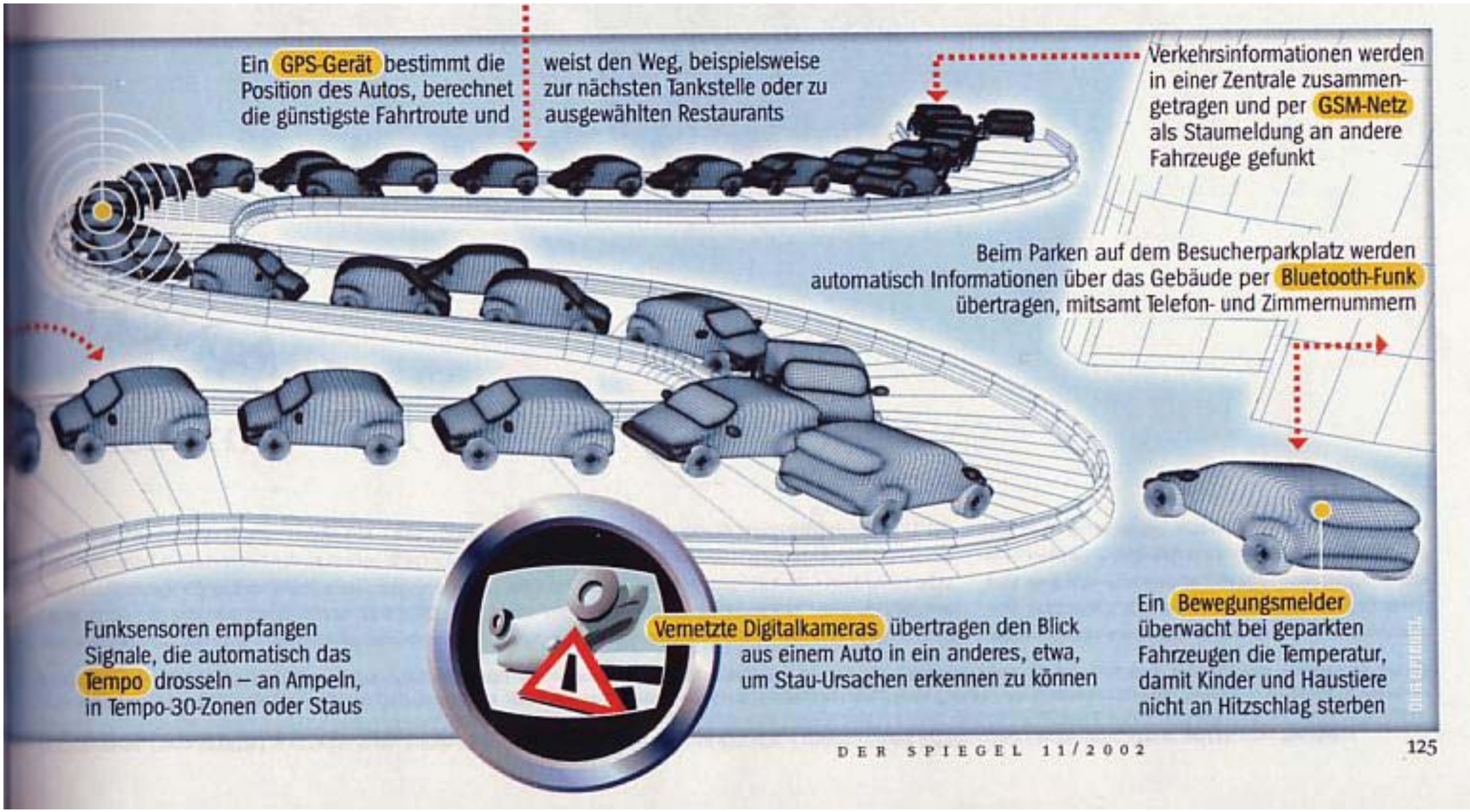
Mobility

Chapter 13

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

More Car Network Ideas



- CAR2CAR Consortium: Audi, BMW, Daimler, Fiat, GM, Honda, Renault, VW

Rating

- Area maturity

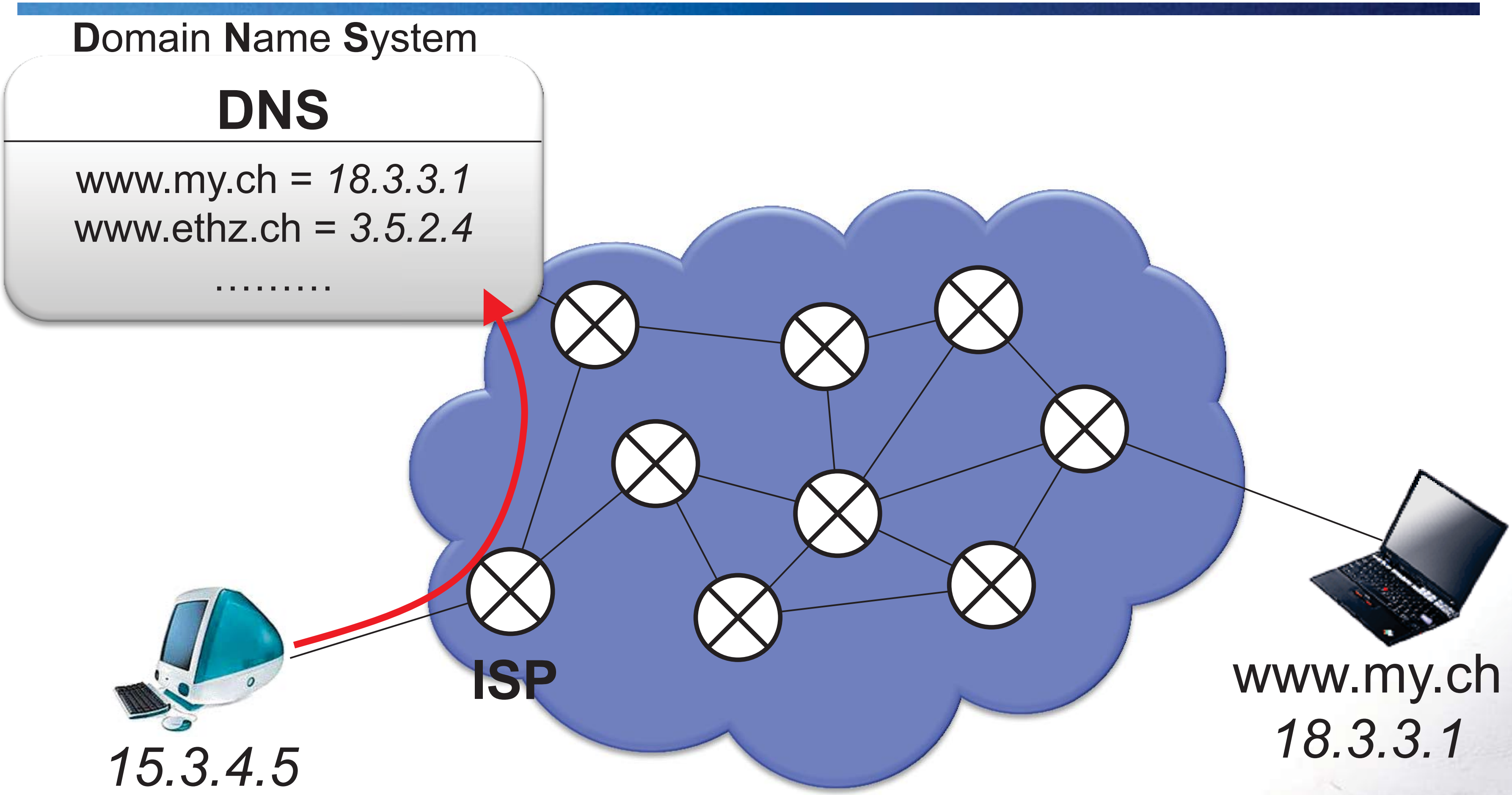
- Practical importance

- Theoretical importance

Overview

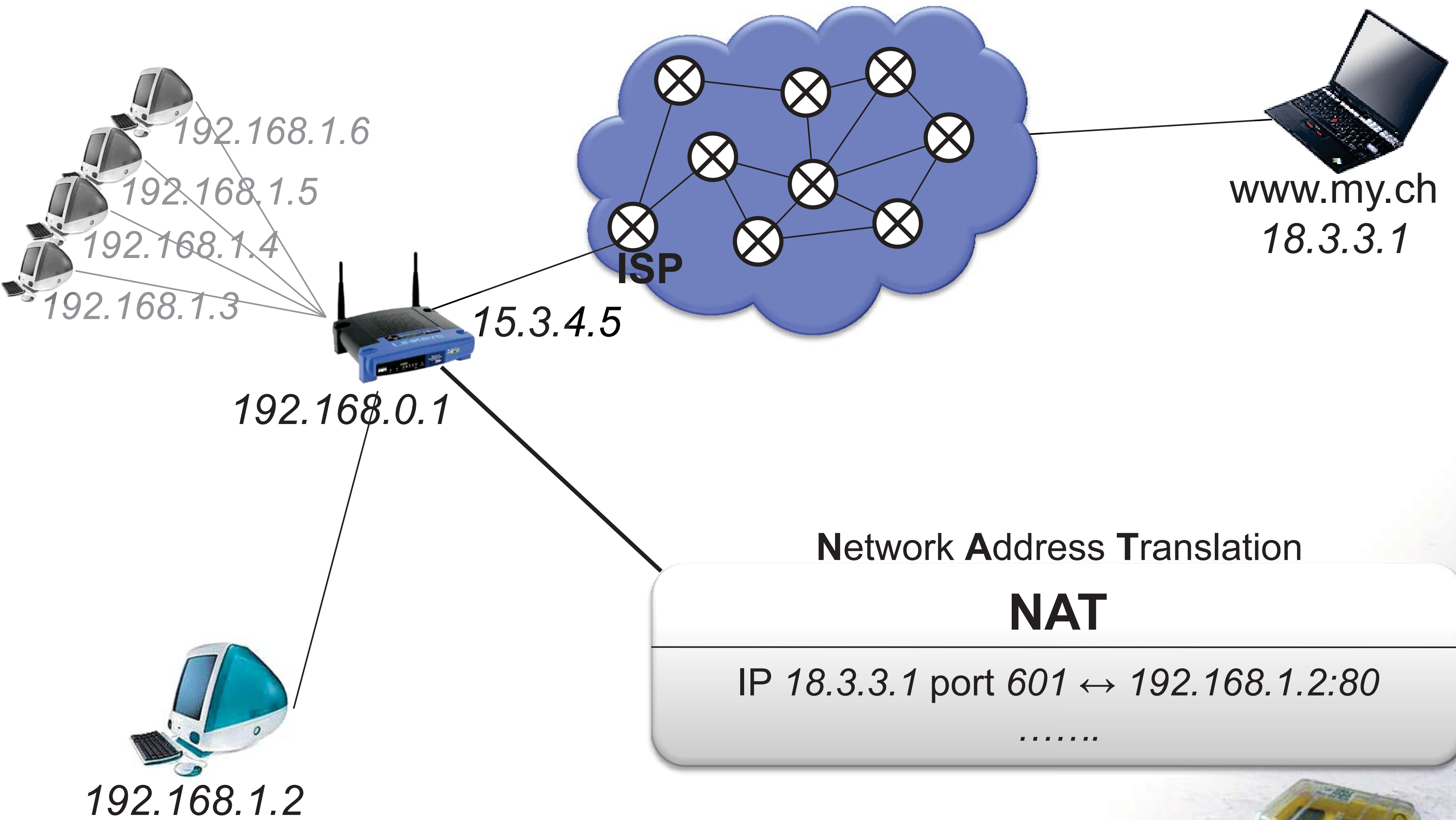
- Mobile IP
- GSM Networks
- Location Services
- Mobility Models

Basics of static routing



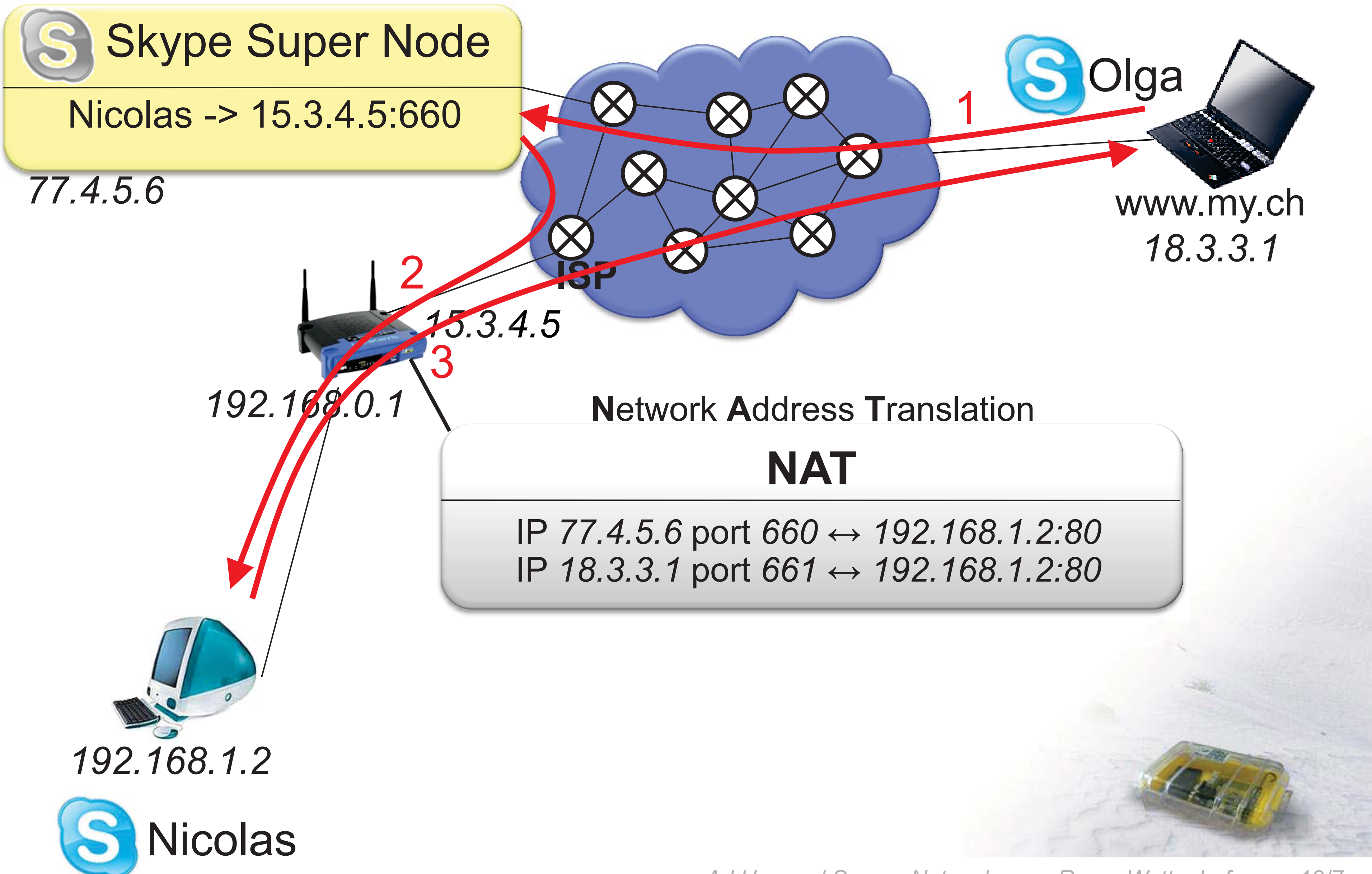
DHCP: gets IP, DNS servers & gateway from ISP
ISP: Internet service provider

