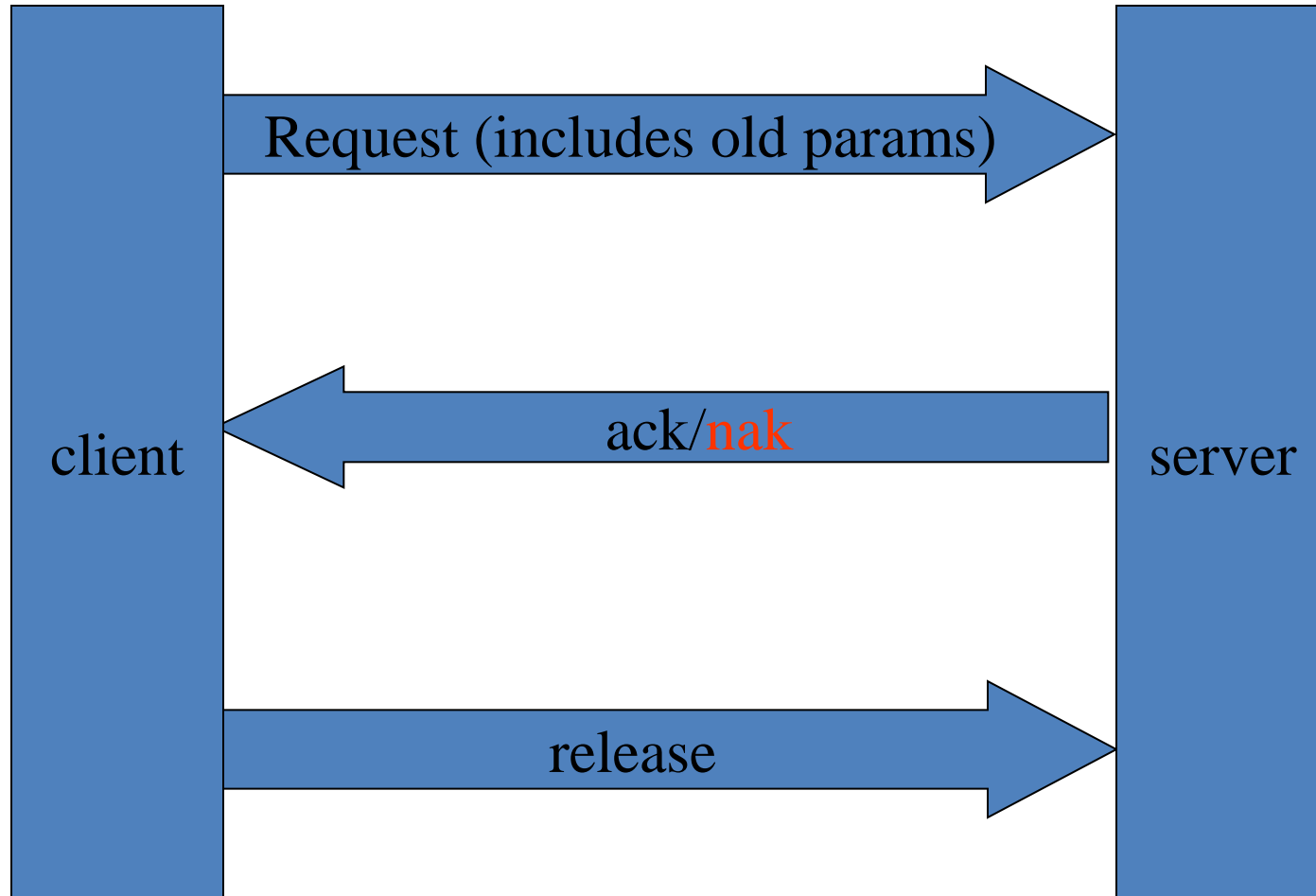
# Protocol scheme



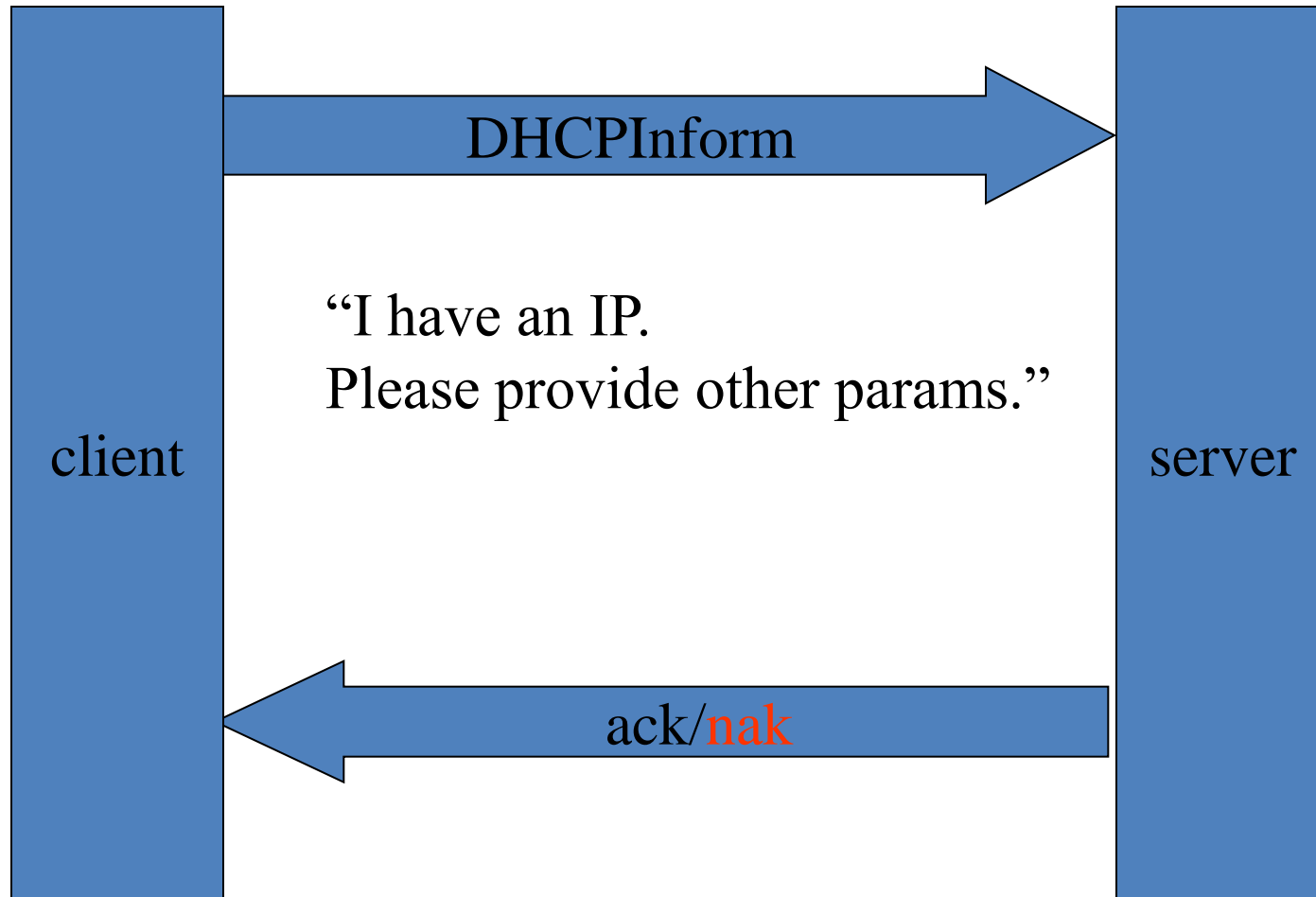
# Message essentials

- operation:1 = REQUEST, 2 = REPLY
- xid: Transaction ID, a random number, used by the client and server
- ciaddr: Client IP; filled in by client when verifying previous parameters.
- yiaddr: 'your' (client) IP address
- giaddr: Relay agent IP address
- chaddr: Client hardware address

# Protocol: "reuse" scheme



# Protocol: additional info scheme



# RFC requirement

- “DHCP should not require a server on each subnet. To allow for scale and economy, DHCP must work across routers or through the intervention of BOOTP relay agents. “
- Consequence:
  - use IP-based protocol?
  - Relay agents?
- Chicken&Egg problem
  - How can the server send an IP datagram to the client, if the client doesn't know its own IP address (yet)?

