Mobilitate la nivel rețea

- internet: Mobile IP
- local: Zeroconf

Motivation for Mobile IP

- Routing
  - IP destination address, network prefix => physical subnet
  - change of physical subnet => change of IP address
    - or needs special entries in the routing tables
- Specific routes to end-systems?
  - change of all routing tables toward the right destination
  - does not scale with
    - number of mobile hosts
    - frequent changes in the location
    - security problems

Motivation for Mobile IP

- Changing the IP-address?
  - Adjust host IP address depending on the current location
  - hard to find a mobile system, DNS updates take too long
  - TCP connections break
  - security problems

- IP address is both
  1. location identifier
  2. host identity

Design bug!

Mobile IPv4 2002->3220->3344->4721->5944

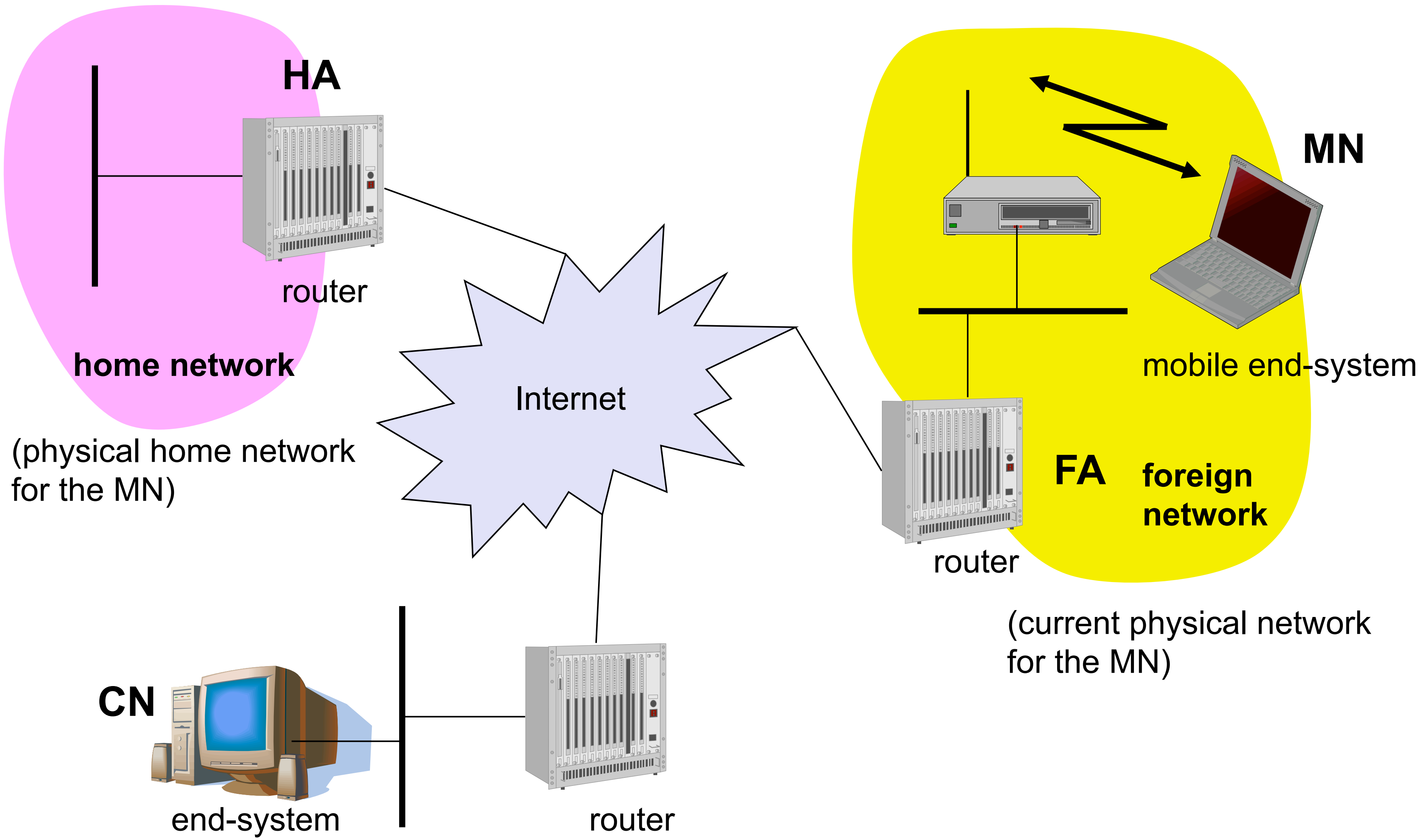
- **Transparency**
  - mobile end-systems keep their IP address
  - continuation of communication after interruption of link
  - point of connection to the fixed network can be changed
- **Compatibility (wished)**
  - support of the same layer 2 protocols as IP
  - no changes to current end-systems and routers
  - mobile can communicate with fixed systems
- **Security**
  - authentication of all registration messages
- **Efficiency and scalability**
  - little additional messages to the mobile system required
  - world-wide support

Terminology

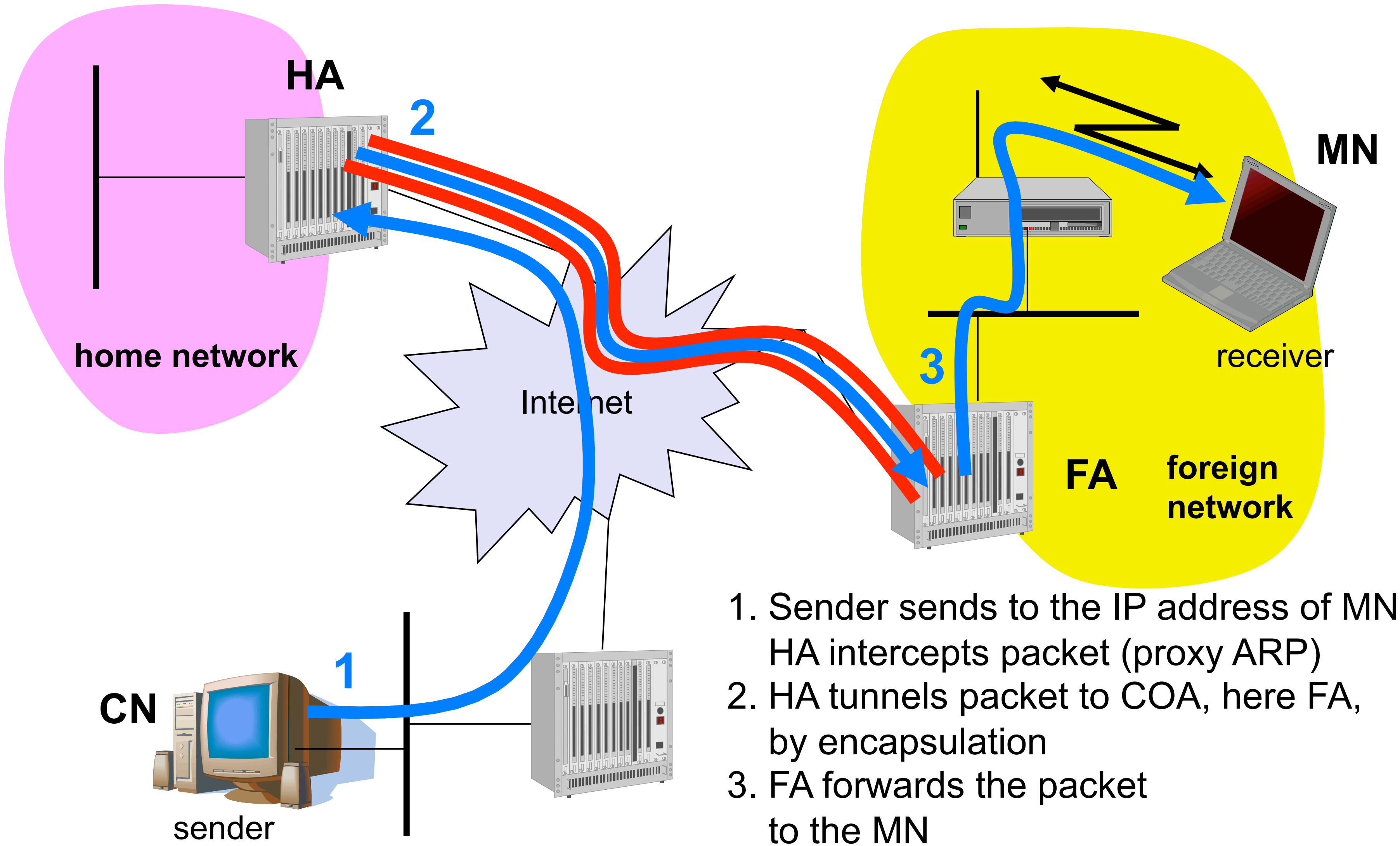
- Mobile Node (MN)
  - system (node) that can change the point of connection to the network without changing its IP address