NAT

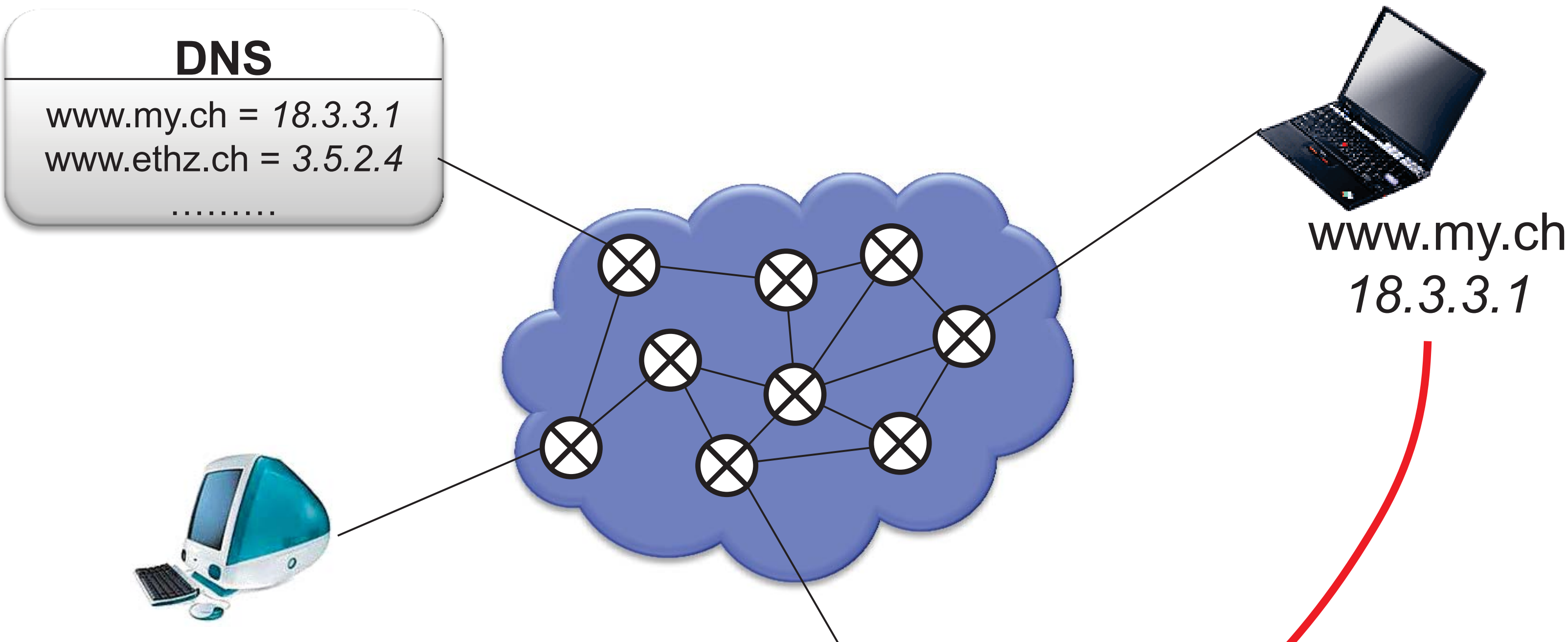


192.168.1.6
192.168.1.5
192.168.1.4
192.168.1.3
192.168.0.1
15.3.4.5
ISP
www.my.ch
18.3.3.1
192.168.1.2
Network Address Translation
NAT
IP 18.3.3.1 port 601 ↔ 192.168.1.2:80
…….

Reachability behind a NAT



Mobile IP



- Update routers & keep IP?
- New IP & update DNS?

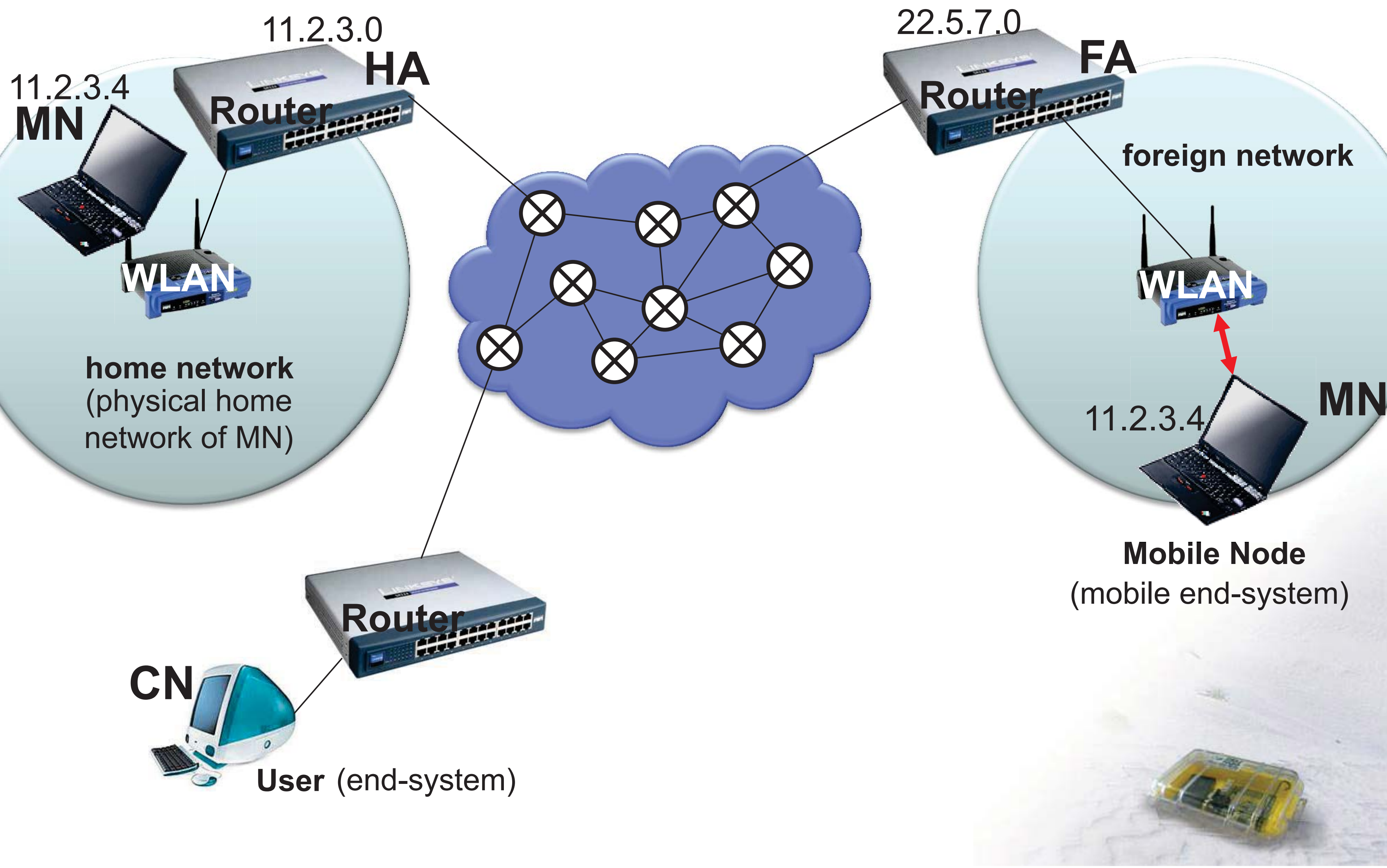
Motivation for mobile IP

- **Routing**
  - based on IP destination address, network prefix (e.g. 129.132.13) determines physical subnet
  - change of physical subnet implies change of IP address to have a topological correct address (standard IP) or needs special entries in the routing tables
- **Changing the IP-address?**
  - adjust the host IP address depending on the current location
  - almost impossible to find a mobile system, DNS updates are too slow
  - TCP connections break
  - security problems
- **Change/Add routing table entries for mobile hosts?**
  - worldwide!
  - does not scale with the number of mobile hosts and frequent changes in their location

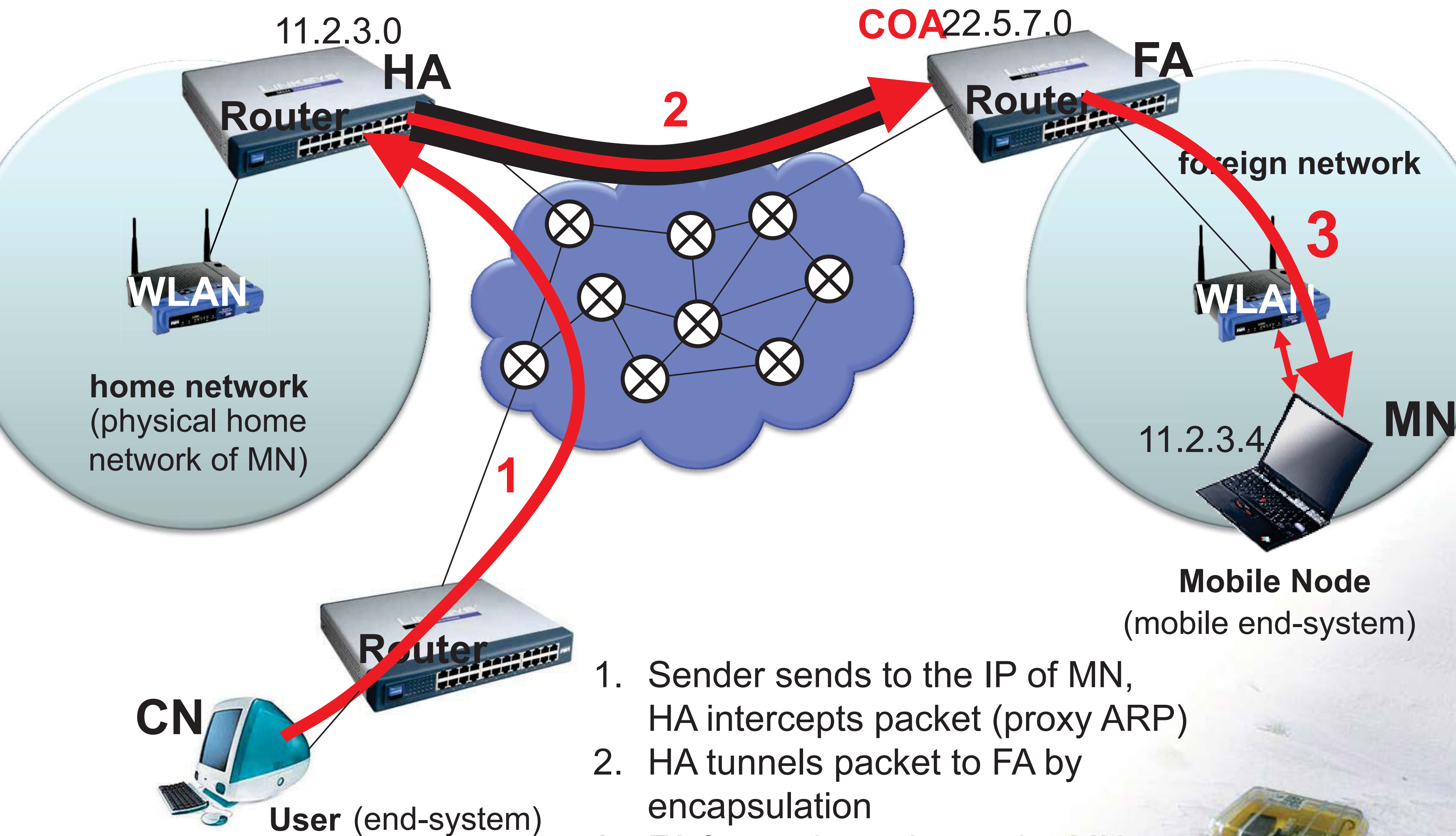
Requirements on mobile IP (RFC 4721)

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

Contacting a mobile node

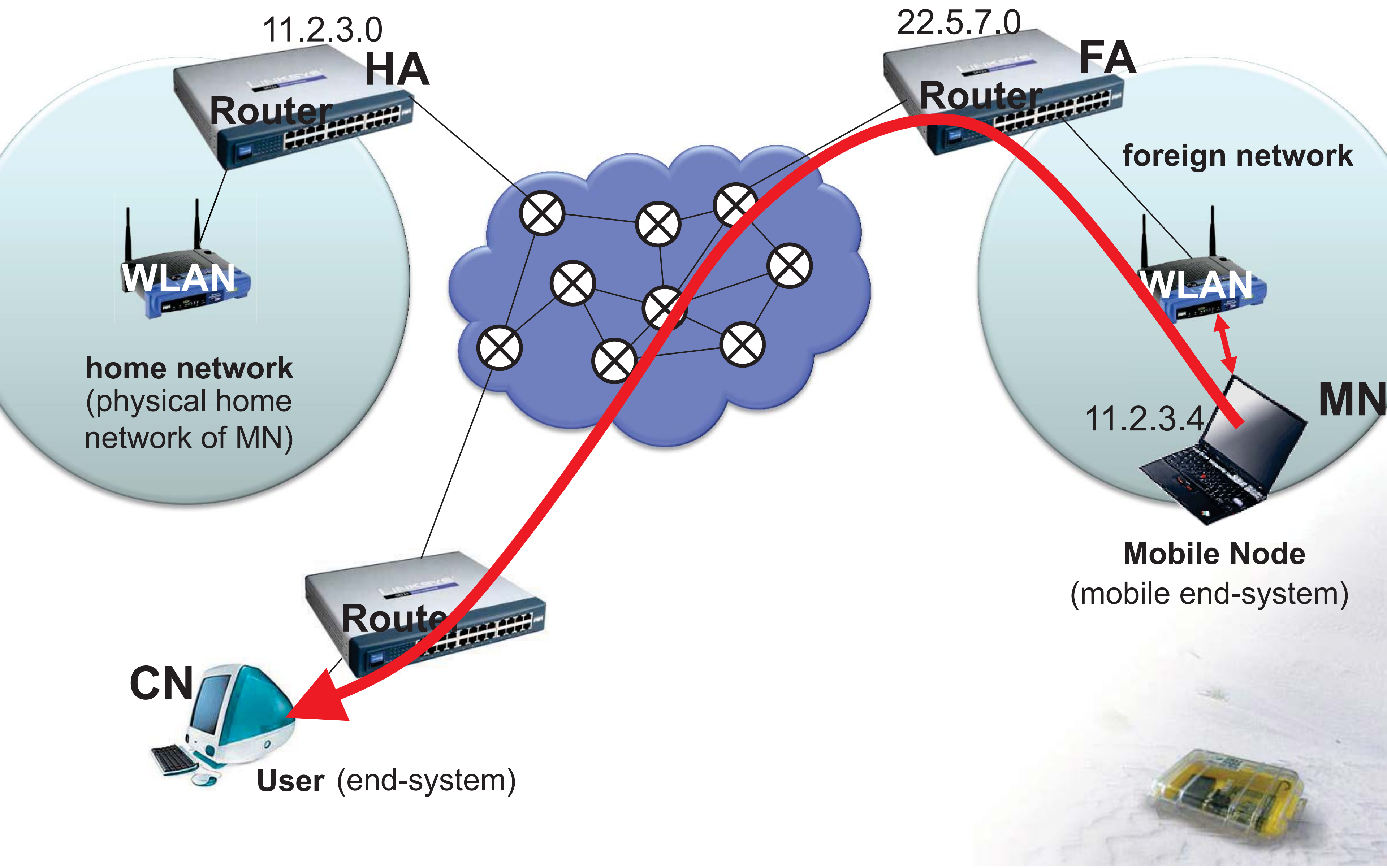


Contacting a mobile node



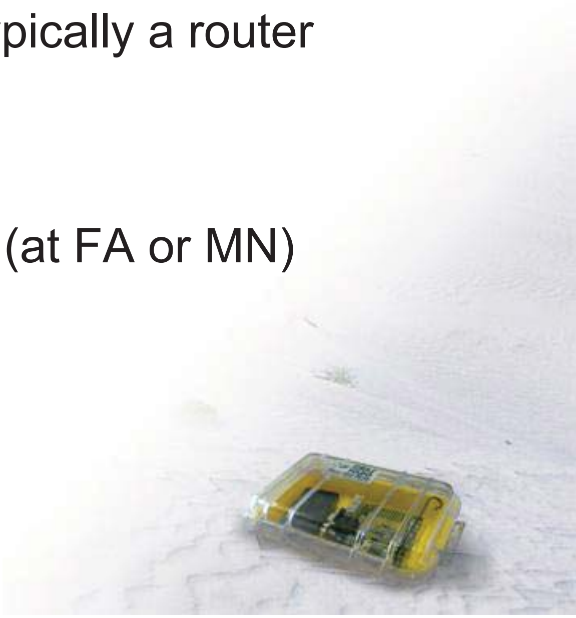
1. Sender sends to the IP of MN, HA intercepts packet (proxy ARP)
2. HA tunnels packet to FA by encapsulation
3. FA forwards packet to the MN