# Chicken&Egg solution

- Client knows its own IP address:
  - normal IP: client will respond to ARPs
- Client does not yet know its IP address
  - The client cannot respond to ARPs, two options:
    - 'manually' construct an ARP address cache entry
    - fill in an entry using the 'chaddr' and 'yiaddr' fields
    - send the bootreply to the client's IP address
  - Send the bootreply to the IP broadcast address on the appropriate interface

# Extra message: DHCPINFORM

- There is one more message: DHCPINFORM
- Introduced in RFC upgrade
- DHCPINFORM (from RFC 2131):
  - “Client to server, asking only for local configuration parameters; client already has externally configured network address.”

# DHCP security

- “A host should not act as a DHCP server unless explicitly configured to do so by a system administrator. The diversity of hardware and protocol implementations in the Internet would preclude reliable operation if random hosts were allowed to respond to DHCP requests. “
- What do you think if you are a hacker?

# Security of DHCP

- DHCP is built directly on UDP and IP
- UDP, IP: inherently insecure
- DHCP is quite insecure:
  - Pirate DHCP servers: easy
    - Sometimes systems assume role as DHCP server by default: can wreak havoc on a LAN
  - Malicious DHCP clients:
    - masquerade as legitimate clients
    - retrieve information intended for legitimate clients.
- See also
  - RFC 3118: Authentication for DHCP Messages

# How to abuse DHCP

- Give a wrong IP?
  - So that you can use his?
- Give a wrong subnet?
  - So that “some” communication works/does not work?
- Give a wrong gateway?
  - Like, a node under your control?
  - Like, an insecure path versus a secure one?
- Give a wrong name server?
  - Maybe one you control?
  - Maybe see the requests ?
  - Maybe MitM on certain sites?

# Configuration

- Host: DNS, routing, TCP configuration, ...
  - Interface 1: MAC, IP config: netmask 1, ...
    - IP 1
    - IP 2
  - Interface 2: MAC2, IP config: netmask 2, ...
    - IP 3
    - IP 4

# Parameters: interface & host

- IP-layer, per interface
  - IP address (address)
  - Subnet mask (address mask)
  - MTU (integer)
  - Perform router discovery (on/off)
  - Router solicitation address (address)
- IP-layer, per host
  - Be a router (on/off)
  - Maximum reassembly size (integer)
  - Default TTL (integer)

# Parameters


- Link-layer, per interface:
  - Trailers (on/off)
  - ARP cache timeout (integer)
  - Ethernet encapsulation (RFC 894/RFC 1042)
- TCP, per host:
  - TTL (integer)
  - Keep-alive interval (integer)
  - Keep-alive data size (0/1)


# Parameters: routing


- List of Default routers
  - router address (address)
  - preference level (integer)
- List of static routes
  - destination (host/subnet/net)
  - destination mask (address mask)
  - type-of-service (integer)
  - first-hop router (address)