- Home Agent (HA)
  - system in the home network of the MN, typically a router
  - registers the location of the MN, tunnels IP datagrams to the COA
- Foreign Agent (FA)
  - system in the current foreign network of the MN, typically a router
  - forwards the tunneled datagrams to the MN, typically also the default router for the MN
- Care-of Address (COA)
  - address of the current tunnel end-point for the MN (at FA or MN)
  - actual location of the MN from an IP point of view
  - can be chosen, e.g., via DHCP
- Correspondent Node (CN)
  - communication partner

Example network

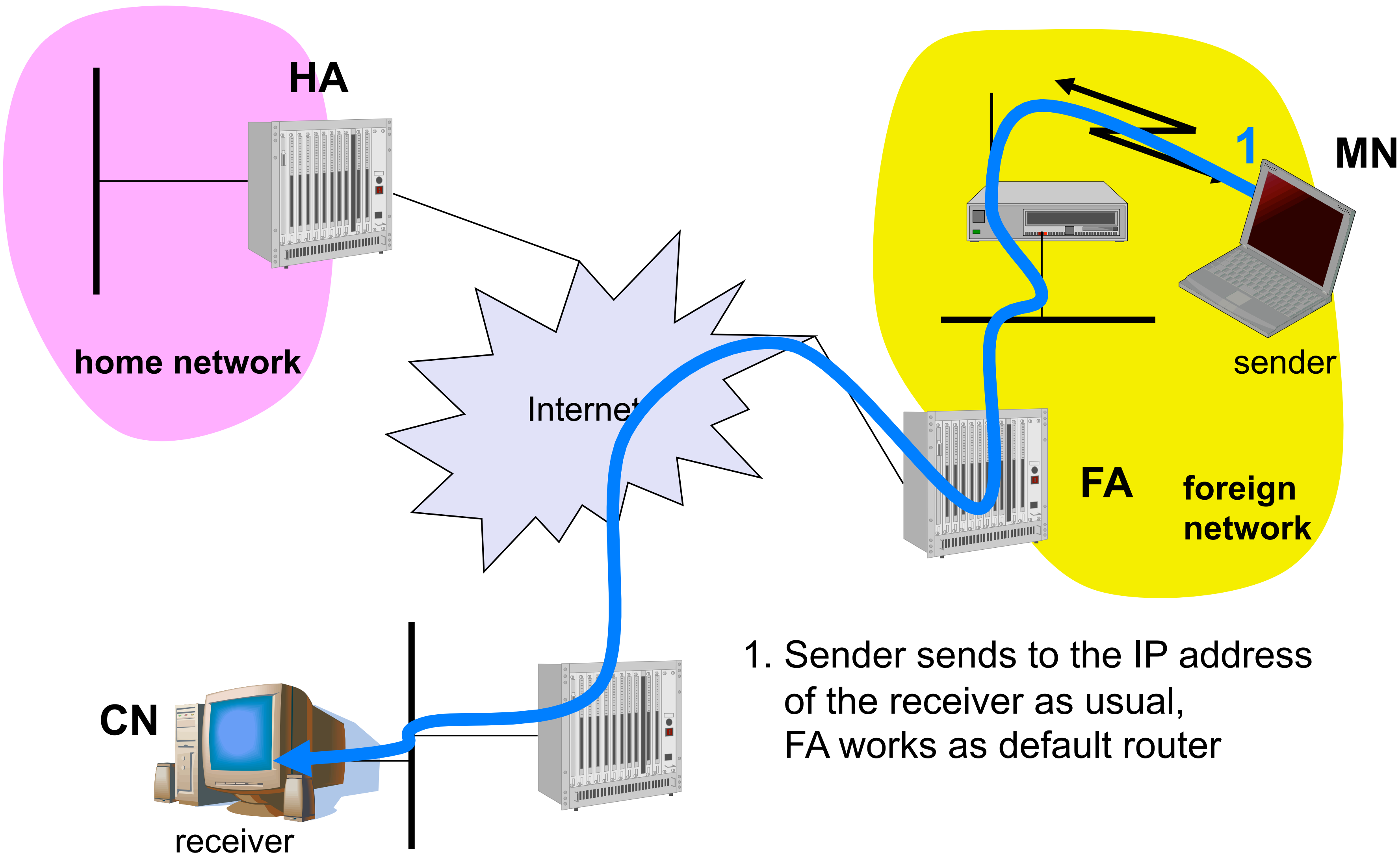


Data transfer to the mobile system



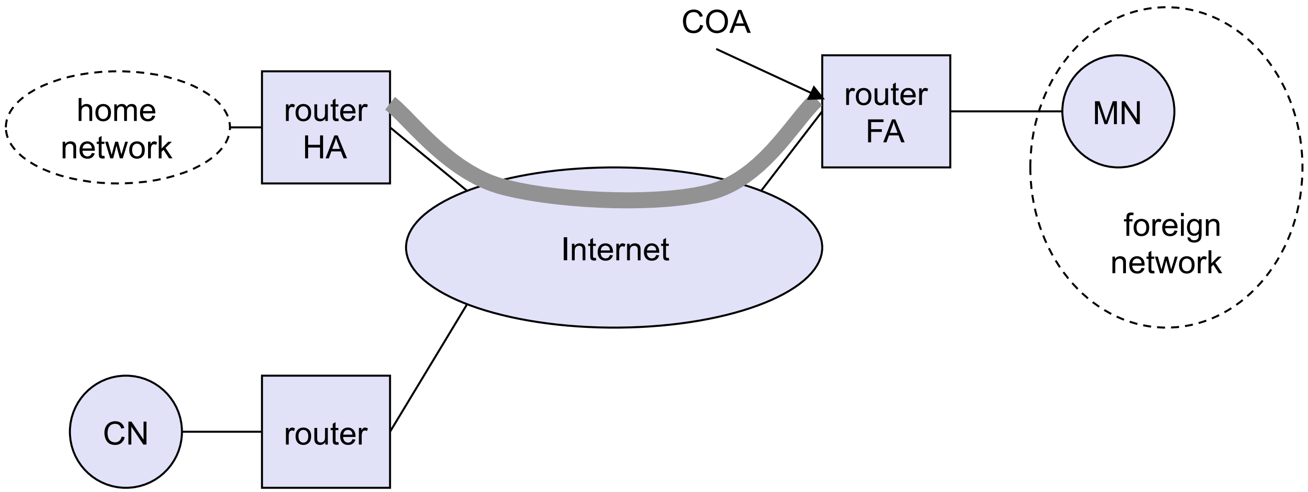
1. Sender sends to the IP address of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to COA, here FA, by encapsulation
3. FA forwards the packet to the MN