Direct answer

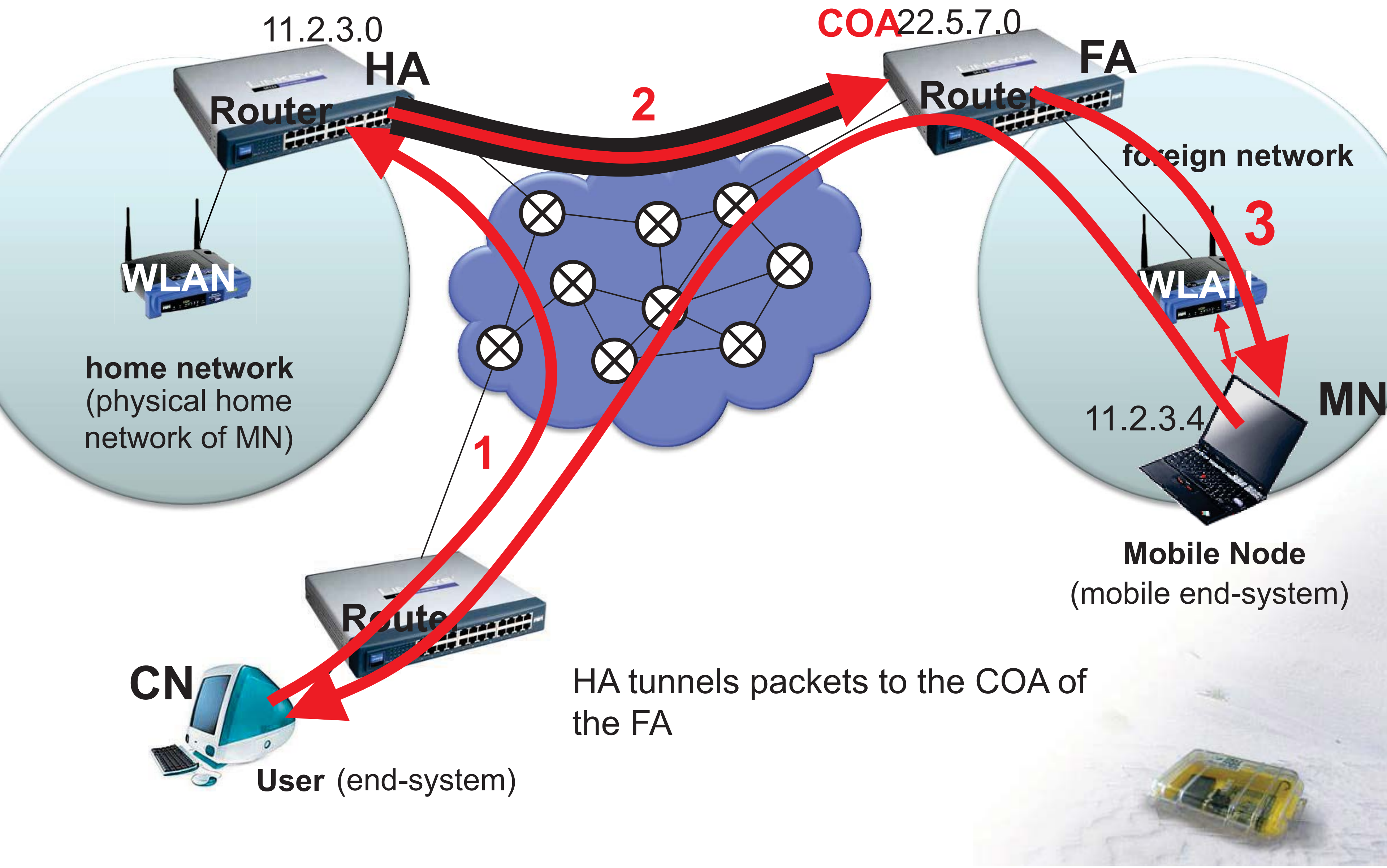


Terminology

- **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
  - typically 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)**



Overview

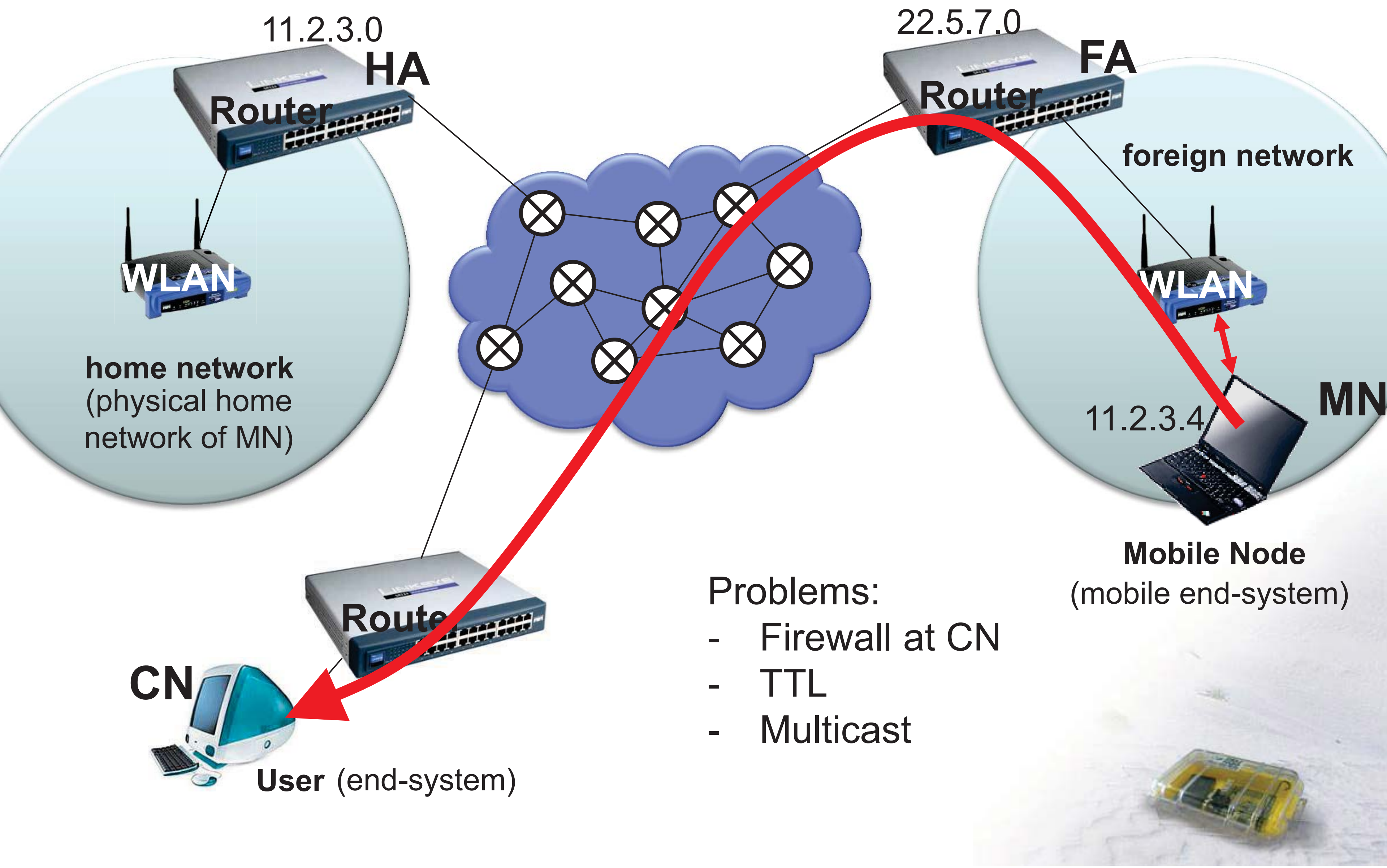


HA tunnels packets to the COA of the FA

How it works…

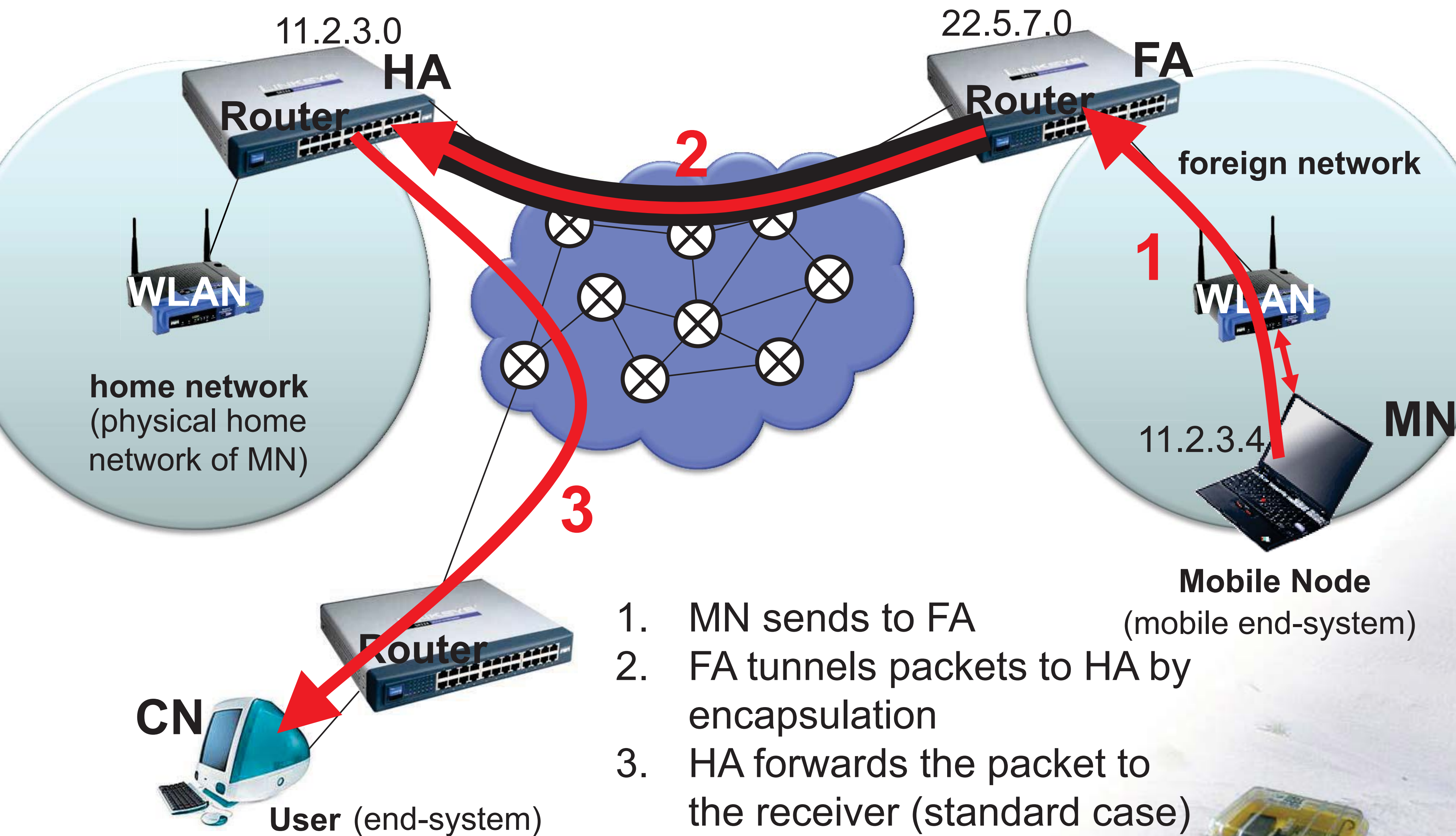
- **Agent Advertisement**
  - HA and FA periodically send advertisement messages into their physical subnets
  - MN listens to these messages and detects if it is in the home or a foreign network (standard case for home network)
  - MN reads a COA from the FA advertisement messages
- **Registration (always 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

Direct answer may not work



Problems:
- Firewall at CN
- TTL
- Multicast

Reverse tunneling (RFC 2344)

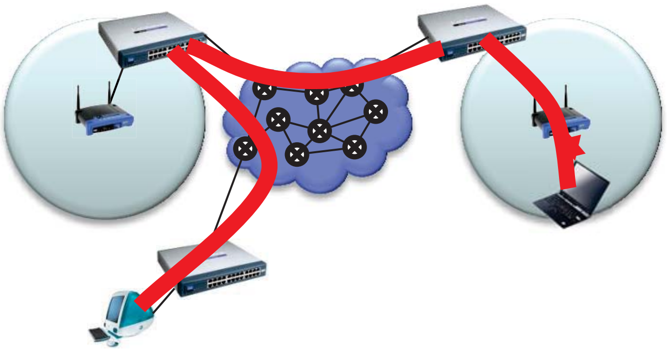


1. MN sends to FA
2. FA tunnels packets to HA by encapsulation
3. HA forwards the packet to the receiver (standard case)

Mobile IP with reverse tunneling

- Router accept often only “topologically correct“ addresses (firewall!)
  - a packet from the MN encapsulated by the FA is now topologically correct
  - furthermore multicast and TTL problems solved (TTL in the home network correct, but MN is too 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)
- Reverse tunneling is backwards compatible
  - the extensions can be implemented easily and cooperate with current implementations without these extensions