# **DHCP GATEWAY**

# Scheme

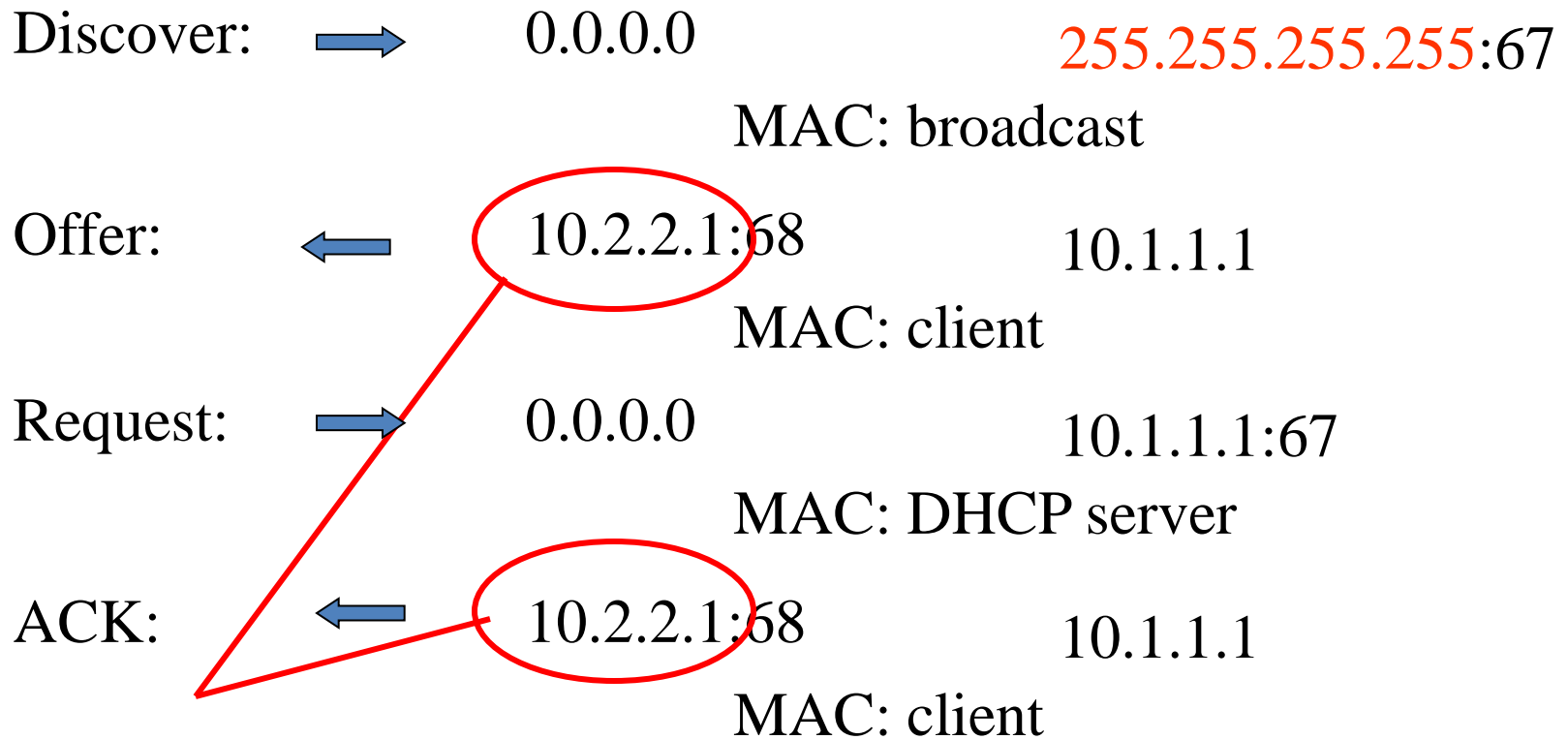
Discover:  0.0.0.0 255.255.255.255:67  
MAC: broadcast

Offer:  10.2.2.255:68 10.1.1.1  
MAC: broadcast

Request:  0.0.0.0 10.1.1.1:67  
MAC: DHCP server

ACK:  10.2.2.255:68 10.1.1.1  
MAC: broadcast

# Actual scheme



Client must listen on any IP

# DHCP relay

- Problems:
  - too many broadcast
  - Source 0.0.0.0, destination 255.255.255.255: BAD
- Solution: DHCP relay
  - Listens on local LAN
  - Captures DHCP requests
  - Turns them into normal UDP communication, including information on source LAN
  - When receiving answers, dispatches them locally