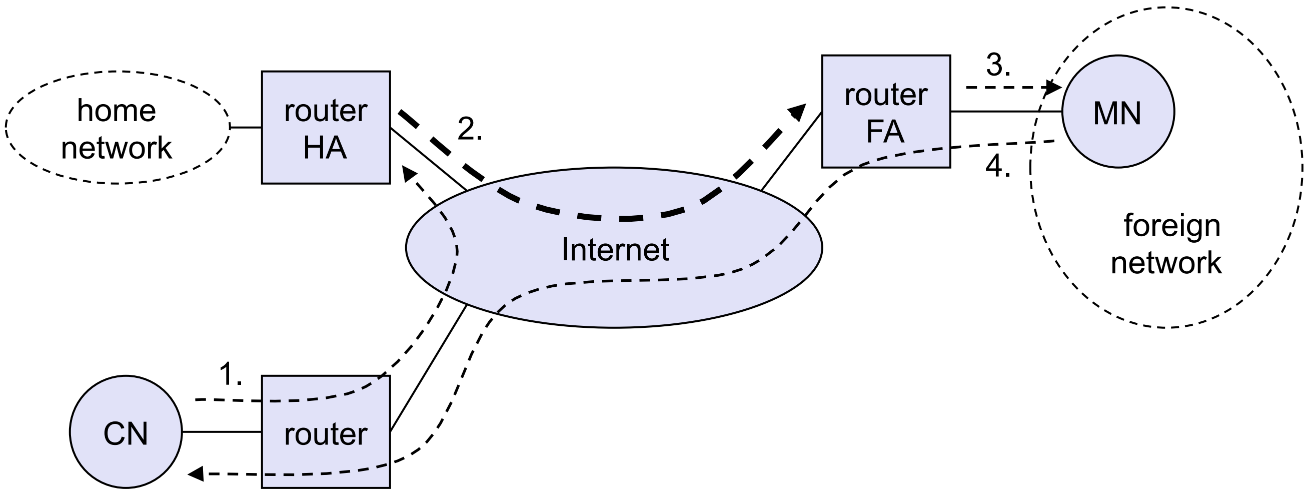
Data transfer from the mobile system



1. Sender sends to the IP address of the receiver as usual, FA works as default router

Overview


COA
home
network
router
HA
router
FA
MN
Internet
foreign
network
CN
router


home
network
router
HA
2.
router
FA
3.
4.
MN
Internet
foreign
network
1.
CN
router

Network integration

- **Agent Advertisement**
  - HA and FA periodically send advertisement messages into their physical subnets
  - MN listens to these messages and detects, if it is in the home or a foreign network (standard case for home network)
  - MN reads a COA from the FA advertisement messages
- **Registration (always limited lifetime!)**
  - MN signals COA to the HA via the FA, HA acknowledges via FA to MN
  - these actions have to be secured by authentication
- **Advertisement**
  - HA advertises the IP address of the MN (as for fixed systems), i.e. standard routing information
  - routers adjust their entries, these are stable for a longer time (HA responsible for a MN over a longer period of time)
  - packets to the MN are sent to the HA,
  - independent of changes in COA/FA

Agent advertisement

0 … 7	8 … 15	16 … 23	24 … 31
type	code	checksum	
#addresses	addr. size	lifetime	
router address 1			
preference level 1			
router address 2			
preference level 2			

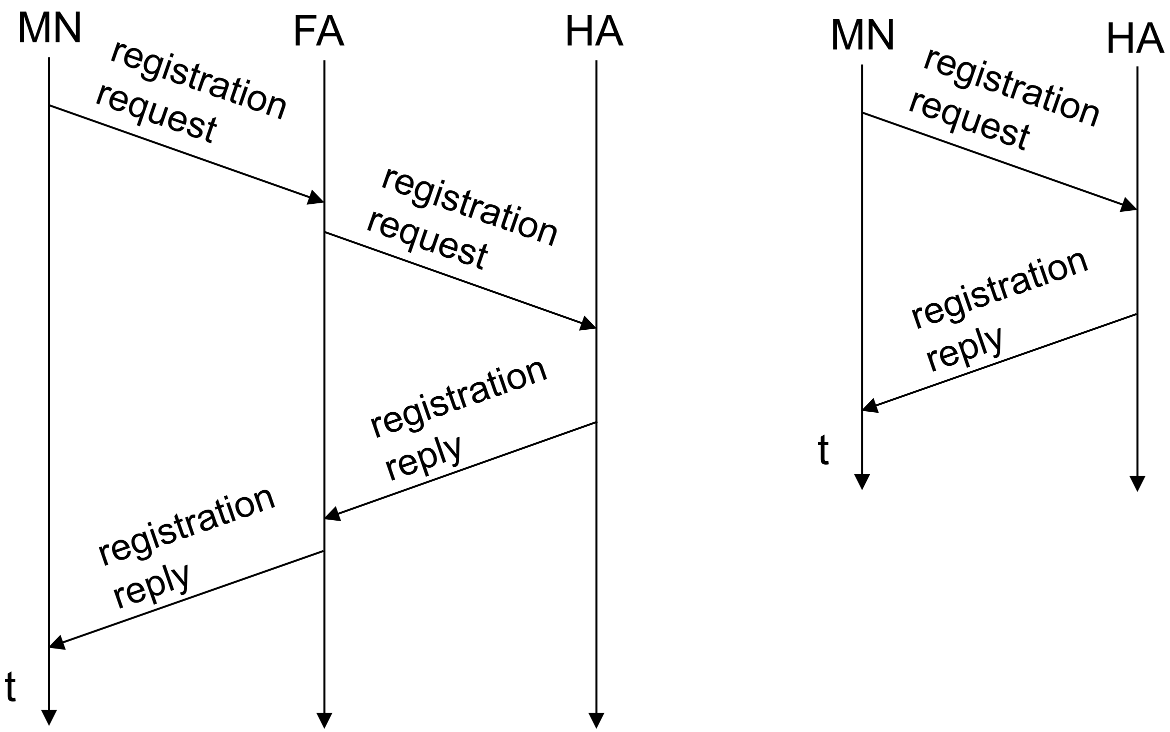
. . .

0 … 7	8 … 15	16 … 23	24 … 31
type = 16	length	sequence number	
registration lifetime		R B H F M G r T	reserved
COA 1			
COA 2			

. . .

type = 16
length = 6 + 4 * #COAs
R: registration required
B: busy, no more registrations
H: home agent
F: foreign agent
M: minimal encapsulation
G: GRE encapsulation
r: =0, ignored (former Van Jacobson compression)
T: FA supports reverse tunneling
reserved: =0, ignored

Registration



Mobile IP registration request

0 7	8 15	16 23	24 31
type = 1	S B D M G r T x	lifetime	
home address			
home agent			
COA			
identification			
extensions . . .			

S: simultaneous bindings
B: broadcast datagrams
D: decapsulation by MN
M mininal encapsulation
G: GRE encapsulation
r: =0, ignored
T: reverse tunneling requested
x: =0, ignored

Mobile IP registration reply

0 — 7	8 — 15	16 — 31
type = 3	code	lifetime
home address		
home agent		
identification		
extensions . . .		