Optimization of packet forwarding



- **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

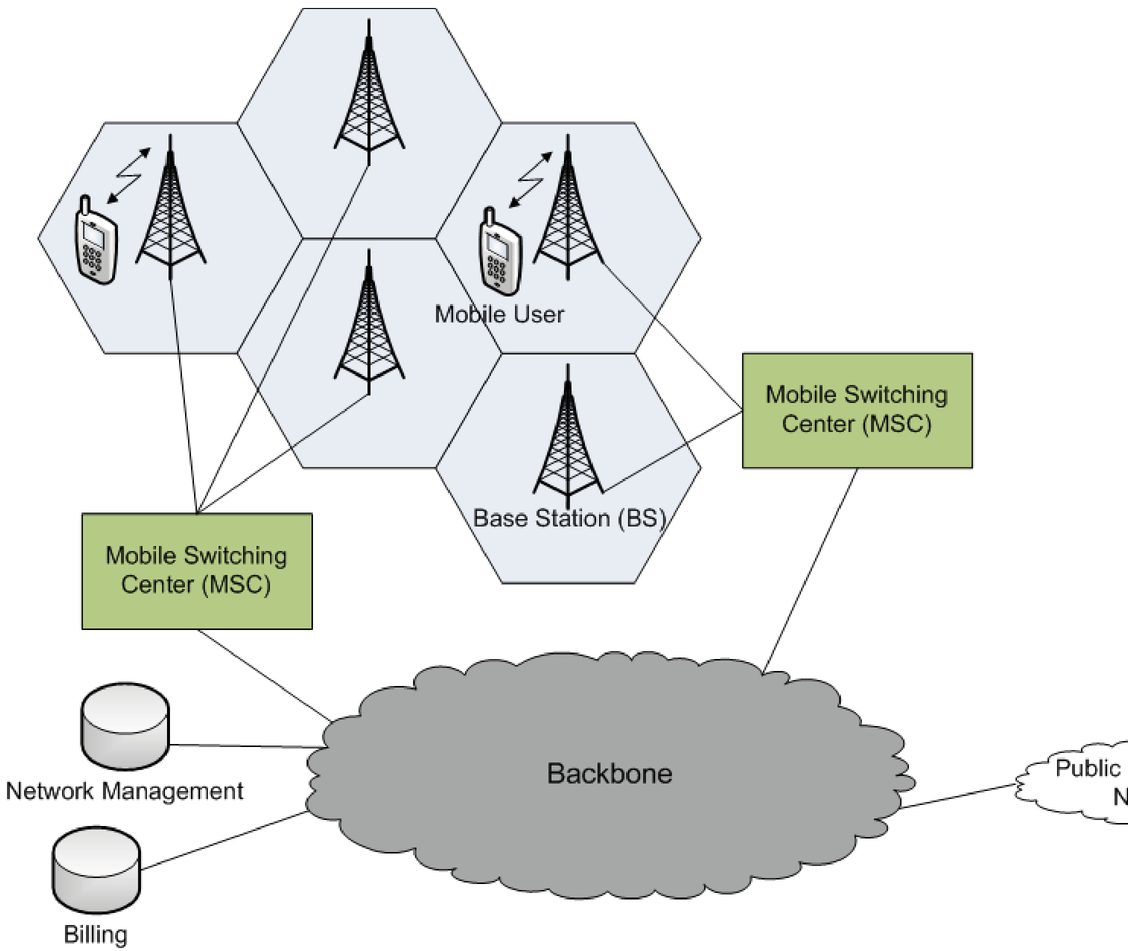
Cellular Networks

- GSM (Global System for Mobile Communications)
  - Standard for a Public Land Mobile Network (PLMN)
  - Specification of the GSM standard in 1991
  - More than 3 billion subscribers worldwide
  - Over 850 GSM network operators

- GSM in Switzerland
  - 8 million mobile phone users (2007)
  - 5 GSM network operators active:

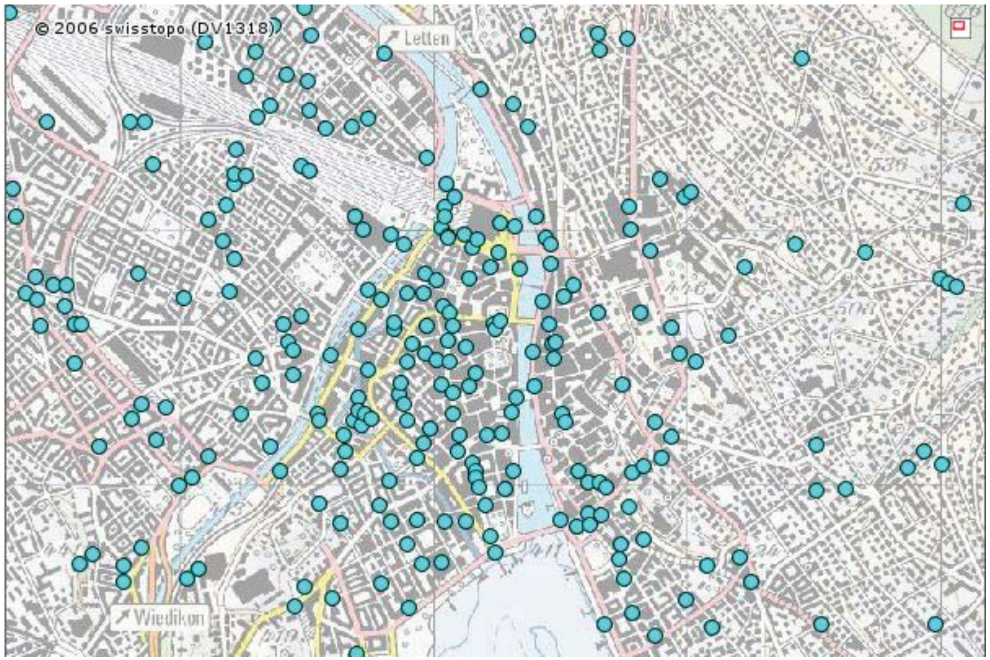
GSM Architecture

- GSM is a combination of wireless and fixed network systems
  - Base station covers mobile users in a cell
  - Different base stations are connected through a backbone network



GSM Coverage

- Coverage in Switzerland is nearly 99.8% of the populated area
  - Approx. 11‘000 base stations (antennas) placed all over the country
- Cell sizes can vary from a few hundred meters up to 30 km
  - Depends on the number of users, geography, transceiver power
- Example: Zurich City



Source: funksender.ch, BAKOM

Handling Mobility in Cellular Networks

- GSM designed for high mobility of users
  - GSM evolved from former car telephony standards (e.g. Natel C)

- Home network:
  - Network of your service provider (e.g. Swisscom, Orange, …)
  - Home Location Register (HLR) : stores profile information (services, billing, preferences) of all customers of a network operator

- Visited network:
  - Network in which mobile user currently resides
  - Visitor Location Register (VLR): contains an entry for each user currently in the network, entry for a mobile user is copied from the HLR of your home network

Addressing Scheme for Cellular Subscribers

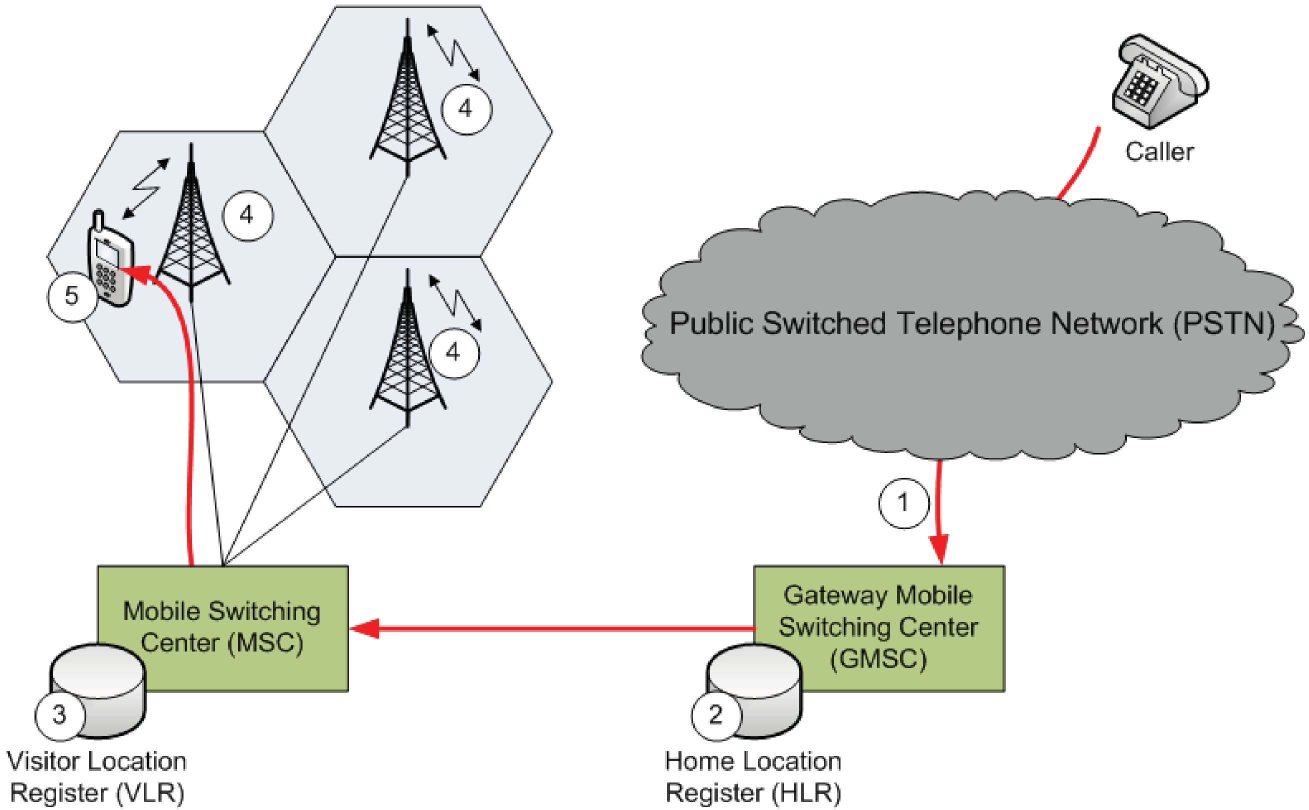
- Global addressing scheme for mobile users
  - Mobile phone numbers need to be globally unique
  - Hierarchical addressing (country, operator, subscriber)

- International Mobile Subscriber Identity (IMSI)
  - Stored on the Subscriber Identity Module (SIM) card
  - 3 digits Mobile Country Code (MCC)
  - 2 digits Mobile Network Code (MNC)
  - Max. 10 digits mobile station identification number
  - Example for an IMSI:

    `228 01 1234567`

    (228 = Switzerland, 01 = Swisscom, 1234567 = identification number)
    Corresponds to the international phone number `+41 79 123 45 67`

Mobile Terminated Call

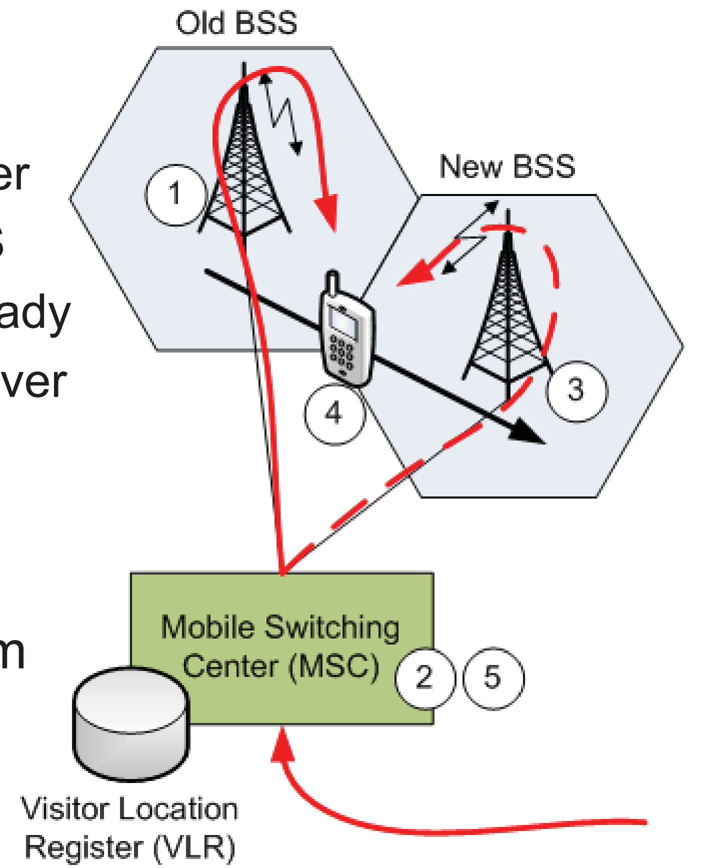


1. Calling a GSM subscriber, call is forwarded to the GMSC
2. Look-up of the current location in the HLR
3. Call is forwarded to the responsible MSC
4. All base stations controlled by the MSC start paging the subscriber
5. Mobile subscriber answers, security checks, call setup

GSM Handover

- Intra-network mobility:
  - User changes between different base stations (BSS) of the same network operator (handover)

- Seamless handover (no call drop)
  1. Old BSS informs MSC of pending handover
  2. MSC sets up path (resources) to new BSS
  3. New BSS signals to MSC and old BSS: ready
  4. Old BSS tells subscriber to perform handover
  5. MSC re-routes call to the new BSS,
     old BSS releases resources

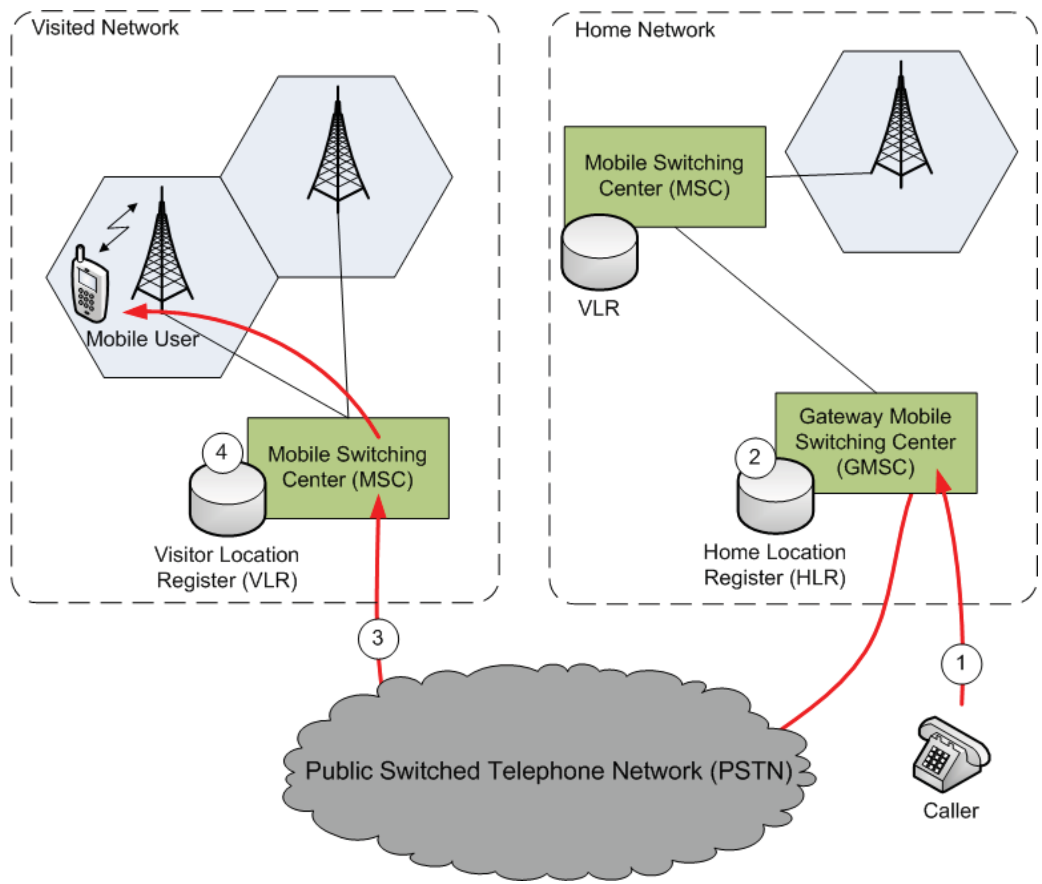


- Sometimes it is even necessary to perform a handover between different MSCs

GSM Roaming

- Inter-network mobility:
  - Home Location Register(HLR) keeps track of the current location
  - Mobile user can be addressed by a temporary Mobile Station Roaming Number (MSRN)

1. Call is forwarded to home network GMSC
2. Location look-up in HLR, obtain temporary MSRN
3. Call is forwarded to visitor MSC over PSTN
4. Lookup location in VLR, forward call to user