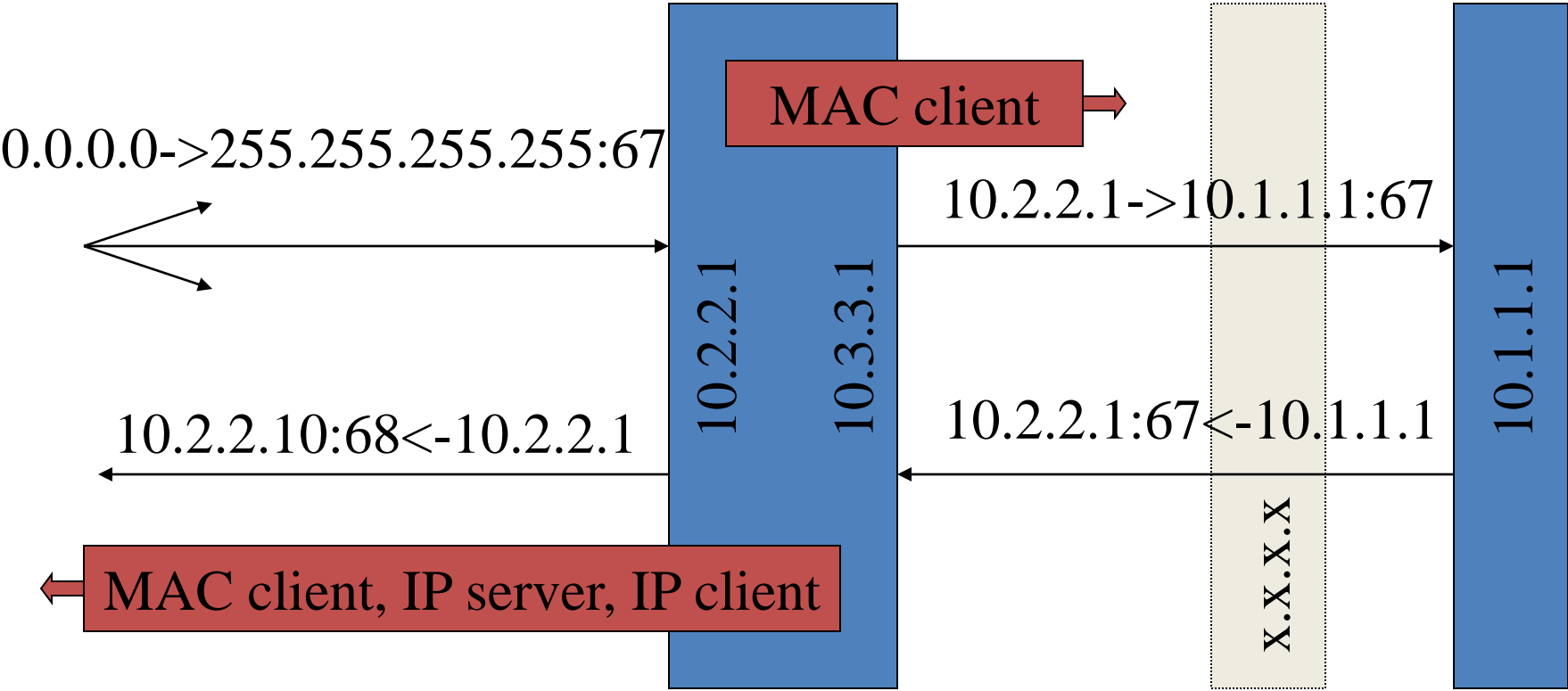
# Fields in packages

- Fields in packages
  - c i addr: client IP address
  - y i addr: your IP address (assigned address)
  - s i addr: server IP address
  - g i addr: gateway IP address
  - c h addr: client hardware address
- main function:
  - yiaddr = dhcp(chaddr)
- more detail:
  - yiaddr = dhcp(chaddr, siaddr [,ciaddr][,broadcast]

# Use of fields

- Server tells
  - your IP address: yiaddr
  - its address: siaddr
- Client tells
  - address it wants/had last time: ciaddr
  - please send reply as broadcast: broadcast flag

# DHCP relay



# Gateway address usage in request

Giaddr:

- If giaddr == 0 (0.0.0.0)
- Then
  - the relay agent **MUST** fill this field with the IP address of the interface on which the request was received.
- Else
  - the 'giaddr' field **MUST NOT** be modified.