Example codes:

registration successful

- 0 registration accepted
- 1 registration accepted, but simultaneous mobility bindings unsupported

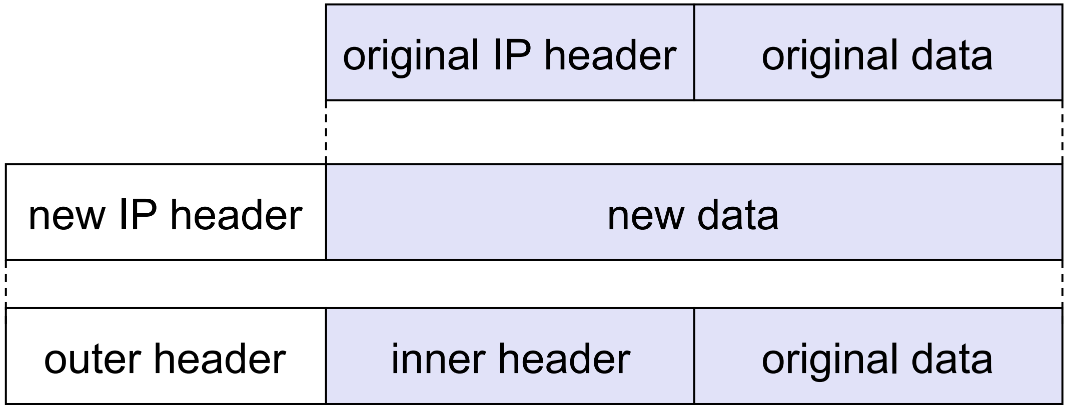
registration denied by FA

- 65 administratively prohibited
- 66 insufficient resources
- 67 mobile node failed authentication
- 68 home agent failed authentication
- 69 requested Lifetime too long

registration denied by HA

- 129 administratively prohibited
- 131 mobile node failed authentication
- 133 registration Identification mismatch
- 135 too many simultaneous mobility bindings

Encapsulation

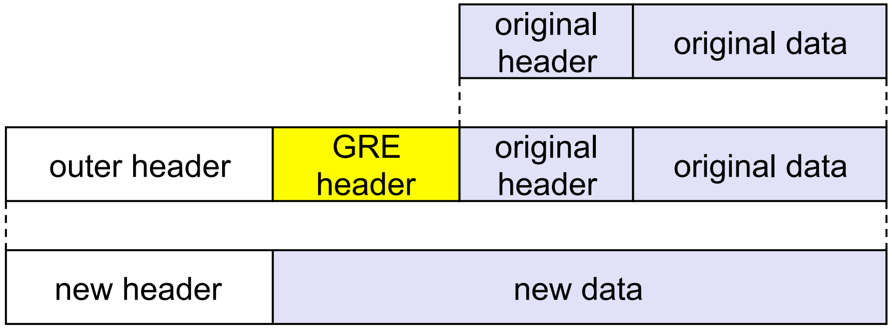


Encapsulation I

- **Encapsulation of one packet into another as payload**
  - **e.g. IPv6 in IPv4 (6Bone), Multicast in Unicast (Mbone)**
  - **here: e.g. IP-in-IP-encapsulation, minimal encapsulation or GRE (Generic Record Encapsulation)**
- **IP-in-IP-encapsulation (mandatory, RFC 2003)**
  - **tunnel between HA and COA**

<table>
<tr><td>ver.</td><td>IHL</td><td>DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="3">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>IP-in-IP</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="5">IP address of HA</td></tr>
<tr><td colspan="5">Care-of address COA</td></tr>
<tr><td>ver.</td><td>IHL</td><td>DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="3">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="2">TTL</td><td>lay. 4 prot.</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="5">IP address of CN</td></tr>
<tr><td colspan="5">IP address of MN</td></tr>
<tr><td colspan="5">TCP/UDP/ ... payload</td></tr>
</table>

Generic Routing Encapsulation



RFC 1701

<table>
<tr><td colspan="4">ver.</td><td colspan="2">IHL</td><td colspan="2">DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="8">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="6">TTL</td><td colspan="2">GRE</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="10">IP address of HA</td></tr>
<tr><td colspan="10">Care-of address COA</td></tr>
<tr><td>C</td><td>R</td><td>K</td><td>S</td><td>s</td><td>rec.</td><td>rsv.</td><td>ver.</td><td colspan="2">protocol</td></tr>
<tr><td colspan="8">checksum (optional)</td><td colspan="2">offset (optional)</td></tr>
<tr><td colspan="10">key (optional)</td></tr>
<tr><td colspan="10">sequence number (optional)</td></tr>
<tr><td colspan="10">routing (optional)</td></tr>
<tr><td colspan="4">ver.</td><td colspan="2">IHL</td><td colspan="2">DS (TOS)</td><td colspan="2">length</td></tr>
<tr><td colspan="8">IP identification</td><td>flags</td><td>fragment offset</td></tr>
<tr><td colspan="6">TTL</td><td colspan="2">lay. 4 prot.</td><td colspan="2">IP checksum</td></tr>
<tr><td colspan="10">IP address of CN</td></tr>
<tr><td colspan="10">IP address of MN</td></tr>
<tr><td colspan="10">TCP/UDP/ ... payload</td></tr>
</table>

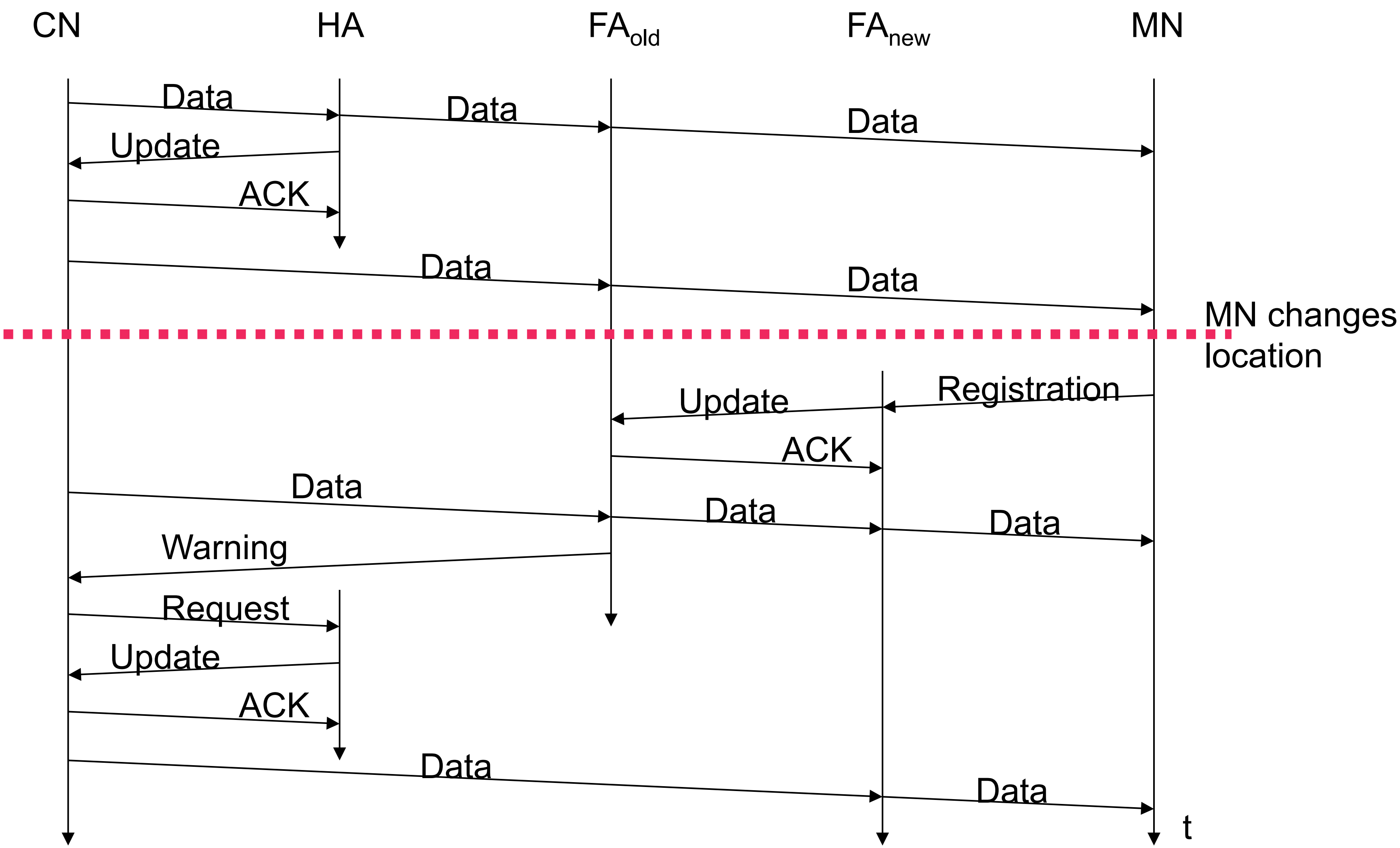
RFC 2784 (updated by 2890)