Mobility: IP versus GSM

Mobile IP	GSM	Comment
Home Agent (HA)	Gateway Mobile Switching Center (GMSC) and Home Location Register (HLR)	Point of contact to obtain the current address (IP) or roaming number (GSM) of the mobile user
Foreign Agent (FA)	Visited Mobile Switching Center (MSC) and Visitor Location Register (VLR)	Stores temporary information about the mobile user
Care-of Address (COA)	Mobile Station Roaming Number (MSRN)	Temporary address (IP) or number (GSM) for the mobile user

Location services

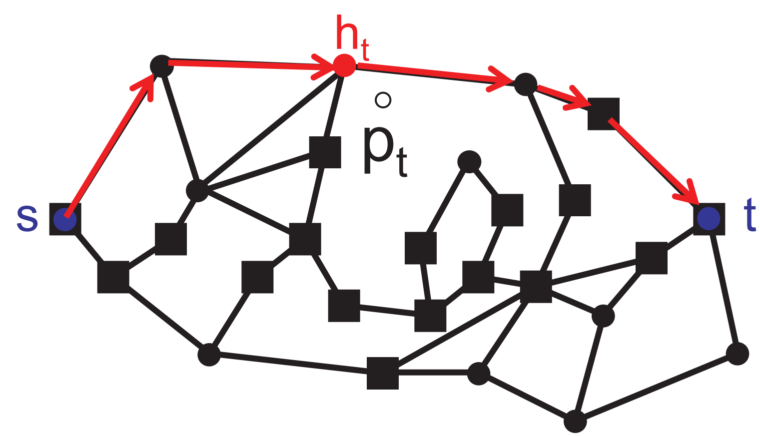
- **Service that maps node names to (geographic) coordinates**
  - Should be distributed (no require for specialized hardware)
  - Should be efficient
- **Lookup of the position (or COA) of a mobile node**
  - Mobile IP: Ask home agent
  - Home agent is determined through IP (unique ID) of MN
  - Possibly long detours even though sender and receiver are close
  - OK for Internet applications, where latency is (normally) low
- **Other application: Routing in a MANET**
  - MANET: mobile ad hoc network
  - No dedicated routing hardware
  - Limited memory on each node: cannot store huge routing tables
  - Nodes are mostly battery powered and have limited energy
  - Nodes route messages, e.g. using georouting

Home based georouting in a MANET

- **How can the sender learn the current position of another node?**
  - Flooding the entire network is undesirable (traffic and energy overhead)
- **Home based approach**
  - Similar to Mobile IP, each node has a *home* node, where it stores and regularly updates its current position
  - The home is determined by the unique ID of the node *t*. One possibility is to hash the ID to a position p_t and use the node closest to p_t as home.
  - Thus, given the ID of a node, every node can determine the position of the corresponding home.

Home based routing

1. Route packet to h_t, the home of the destination t
2. Read the current position of t
3. Route to t



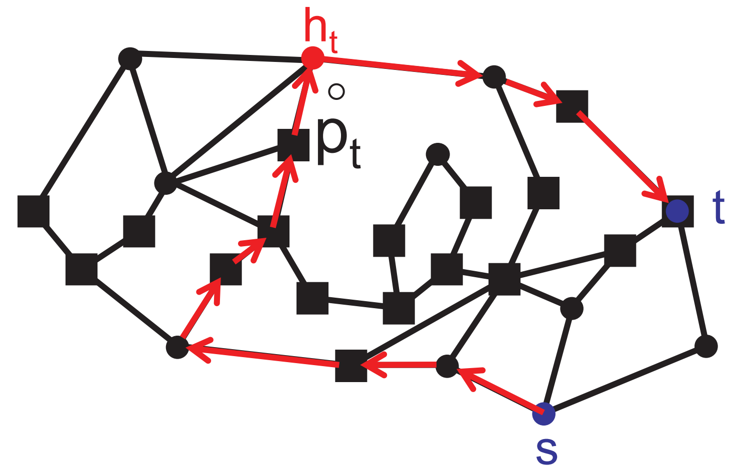
Home based location service – how good is it?

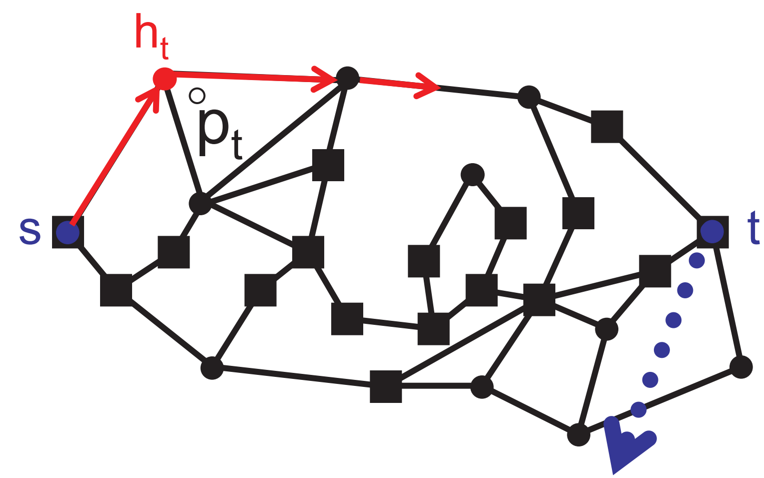
- Visiting the home of a node might be wasteful if the sender and receiver happen to be close, but the home far away
- The routing stretch is defined as

$$\text{stretch} := \frac{\text{length of route}}{\text{length of optimal route}}$$

We want routing algorithms with low stretch.

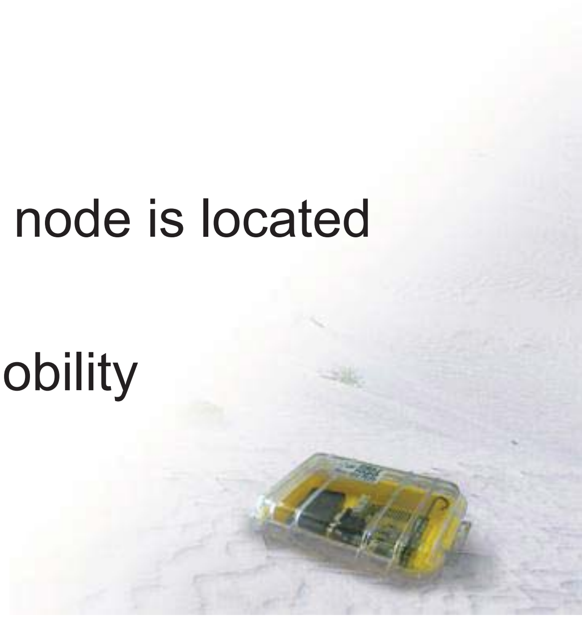
- Simultaneous message routing and node movement might cause problems
- Can we do better?





Classification of location services

- **Proactive**
  - Mobile node divulges its position to all nodes whenever it moves
  - E.g. through flooding
- **Reactive**
  - Sender searches mobile host only when it wants to send a message
  - E.g. through flooding
- **Hybrid**
  - Both, proactive and reactive.
  - Some nodes store information about where a node is located
  - Arbitrarily complicated storage structures
  - Support for simultaneous routing and node mobility



Location services: Lookup & Publish

- **Any node *A* can invoke to basic operations:**
  - Lookup(*A, B*): *A* asks for the position of B
  - Publish(*A*, *x*, *y*): *A* announces its move from position *x* to *y*

- **Open questions**
  - How often does a node publish its current position?
  - Where is the position information stored?
  - How does the lookup operation find the desired information?

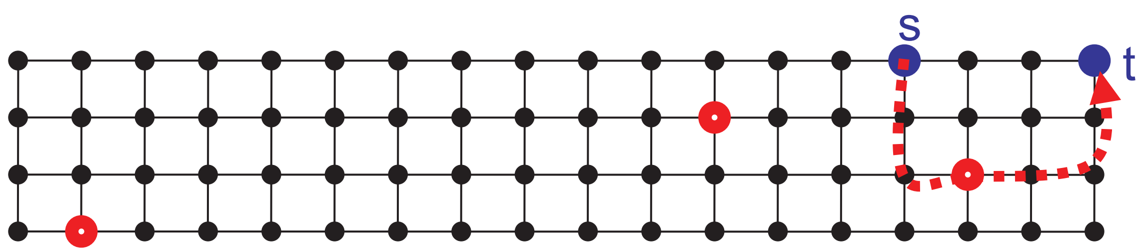
- **Goal**
  - Minimize stretch = $\dfrac{\text{length of route}}{\text{length of } \textit{optimal} \text{ route}}$

Location Service: Goals

- **Publish cost** should depend only on moved distance
- **Lookup cost** should depend only on the distance between the sender and receiver (bounded *stretch*)
- Nodes might **move arbitrarily at any time**, even while other nodes issue lookup requests
- Determine the **maximum allowed node speed** under which delivery is still guaranteed

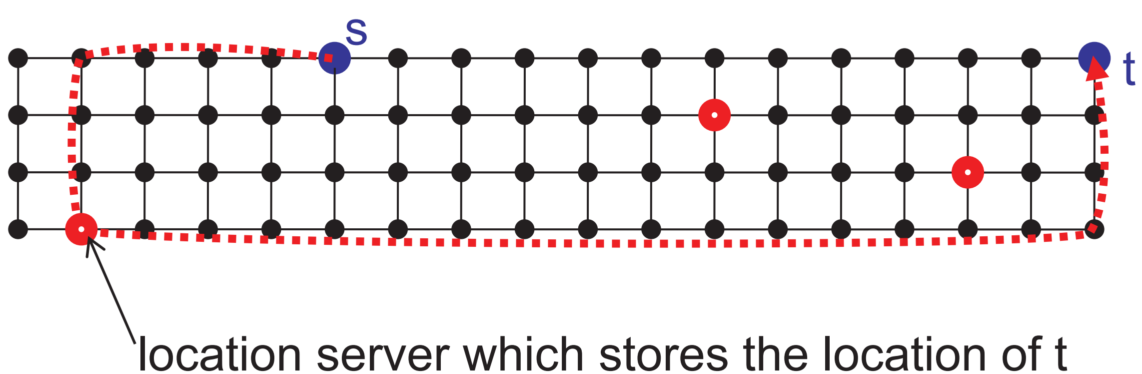
Location Service: Initial observations

- Cannot get reasonable stretch with a single home. Therefore, use several homes (**location servers**).
- If the sender *s* is close to the destination *t*, a location server should be close



$$\text{stretch} = \frac{7}{3} = 2.33$$

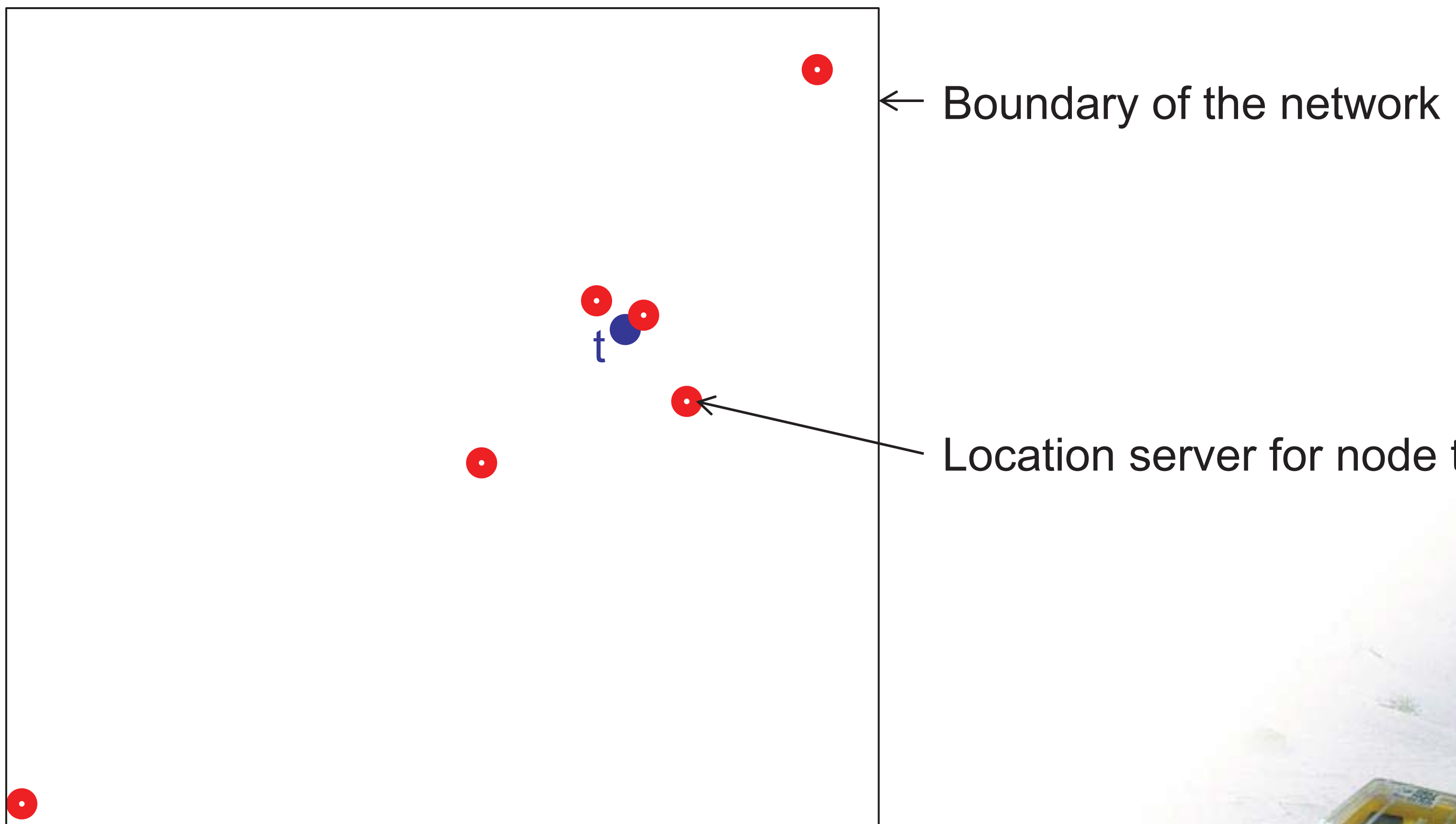
- If the sender is further away, its OK to walk a bit longer until finding a location server