# The server reply handling

- If `giaddr != 0.0.0.0` Then
  - send the BOOTREPLY as an IP unicast to '`giaddr`':67
  - that relay agent will then perform the final delivery to the client.
- Else If broadcast flag on Then
  - use IP broadcast 255.255.255.255:68
  - use link layer broadcast
  - Else
    - use IP unicast to `yiaddr:68` (assigned/proposed IP address)
    - use link layer unicast to `chaddr` (client hardware address == MAC)

# IP MULTICAST

# IP Multicast: Motivation

- There are many situations where multiple clients are interested in the same information.
- The mechanism that is used most often is called publish – subscribe.
- A publisher pushes information out, and subscribers can register to receive it.
- The sender could maintain a subscription list and send the packet to each and everyone of the subscribers. This is how mailing lists operate, for instance.
- On the other hand, packets could be broadcasted, so that multiple hosts could read the same message at the same time, leading to a much increased performance.

# Definition

- IP Multicast:
  - RFC 1112: “Host Extensions for IP Multicasting”:
- “IP multicasting is the transmission of an IP datagram to a "host group", a set of zero or more hosts identified by a single IP destination address. A multicast datagram is delivered to all members of its destination host group with the same "best-efforts" reliability as regular unicast IP datagrams”

# Sending

- IP level: a set of addresses is reserved for multicast use.
- Anyone can send a multicast to one of the addresses.
- The addresses are defined in two sets:
  - sets which are globally managed by IANA
  - addresses for occasional, specific local use.
- Multicast addresses are those addresses in the range 224.0.0.0 to 239.255.255.255.
  - The address 224.0.0.0 means “nobody”, and 224.0.0.1 means “all”.
- A list of assignments can be found at:  
<http://www.iana.org/assignments/multicast-addresses>

# Some examples:

- 224.0.0.0 Base Address
- 224.0.0.1 All Systems on this network
- 224.0.0.2 All Routers on this Subnet
- 224.0.0.4 DVMRP Routers
- 224.0.0.5 OSPFIGP OSPFIGP All Routers
- 224.0.0.6 OSPFIGP OSPFIGP Designated Routers
- 224.0.0.9 RIP2 Routers
- 224.0.0.10 IGRP Routers
- 224.0.0.12 DHCP Server / Relay Agent
- 224.0.0.16 designated-sbm
- 224.0.0.17 all-sbms
- 224.0.0.18 VRRP
- 224.0.0.22 IGMP **Receiving**

# Groups

- Client can dynamically join or leave a group.
- There is no server involved.
- Clients can subscribe to any number of groups.
- Host group: permanent or transient.
  - Permanent: well-known, administratively assigned IP address.
  - The address is permanent.

# Ethernet link

- Ethernet supports multicast
- There is no need to use Ethernet broadcast.
- IP multicast over Ethernet:
  - the 23 low-order bits of the IP Multicast address go in the low-order 23 bits of the Ethernet multicast
  - address 1.0.94.0.0.0. (01-00-5E-00-00-00).
- See
  - <http://www.iana.org/assignments/ethernet-numbers> for other multicast addresses.
  - The STP for instance uses 01-80-C2-00-00-00.