<table>
<tr><td>C</td><td>reserved0</td><td>ver.</td><td>protocol</td></tr>
<tr><td colspan="3">checksum (optional)</td><td>reserved1 (=0)</td></tr>
</table>

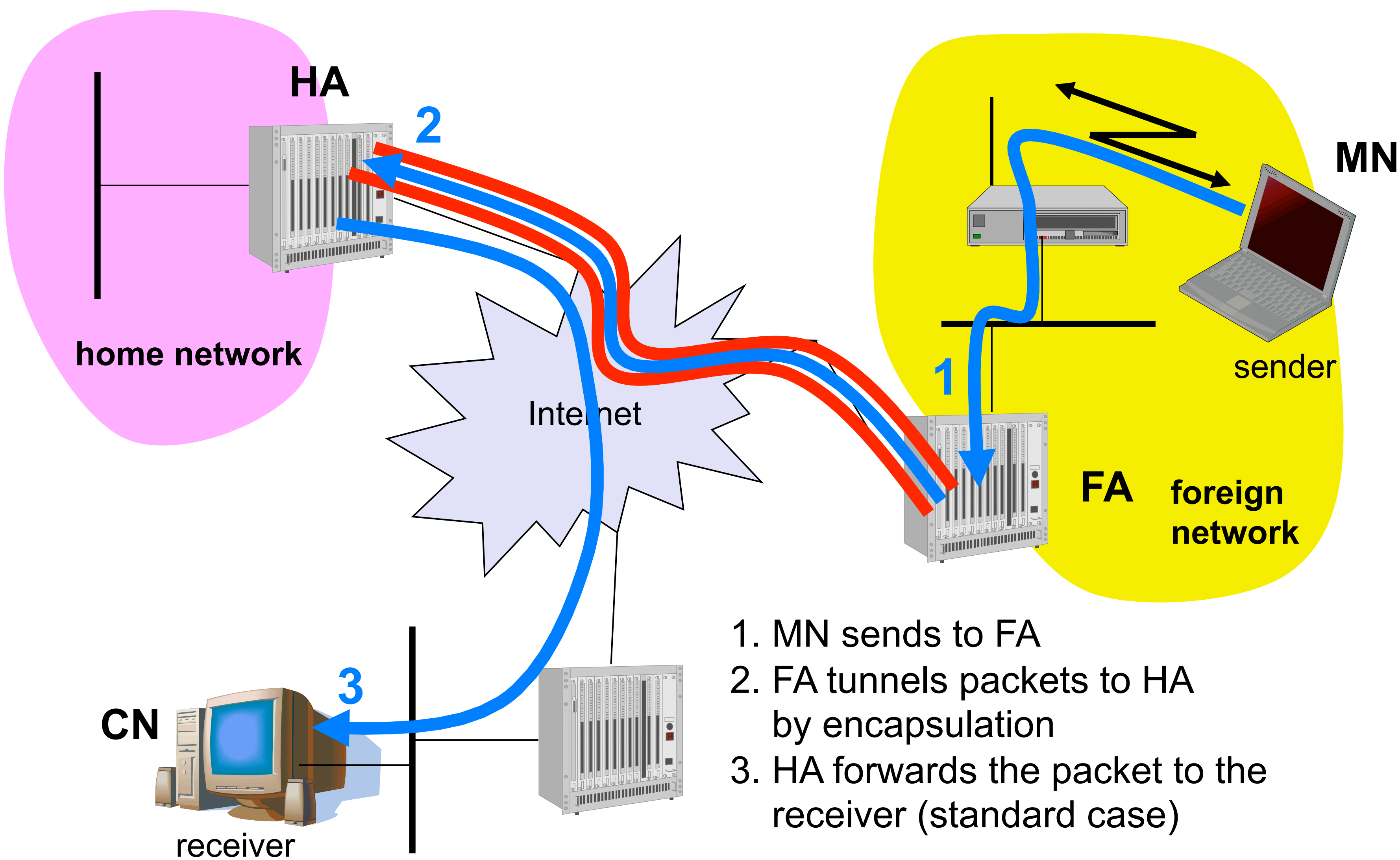
Optimization of packet forwarding

- Problem: Triangular Routing
  - sender sends all packets via HA to MN
  - higher latency and network load
- “Solutions”
  - sender learns the current location of MN
  - direct tunneling to this location
  - HA informs a sender about the location of MN
  - big security problems!
- Change of FA
  - packets on-the-fly during the change can be lost
  - new FA informs old FA to avoid packet loss, old FA now forwards remaining packets to new FA
  - this information also enables the old FA to release resources for the MN

Change of foreign agent



Reverse tunneling (RFC 3024, 2344)



1. MN sends to FA
2. FA tunnels packets to HA by encapsulation
3. HA forwards the packet to the receiver (standard case)

Mobile IP with reverse tunneling

- **Router accept often only “topological correct“ addresses (firewall!)**
  - **a packet from the MN encapsulated by the FA is now topological correct**
  - **furthermore multicast and TTL problems solved (TTL in the home network correct, but MN is to far away from the receiver)**
- **Reverse tunneling does not solve**
  - **problems with *firewalls*, the reverse tunnel can be abused to circumvent security mechanisms (tunnel hijacking)**
  - **optimization of data paths, i.e. packets will be forwarded through the tunnel via the HA to a sender (double triangular routing)**
- **The standard is backwards compatible**
  - **the extensions can be implemented easily and cooperate with current implementations without these extensions**
  - **Agent Advertisements can carry requests for reverse tunneling**

Problems with mobile IP

- Security
  - authentication with FA problematic, for the FA typically belongs to another organization
  - no protocol for key management and key distribution has been standardized in the Internet
  - patent and export restrictions
- Firewalls
  - typically mobile IP cannot be used together with firewalls, special set-ups are needed (such as reverse tunneling)
- Security, firewalls, QoS etc. are topics of research and discussions
- requires changes of MN
- NAT

Mobile IP usage

- Not in original form
- PMIPv6 = Proxy MIP
  - Proxy: client doesn’t run MIP, but a proxy
  - client@SGSN, FA/CoA@GGSN, HA@somewhere in CN
  - 3G (UMTS) and 4G(LTE, WiMAX) networks
  - Maintain mobility in core network
  - Support in many CISCO boxes: ASR, ISR, WLC
  - Mobile offloading
  - Large WiFi deployments

Mobile IP summary

- IP = the narrow waist of the Internet
- hard to upgrade
- basic mobility solution
  - tunneling IP in IP
  - triangle routing
  - Double triangle routing
- Deployment problems
  - compatibility, security, NAT
- MIP not used in original form
  - Setups where HA, FA, clients are under control

Zeroconf

1) Address assignment
   - What IP address do I have?

2) DNS without a server
   - What is my name?
   - mDNS (Multicast DNS)

3) Service location discovery
   - What network services are available?