$$\text{stretch} = \frac{26}{12} = 2.17$$

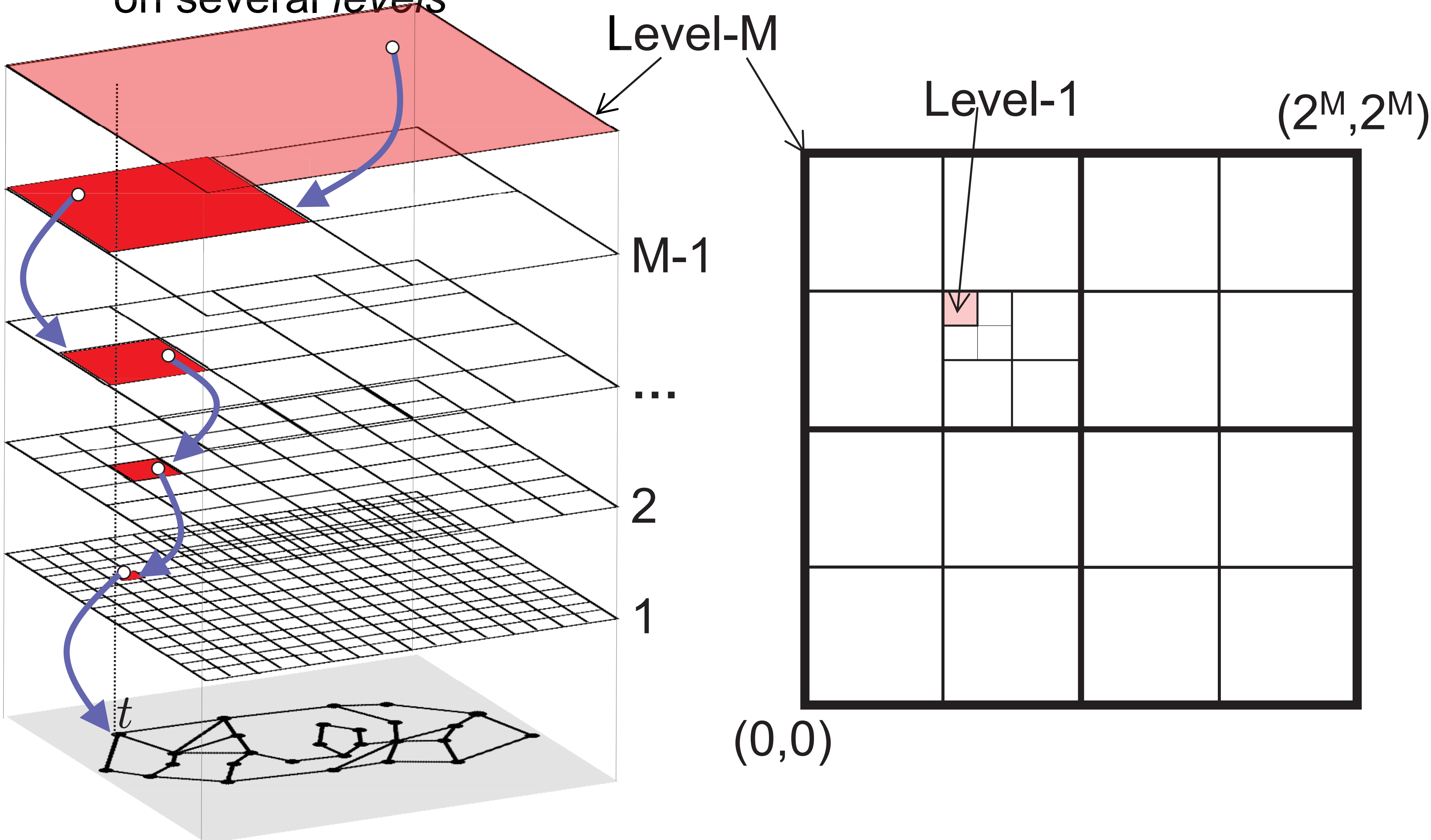
Location Service: Initial observations

- Many location servers close to the node
- Sparser distribution of location servers the further away we are



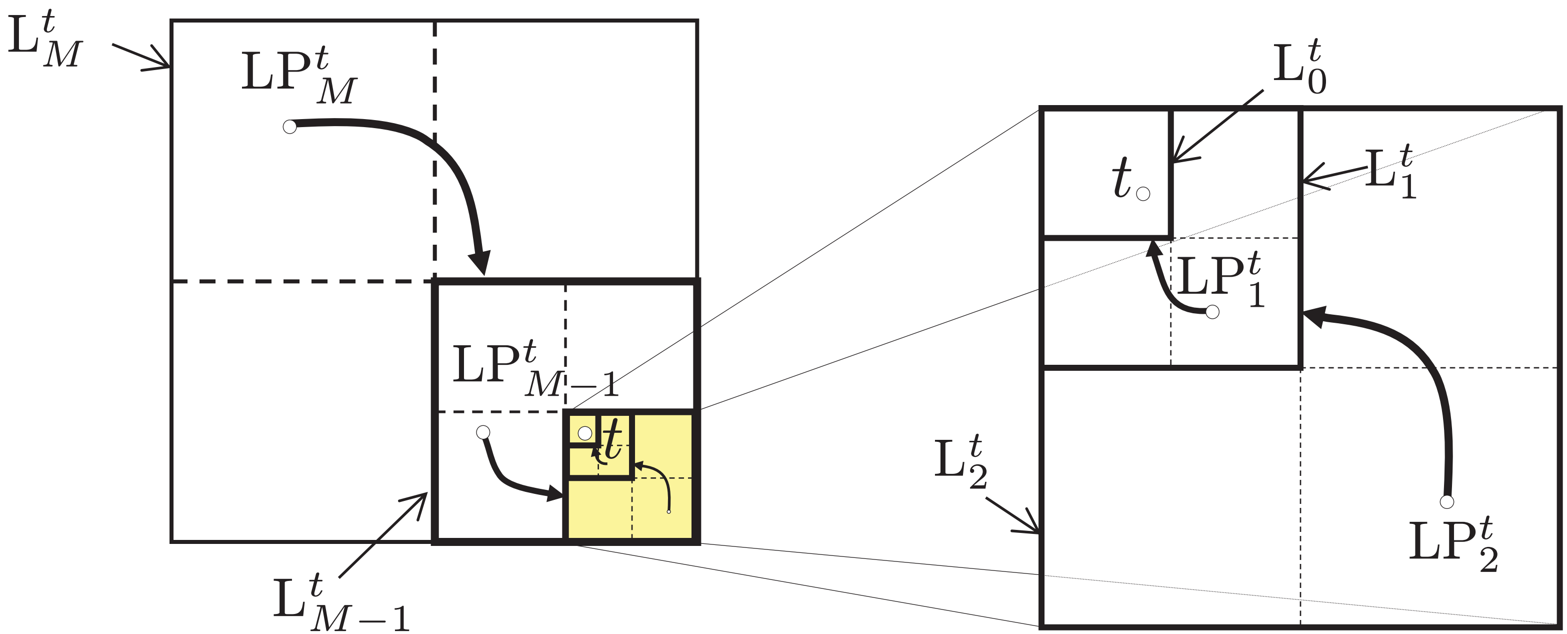
MLS: Location Service for Mobile Ad Hoc Networks

- Geometric decomposition of the network into quadratic cells on several *levels*



Location pointers (aka location servers)

- On every level, *t* stores one location pointer in the cell containing *t*
  - L^t_M denotes the quadratic cell on level *M* containing *t*
  - On level k, the *location pointer* for *t* is called LP^t_M

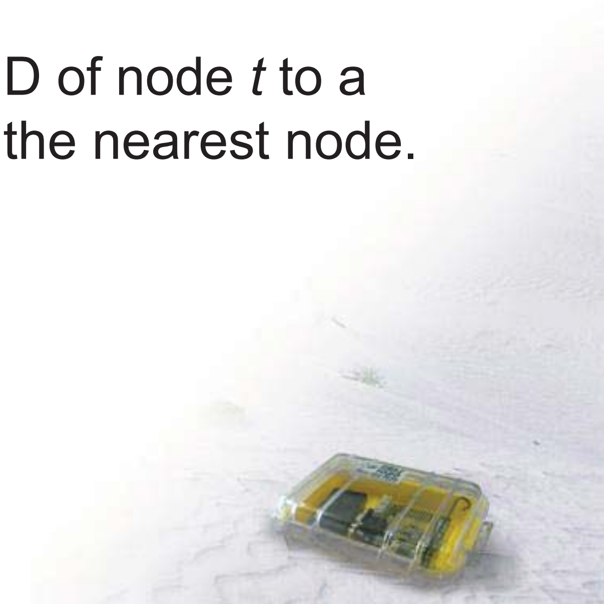


Location pointer & Notation

- Notation:
  - LP_k^t Location pointer for node *t* on level-*k*
  - L_k^t Level-*k* cell that contains node *t*

- The location pointers are placed depending on the ID of the node, as in the home-based lookup system.

- The position of LP_k^t is obtained by hashing the ID of node *t* to a position in L_k^t . The location pointer is stored on the nearest node.



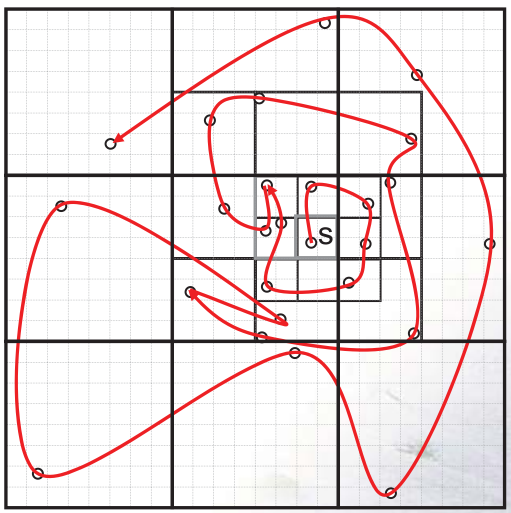
Routing in MLS

- Routing from a node *s* to a node *t* consists of two phases:
    1. Find a location pointer LP_k^t
    2. Once a first location pointer is found on level-*k*, we know in which of the 4 sub-squares *t* is located and thus in which L_{k-1} *t* has published its location pointer LP_{k-1}^t.
       Recursively, the message is routed towards location pointers on lower levels until it reaches the lowest level, from where it can be routed directly to *t*.

Routing in MLS (2)

- When a node *s* wants to find a location pointer of a node *t*, it first searches in its immediate neighborhood and then extends the search area with exponential growing coverage.
  - First, try to find a location pointer LP_0^t in L_0^s or one of its 8 neighboring levels.
  - Repeat this search on the next higher level until a LP_k^t is found

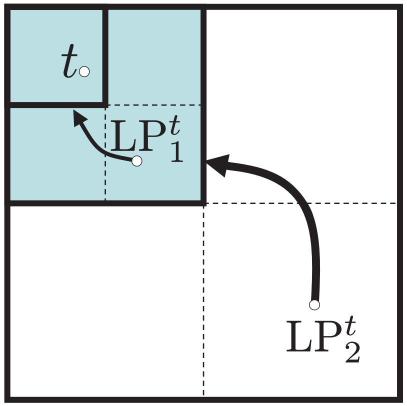
- The lookup path draws a spiral-like shape with exponentially increasing radius until it finds a location pointer of *t*.
- Once a location pointer is found, the lookup request knows in which sub-square it can find the next location pointer of *t*.

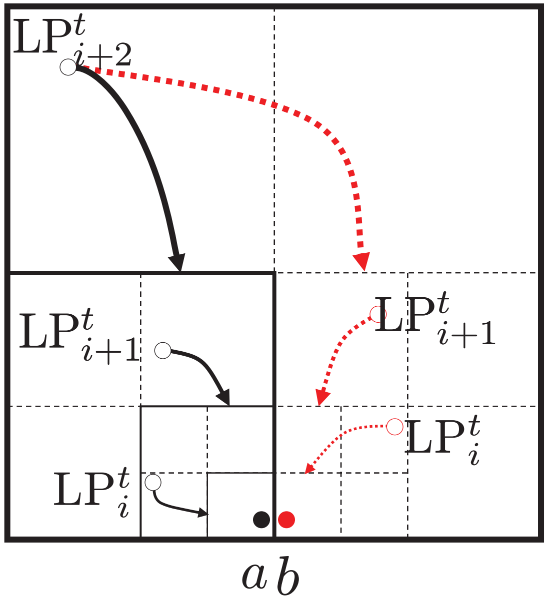


Support for mobility in MLS

- A location pointer only needs to be updated when the node leaves the corresponding sub-square.
  - LP_2^t is OK as long as t remains in the shaded area.
  - Most of the time, only the closest few location pointers need to be updated due to mobility.

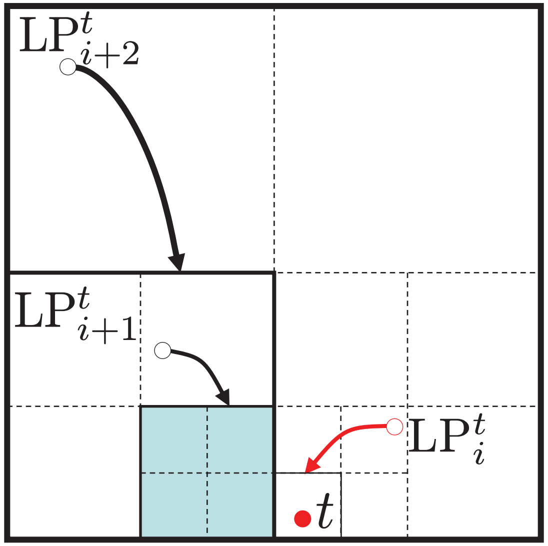
- Not enough: If a node moves across a level boundary, many pointers need to be updated. E.g. a node oscillates between the two points a and b.





Lazy publishing

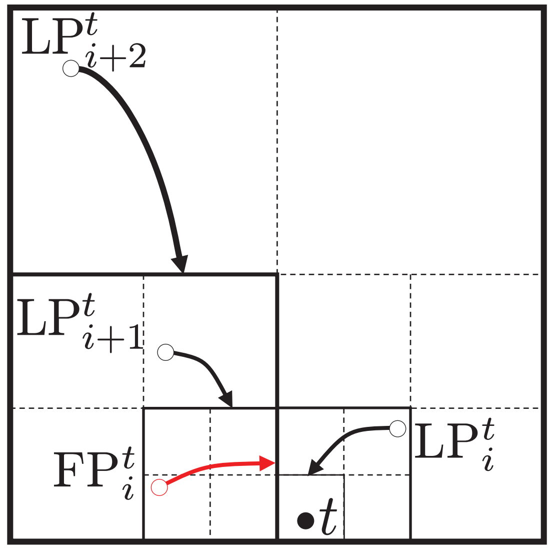
- Idea: Don't update a level pointer LP_k^t as long as *t* is still somewhat close to the level L_k where LP_k^t points.



- Breaks the lookup: LP_{i+1}^t points to a level that does not contain LP_i^t

Lazy publishing with forwarding pointers

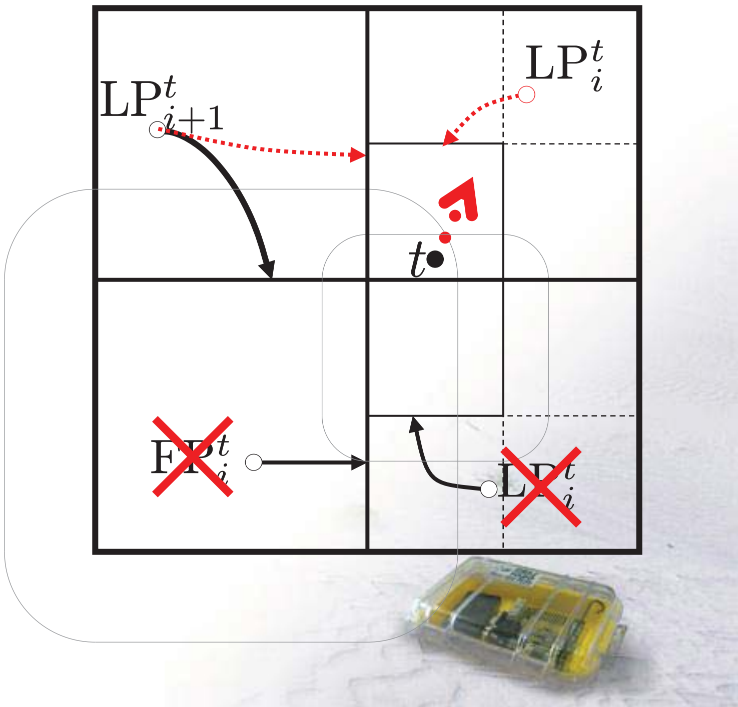
- No problem, add a **forwarding pointer** that indicates in which neighboring level the location pointer can be found.