# Routing

- See RFC 3376: Internet Group Management Protocol, Version 3.
- The Internet Group Management Protocol (IGMP) provides multicast routers with information on group membership. Multicast routers request membership information, and nodes respond with a membership report.
- Multicast routers can use the Distance Vector Multicast Routing Protocol (DVMRP) protocol to route multicast messages.
- See: <http://www.ietf.org/internet-drafts/draft-ietf-idmr-dvmrp-v3-11.txt>

# Switches revisited: attack - mitigation

- Layer 2 Network Protections against Man in the Middle Attacks
  - <http://isc.sans.org/diary.html?storyid=7567>

# Layer 2 MITM attacks:

- often based on ARP poisoning
  - attacker sending an unsolicited ARP reply to the target hosts
  - target client and server both think that the attacker is the host at the other end of the conversation.
- attacker will now intercept all traffic between the hosts
  - which can be simply recorded and forwarded on
  - modified before forwarding.
- mitigation?
  - DHCP Snooping
  - Dynamic ARP inspection
  - IP Source Guard

# DHCP Snooping

- default: blocks all DHCP offer packets inbound to the switch port
- consequence: if a DHCP server is on that port, the DHCP requests will reach the server, but the replay offers will never reach the DHCP clients
- must configure any ports that have DHCP servers attached, or uplink to switches with DHCP servers as “trusted”.

# Dynamic ARP inspection (DAI):

- uses the table created by the DHCP Snooping feature to validate all ARP responses that arrive inbound to switch ports
- DAI:
  - drops all ARP replies that do not have corresponding entries in the DHCP Snooping binding database.
  - ARP reply packets where the MAC address in the body of the packet do not match the MAC address in the Ethernet header are also dropped.
- non DHCP environments?
  - DAI operates against a statically defined ARP ACL.
  - Ports that have multiple MAC addresses, such as uplink to physical or virtual switches, should be configured as “trusted”, this bypasses all DAI functions.

# IP Source Guard:

- uses the DHCP Snooping database
- When a client host powers on, IP Source Guard will filter all traffic to and from that port except for the DHCP request and reply traffic.
- Once the address is assigned and the DHCP Snooping entry is populated for the port, any traffic received from that port from a different IP address is filtered.

# Caveats and Comments on Layer 2 Network Protections

- Many network components require on changing MAC addresses!
  - protections against ARP poisoning may kill how they work
- Disable protective features on interfaces that:
  - Have components with standby features such as vrrp, vrrp-e, hsrp and similar router or firewall clustering applications
  - Connect to load balancers
    - Including hosts configured with load balancing solutions such as Microsoft NLB
  - Run any clustering application such as Microsoft, Linux, Solaris or other clustering solutions
  - connect to VMware ESX vswitch uplink ports
    - a vmotion involves a virtual server MAC migrating from one physical switch port to another
  - Switch ports that connect to ESX uplink ports that support load balanced or failover configurations for the ESX Service Console or vmkernel ports.

# References

- RFC 1533: DHCP Options and BOOTP Vendor Extensions
- RFC 1122: Requirements for Internet Hosts -- Communication Layers
- RFC 1123: Requirements for Internet Hosts -- Application and Support
- RFC 783: The TFTP Protocol (Revision 2)
- RFC 1931: Dynamic RARP
- RFC 2131: Dynamic Host Configuration Protocol (DHCP)
- RFC 951: Bootstrap Protocol (BOOTP)
- Radia Perlman, Interconnections, second edition  
bridges, routers, switches, and Internetworking Protocols  
ISBN 0201634481

# References

- TCP/IP
  - RFC 791: Internet Protocol
  - RFC 792: Internet Control Message Protocol
  - RFC 793: Transmission Control Protocol
  - RFC 768: User Datagram Protocol
  - TCP/IP Illustrated, Volume 1: The Protocols, Addison-Wesley, 1994, ISBN 0-201-63346-9.
  - TCP/IP Illustrated, Volume 2: The Implementation, Addison-Wesley, 1995, ISBN 0-201-63354-X.
  - TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP, and the UNIX Domain Protocols, Addison-Wesley, 1996, ISBN 0-201-63495-3.

# RFCs

- RFC 1027: Using ARP to Implement Transparent Subnet Gateways
- RFC 925: Multi-LAN Address Resolution
- RFC 826: ARP
- RFC 903: RARP
- RFC 1868: UNARP
- RFC 814: Names, Addresses, Ports, and Routes.
- RFC 917: Internet Subnets
- RFC 919: Broadcasting Internet Datagrams.