Zeroconf

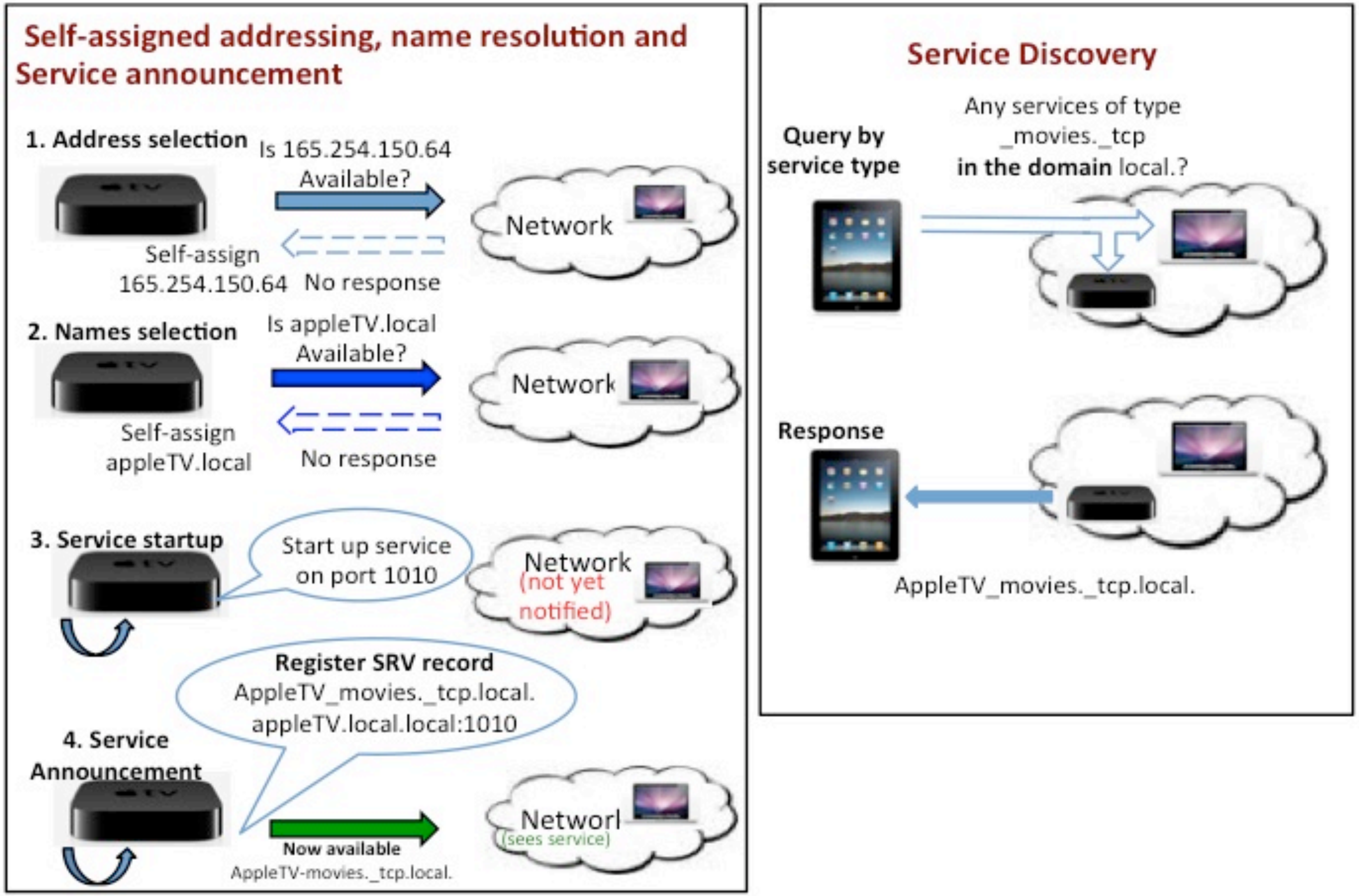
requirement	Linux, BSD *Avahi*	OSX *Bonjour*	Windows
Automatic IP allocation	Link-local	Link-local	Link-local
Name resolution	mDNS	mDNS	LLMNR
Service discovery	DNS-SD	DNS-SD	SSDP(UPnP)

Zeroconf hardware, software

- Apple
  - AppleTV , AirPort, (AirPlay protocol), AirDrop, iTunes, etc
- Google
  - Chromecast
- Various vendors
  - Printers, NAS, network video players, projectors, TVs

Example

Self-assigned addressing, Name resolution and service publication



Obtain an IP address

- Manual assignment
  - Netmask
  - Router
  - Broadcast domain
  - Conflict resolution

- DHCP
  - Conflict resolution

- Link-local (self assigned)

Link-local Address Assignment

- IPv6
  - Link-Local FE80::/16
  - Duplication Address Discovery (DAD)

- IPv4
  - 169.254.0.0/16
  - first and last 254 addresses are reserved
  - Random based address selection; seed=MAC address
  - ARP-based duplicate discovery
  - Conflict probability for 1300 hosts
    - 98% to succeed in first try
    - 99.96% to succeed in two tries

Claim a local address

Time	Source	Destination	Protocol	Info
3.703964	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
3.983703	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.004198	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
4.283867	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.304479	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 0.0.0.0
4.584088	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.884300	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 0.0.0.0
4.905167	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 169.254.187.245
5.184522	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.186.86
5.205780	dimsumthinking.local	Broadcast	ARP	Who has 169.254.187.245? Tell 169.254.187.245
5.485642	foo.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.186.86
26.260885	dimsumthinking.local	Broadcast	ARP	Who has 169.254.186.86? Tell 169.254.187.245
26.260929	foo.local	Broadcast	ARP	169.254.186.86 is at 00:03:93:ef:c4:8c

time 3.7-4.8: each machine tries and address

time 4.9-5.5: machines claim IP addresses

time 26.2: actual ARP query, response

Link-Local Issues

Maintenance

– **Defending your address**

– **Late conflicts: when someone claims your IP, send a single ARP in defense**

Multiple interfaces

– **broadcast on all local interfaces**

Address selection

– **try to prefer routable addresses**

– **stop using local address when a global one is available**

– **local addresses are not globally reachable**

Zeroconf

1) Address assignment

- What IP address do I have?

2) DNS without a server

- **What is my name?**
- **mDNS (Multicast DNS)**

3) Service location discovery

- What network services are available?

We have an address -now what?

Communication via link-local is a pain
- Need to look up raw addresses
- Need to type addresses in directly

DNS would be nice, but
- there is no DNS server available, or
- if there is a server I don't know where it is

Multicast

IP Multicast addresses

– `1110xxxx xbbbbbbb bbbbbbbb bbbbbbbb`

Ethernet Multicast addresses

– `00000001 00000000 01011110 0bbbbbbb bbbbbbbb bbbbbbbb`

Hosts “join” multicast groups

– **have ethernet card listen to multicast addresses**

– **respond to IP multicast**

– **tell routers that you want to participate**

Multicast cont.