Concurrency in MLS

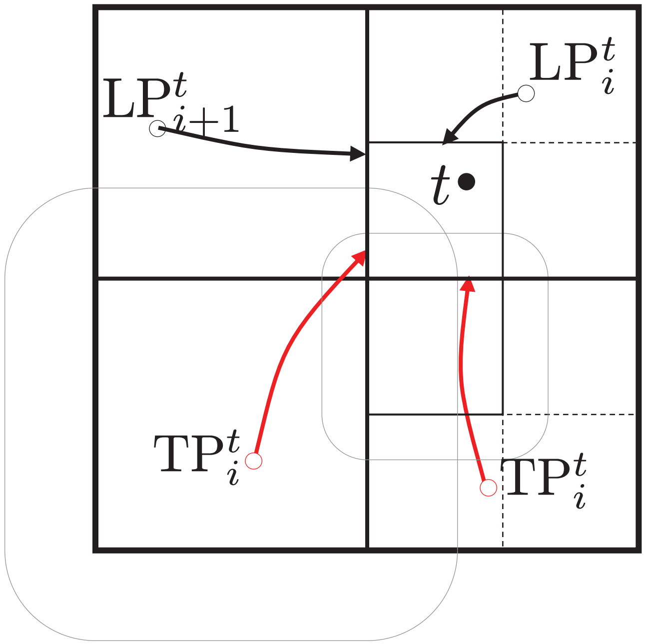
- Allowing for concurrent lookup requests and node mobility is somewhat tricky, especially the deletion of pointers.
- Note that a lookup request needs some time to travel between location pointers. The same holds for requests to create or delete location (or forwarding) pointers.

- Example:
  - A lookup request follows LP_{i+1}^t, and node *t* moves as indicated
  - *t* updates its LP_i^t and LP_{i+1}^t and removes the FP_i^t and the old LP_i^t
  - The lookup request fails if it arrives after the FP_i^t has been removed



Concurrency in MLS (2)

- No problem either: Instead of removing a location pointer or forwarding pointer, replace it with a **temporary pointer** that remains there for a *short time* until we are sure that no lookup request might arrive anymore on this outdated path.
- Similar to the forwarding pointer, a temporary pointer redirects a lookup to the neighbor level where the node is located.



Properties of MLS

- Constant lookup stretch
  - The length of the chosen route is only a constant longer than the optimal route
- Publish cost is O(*d* log *d*) where moved distance is *d*
  - Even if nodes move considerably, the induced message overhead due to publish requests is moderate.
- Works in a concurrent setup
  - Lookup requests and node movement might interleave arbitrarily
- Nodes might not move faster than 1/15 of the underlying routing speed
  - We can determine the maximum node speed that MLS supports. Only if nodes move faster, there might arise situations where a lookup request fails.

MLS Conclusions

- It's somewhat tricky to handle concurrency properly
  - Use of temporary forwarding pointers
- MLS is the first location service that determines the maximum speed at which nodes might move
  - Without the speed limitation, no delivery guarantees can be made!
- Drawbacks
  - MLS utilizes an underlying routing algorithm that can deliver messages with constant stretch given the position of the destination
  - MLS requires a relatively dense node population

History of location services

- Grid Location Service (GLS) by Li et al. (2000)
  - No bound on the stretch
- Locality Aware Location Service (LLS) by Abraham et al. (2004)
  - Similar to MLS
  - No concurrency support
- MLS: Location Service for Mobile Networks by Flury et al. (2006)

- … Still many open problems to solve

Mobility Models

- Mobility is an important aspect when designing protocols for wireless ad-hoc networks
  - Node density depends highly on the mobility pattern
  - Links break more frequently when the node mobility is high

- When studying mobility, one might resolve to models
  - Simulations of mobility models are fast, cheap and repeatable
  - Real-world experiments are often infeasible

- Mobility models specify how nodes move in a certain area of interest
  - Many different ideas how to model mobility (e.g. physics, road traffic)
  - Only key features are taken into account to keep the model simple (but often too simplified)
  - Should match the real-world behavior of nodes as good as possible

Mobility Model Classification

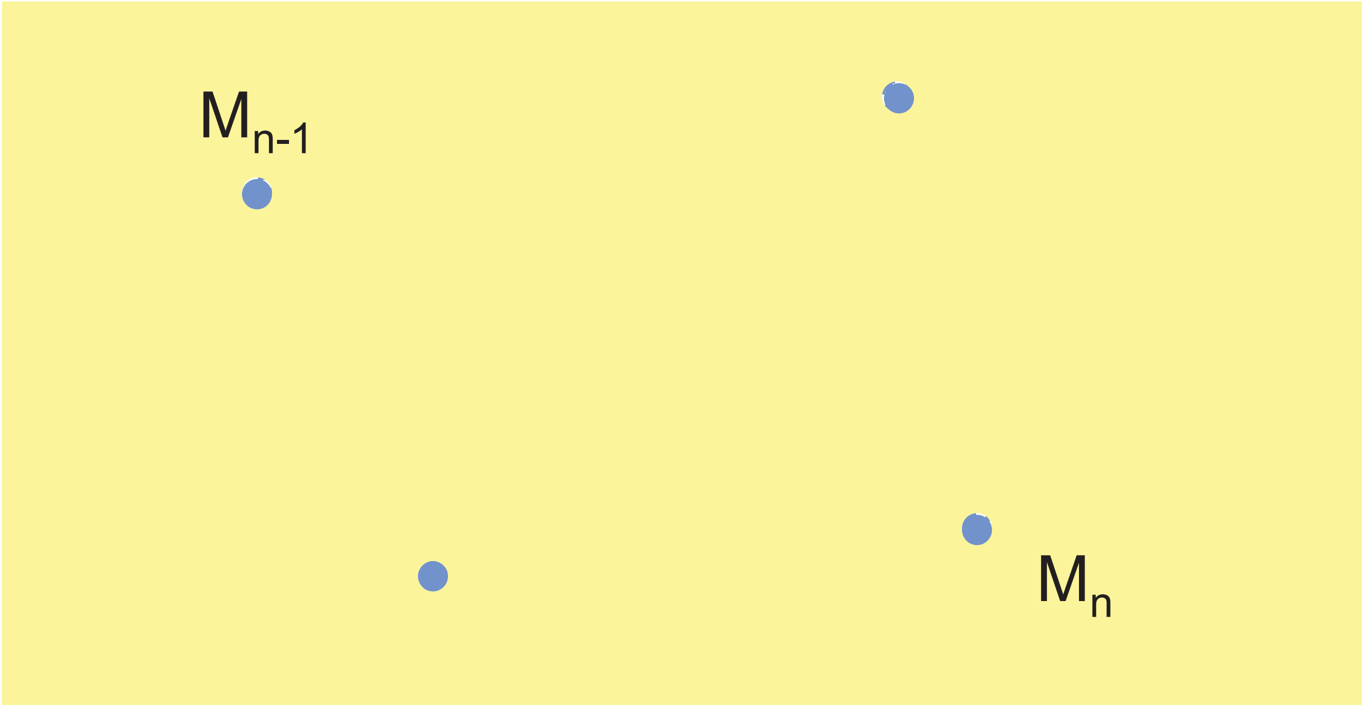
- Random entity-based mobility models
  - Each node acts independently of other nodes (e.g. Brownian motion, Random waypoint, Random trip)
- Group-based mobility models
  - Simulate the behavior of nodes moving together towards a common destination (e.g. Reference Point Group Mobility model)
- Survey-based mobility models
  - Street maps, augmented with info what kind of people live/work/shop/… in which areas, and how do they move between these areas
  - This is also popular in research about viruses and diseases
- Vehicular mobility models
  - Model the behavior of vehicles participating in road traffic
  - Driver models, intersections, congestion
  - Traffic simulators

The Random Waypoint Model

[following slides by JY Le Boudec, EPFL]

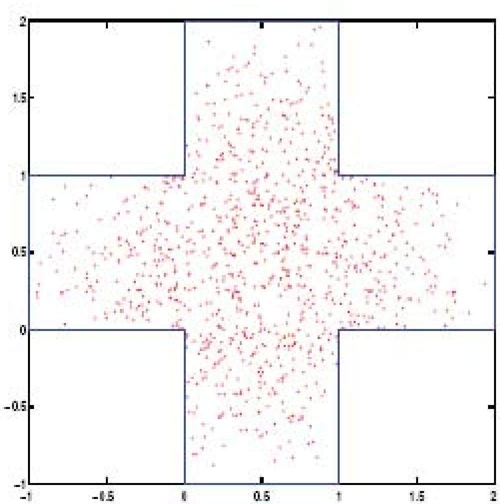
In its simplest form:

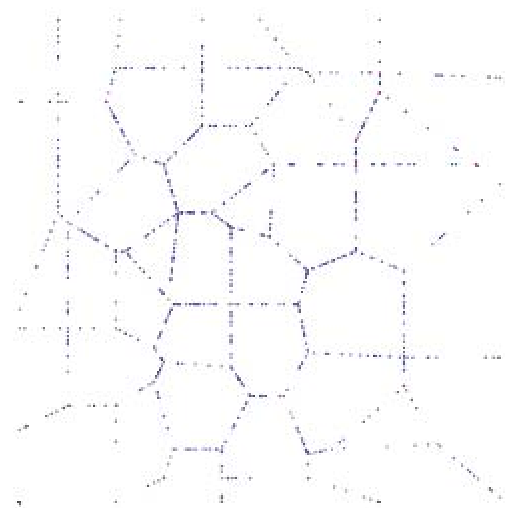
- Mobile picks next waypoint M_n uniformly in area
- Mobile picks next speed V_n uniformly in $[v_{min}; v_{max}]$
- Both independent of past and present
- Mobile moves towards M_n at constant speed V_n

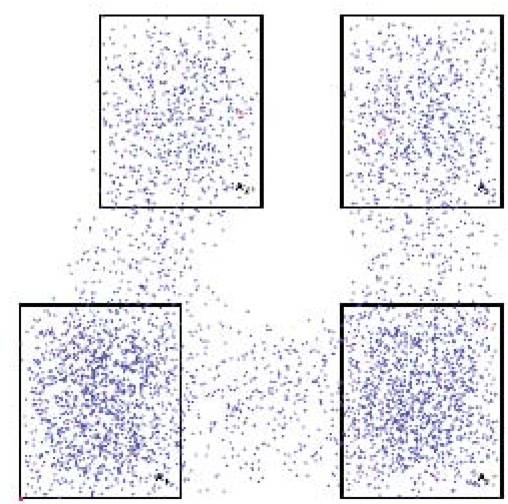


The Random Trip model

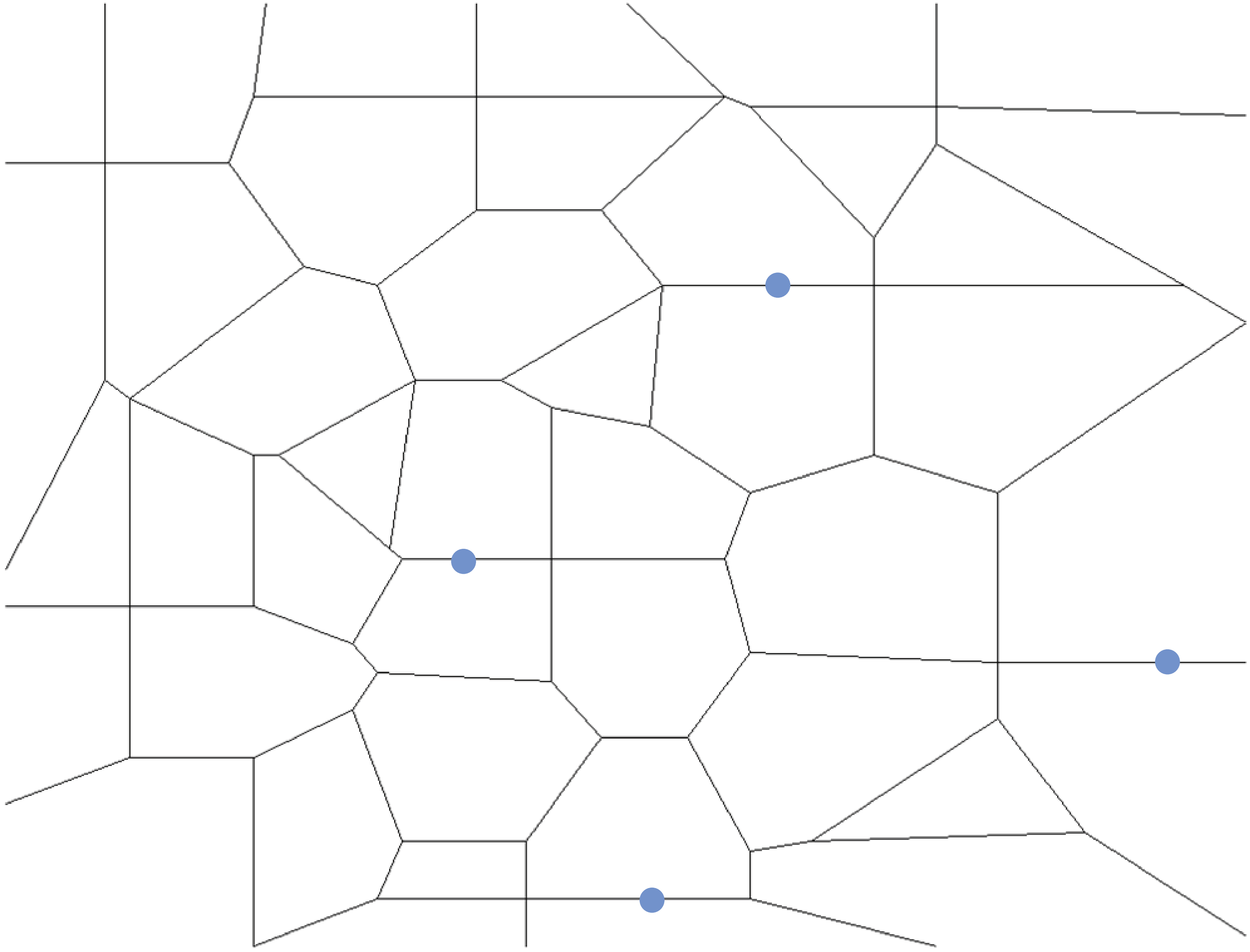
- Random Waypoint is a special case of Random Trip:
  - Mobile picks a path in a set of paths and a speed
  - At end of path, mobile picks a new path and speed
  - Mobiles may decide to wait and sleep at destinations before going on the next leg
  - E.g. shortest Euclidean path in non-convex area, or shortest path on street map