On a local link multicast is very efficient

– convert the layer 2 multicast
– does not disrupt non-participants

On the global internet

– Multicast should be efficient for one-to-many delivery
– Routers have to keep track of participants = complexity and “state” in the router
– Efficiency looses to simplicity

Local Name Discovery

- Has long been used on Mac OS, Windows, and Novell
  - –NETBIOS Names
  - –AppleTalk
- Broadcast-based name announcements
- “Chatty” Protocols

Multicast DNS (mDNS)

Issue an (almost) standard DNS query

Target is not a DNS server but a multicast address

- 224.0.0.251 for IPv4
- FF02::FB for IPv6
- to/from port 5353 (standard DNS is 53)
- DNS packet structure maintained
- DNS packet semantics slightly change

mDNS Implementation

Client wants to resolve a name

- –Multicast the query

One or more members of the multicast group reply

- –One reply for unique information (name to address)
- –Many replies for shared information (services)

Replies are multicast to allow all clients to use the answer

mDNS queries

1) One-shot with single answers
   - Example: http://mylaptop.local triggers 224.0.0.251:5353
2) One-shot with multiple answers
   - wait for multiple answers
   - on retransmission include answers so far
3) Ongoing
   - Repeat w exponential backoff
   - New clients send gratuituous responses
   - Known answers in query
   - Responses are multicast

mDNS Implementation

- Windows/OSX/IOS/Linux
- Names in the ".local." domain are resolved by mDNS
- No hierarchy is implied or allowed
- There are no NS or SOA "records"
- Replies must have TTL= 255
  - Protect against attackers injecting malicious answers from outside the network

Name/Content Assignment

DNS query type A

mDNS Query type T_ANY

- returns <u>all</u> matching records
- If no conflict, repeat after 250ms
- After 750ms (3 queries), name is unique

Name/Content Assignment

Unique information, e.g. host names

- Host creates a name it wants to use
- Issues a query to see if there is a conflict
- Host who got the name first “wins”
- In a race condition (two hosts start using the same name at the same time) the one with the lower address “wins”

Shared information

- Host responds to queries as appropriate

Zeroconf

1) Address assignment
   - What IP address do I have?

2) DNS without a server
   - What is my name?
   - mDNS (Multicast DNS)

3) **Service location discovery**
   - **What network services are available?**

DNS-Based Service Discovery

- <u>Browse for services, not devices</u>
- DNS SRV records [RFC 2782]:

"_http._tcp.local." lists all address/port combinations for http servers reachable by TCP in the local. domain

- DNS-SD adds one level of indirection to allow a named list of services that can be presented to the user

DNS-Based Service Discovery

Service discovery requires a central aggregation server

- DNS already has one: it's called a DNS server.

Service discovery requires a service registration protocol

- DNS already has one: it's called DNS Dynamic Update.

Service discovery requires a query protocol

- DNS already has one: it's called DNS.

Service discovery requires security mechanisms

- DNS already has security mechanisms: they're called DNSSEC.

Service discovery requires a multicast mode for ad-hoc networks

- Zeroconf environments already require a multicast-based, DNS-like name lookup protocol for mapping hostnames to addresses, so it makes sense to let one multicast-based protocol do both jobs.

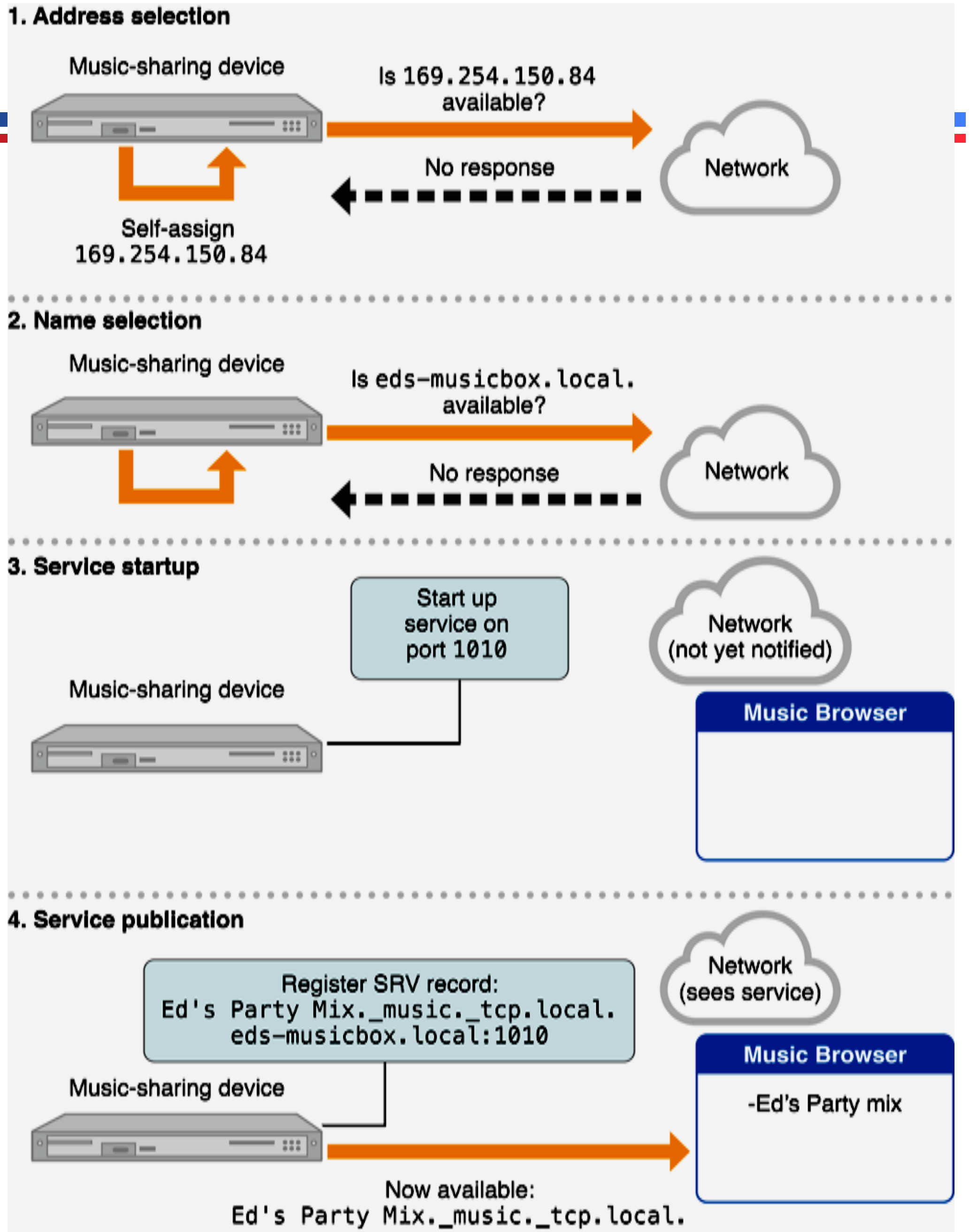
DNS-SD

- Query for PTR records (instead of SRV)
  - query for PTR with name "_ipp._tcp.local."
  - get a list of <instance>.<service>.<domain> records
    - "ColorPrinter. _ipp._tcp.local." "SlowPrinter. _ipp._tcp.local."
- Give the user a list of options
  - key=value in TXT record
- Issue SRV (and TXT) query for the desired instance
- Late binding

DNS-SD example

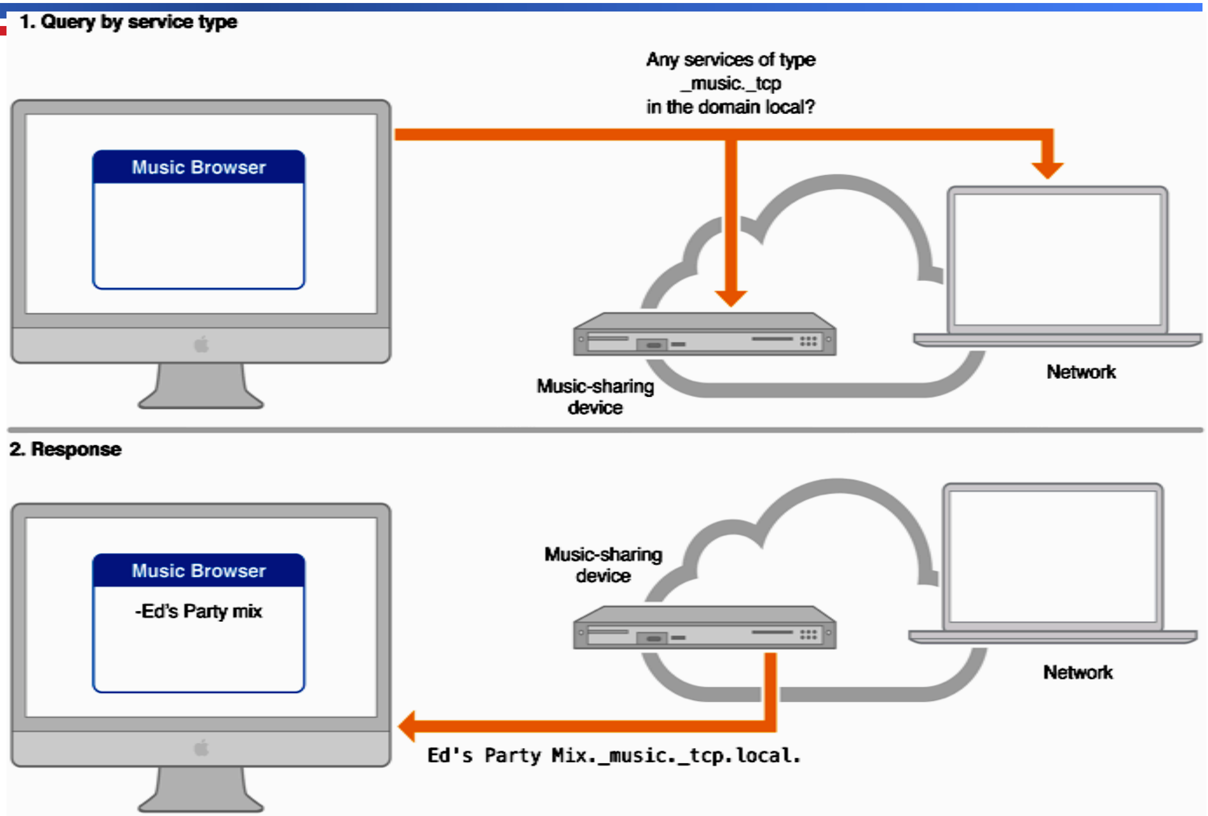
PUBLICATION

1. Client randomly selects 169.254.150.84/16
   - Advertises, claims address
2. mDNS responder claims name eds-musicbox.local.
3. device selects free port, start service
4. Publishes service _music._tcp, under the name "Ed's Party Mix", creates
   - SRV record named Ed's Party Mix._music._tcp.local. that points to eds-musicbox.local. on TCP port 1010
   - PTR record named _music._tcp.local. that points to the Ed's Party Mix._music._tcp.local. service.



https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Articles/NetServicesArchitecture.html

DNS-SD example



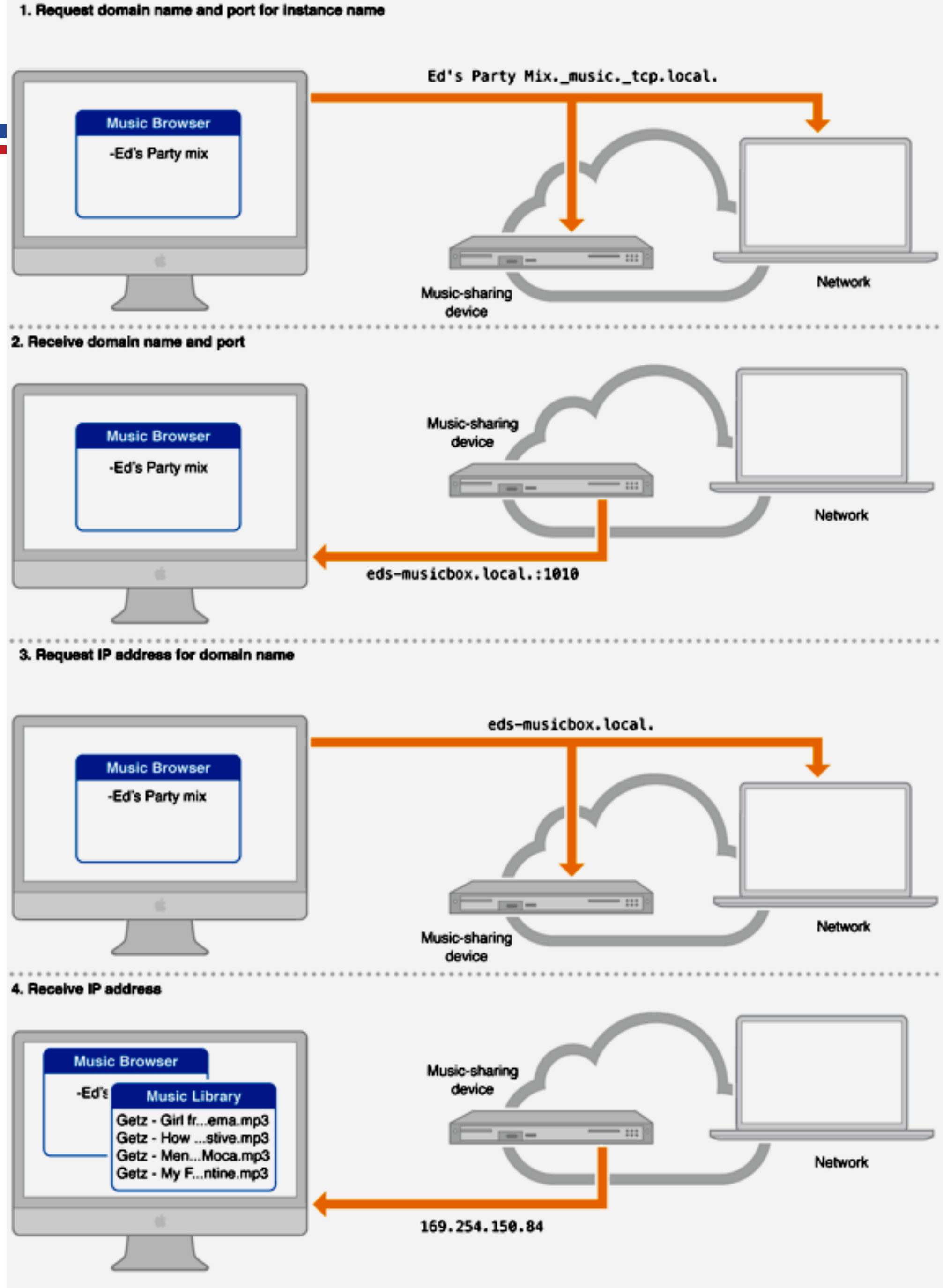
DISCOVERY

1. **App queries for PTR record _music._tcp.local.**
   - **IP Multicast to 224.0.0.251:5353**
2. **mDNS responders on devices answer with service instance names**
   - **Ed's Party Mix._music._tcp.local.**

- **App prompts the user with the instance list**

https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Articles/NetServicesArchitecture.html

DNS-SD example



RESOLUTION

1 App DNS lookup for a SRV record with the name of the service for Ed's Party Mix._music._tcp.local.

2 Receive instance location eds-musicbox.local., 1010

3 Resolve name by multicast: 169.254.150.84

4 Connect to 169.254.150.84:1010, use service

https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/NetServices/Articles/NetServicesArchitecture.html

Other topics

- Apple
  - Bonjour Sleep Proxy: mDNS + magic packet
    - (file share, printer share, ssh)
  - Bonjour gateways, VLAN separation
  - Problems in enterprise networks http://www.networkworld.com/article/2161302/lan-wan/apple-seeks-standard-to-appease-angry-university-net-managers.html

- Microsoft
  - LLMNR (Link-Local Multicast Name Resolution)
  - UPnP
    - Simple Service Discovery Protocol (SSDP)
    - Windows Internet Naming Service (WINS)