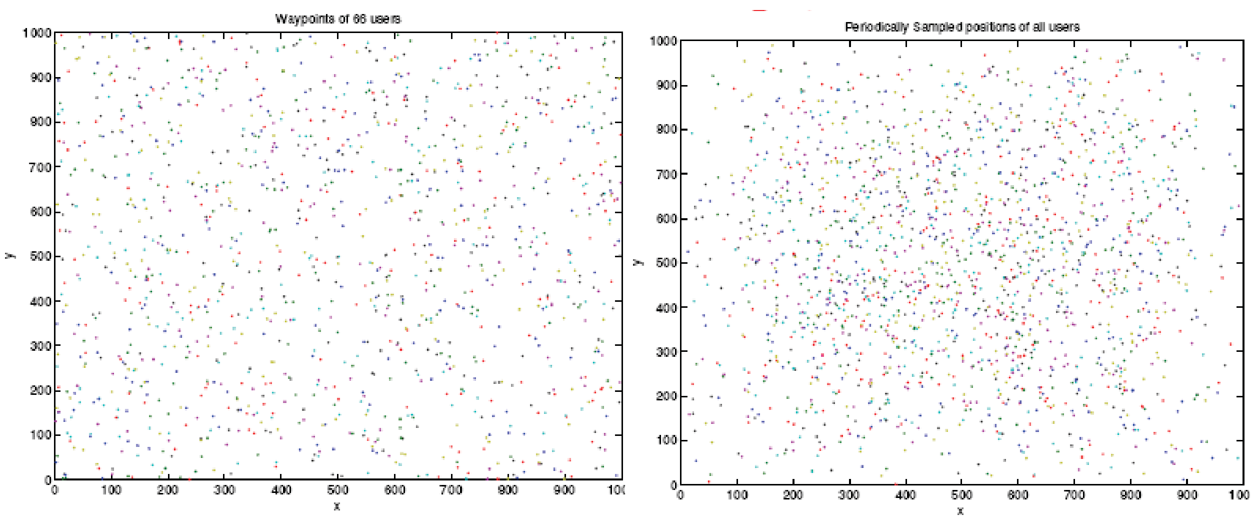


Example: Random Trip

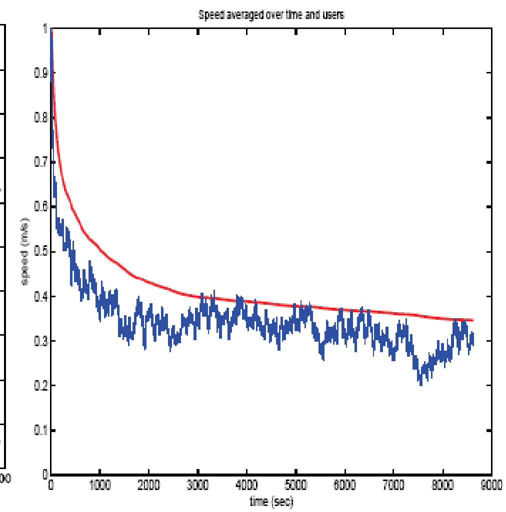


Simulation problems

- If you simulate mobility, you need to take care about system leveling off…



Samples of location at times 0s and 2000s

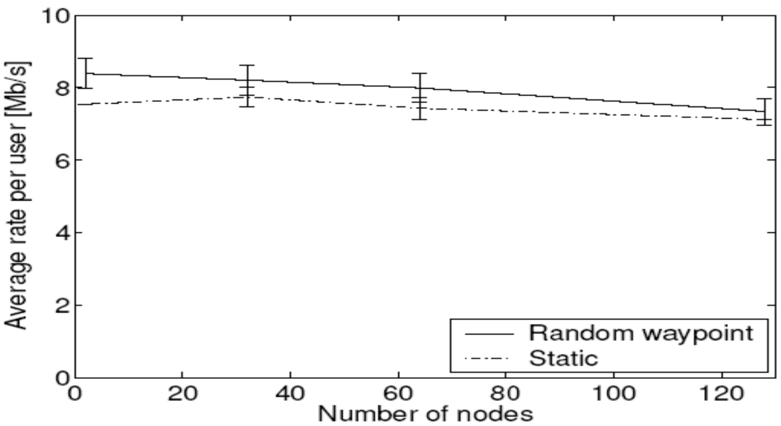


Average node speed

- The problem is that the steady-state is in the infinite…

Simulation “solutions”

- The problem is that your simulations may show results which differ from “reality”.

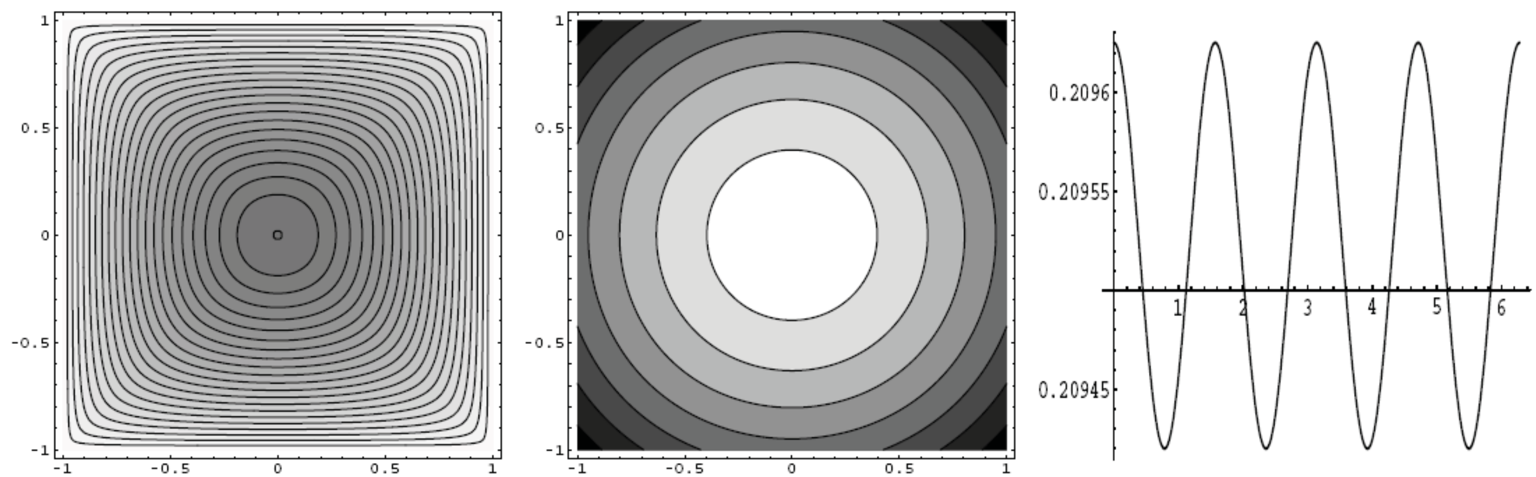


- A simple rule of thumb (which is wrong, but somehow “acceptable”): If you want to simulate for time T, you really need to simulate time 2T, and throw the first half of your simulation away.

- Another (also wrong) solution is to start each node at a position p which is uniformly random between uniformly random points s and t, and with velocity according to the distribution 1/v. However, also this is not correct…

Simulation solution

- The correct solution is to simultaneously draw position and velocity from the steady-state distribution (see work by Le Boudec for details.)

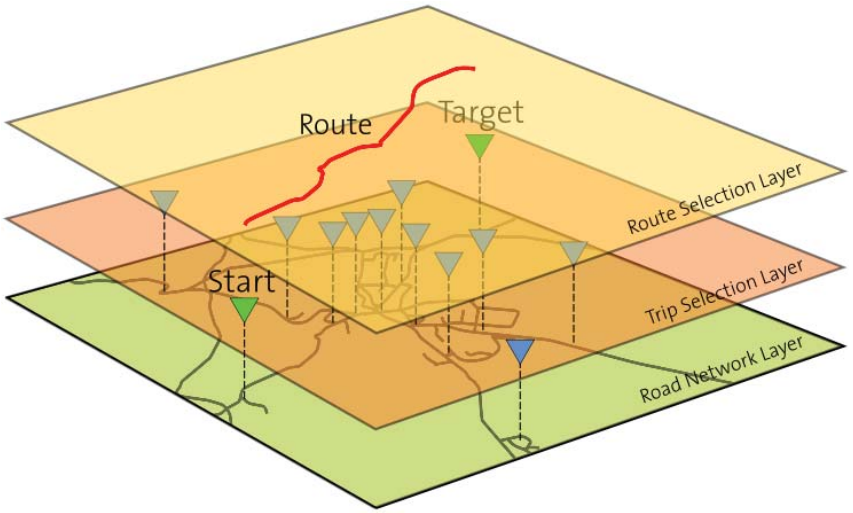


(a) Node Location (b) Next Waypoint (c) Next Waypoint

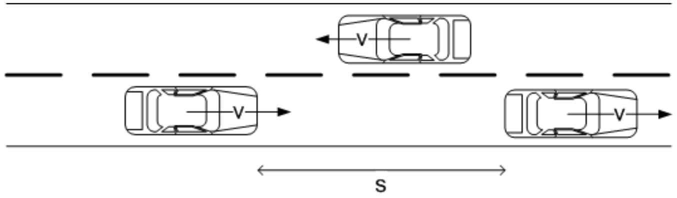
Figure 9: Density of the time-stationary distribution of (a) node location $M(t)$ and (b) the previous (or next) waypoint. The contour plots (a) and (b) show lines of equal density. The difference in density between two adjacent lines is 0.025. In (a) the density decreases from 0.5503 (center) to 0 (edge); in (b) it increases from 0.1834 to 0.3669. The density in (b) appears to have circular symmetry, but this is not exactly true: (c) plots, for the same case as (b), the density as a function of θ for points of the form $x = 0.5\cos(\theta), y = 0.5\sin(\theta)$ (i.e. points that are at a fixed distance $r = 0.5$ from the center).

Vehicular Mobility Models

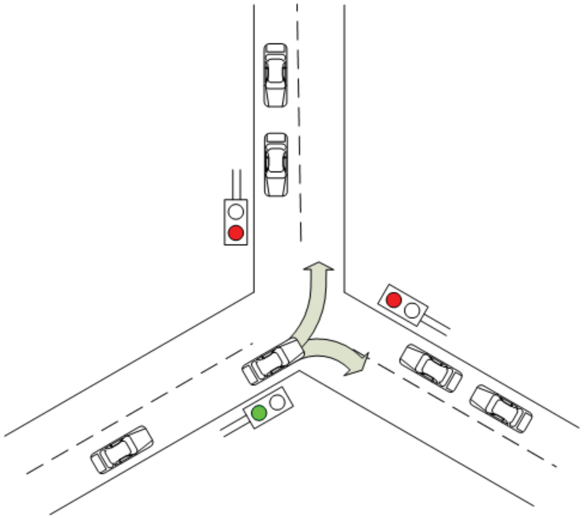
- Global behavioral rules
  - Random trips constrained to a realistic road network (GIS)



- Local behavioral rules
  - Speed adjustement and car-following (keep minimal distance to the car in front)

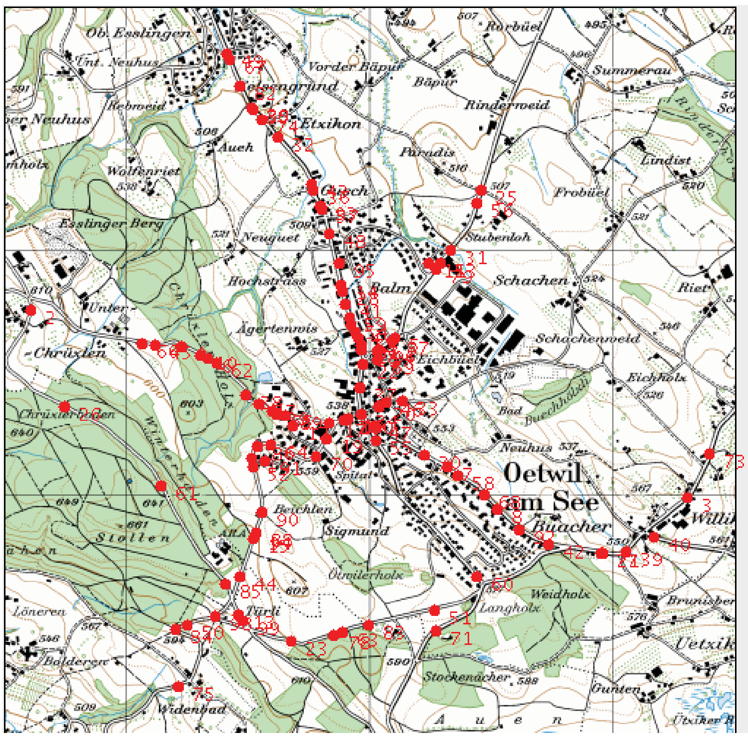


  - Intersection management (stop at red traffic lights, then start to accelerate again on green light)

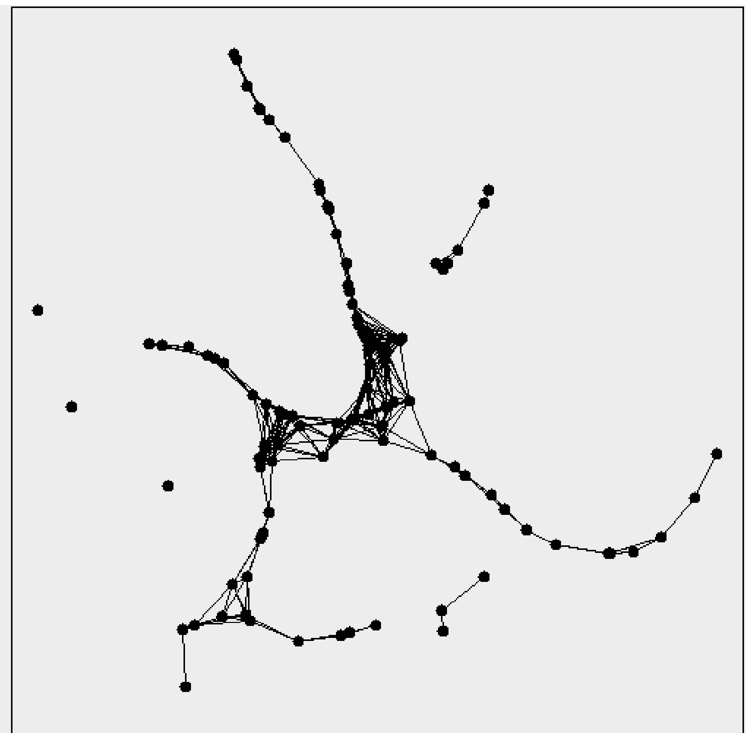


Example: Vehicular Mobility using GIS data

- Use the road topology from a geographic information system (GIS) to model vehicular mobility for a simulation area
  - Information about road category, speed limits
  - Very detailed geographical data available (resolution is around 1m)



Map

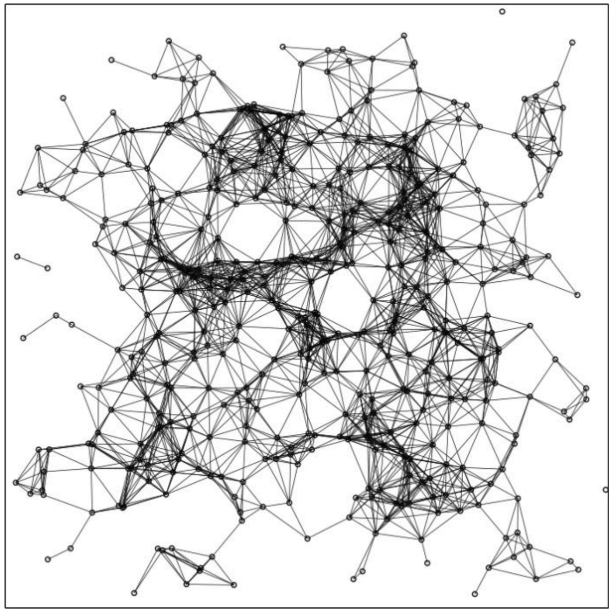


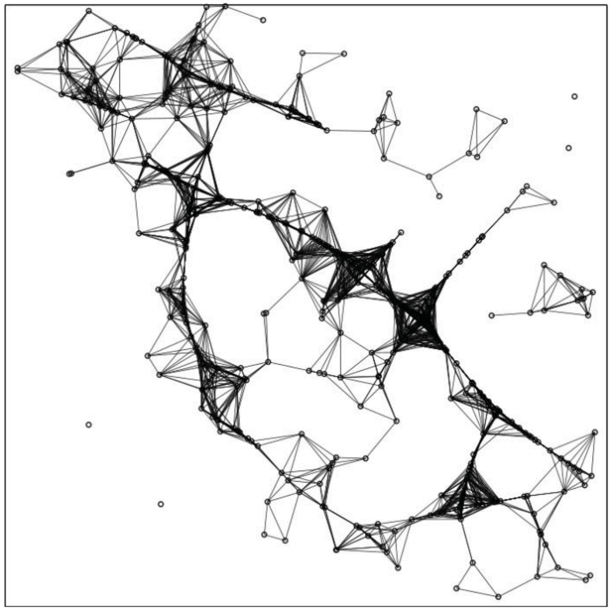
Network graph

Impact of the Mobility Model on the Network Topology

- Choosing the „right“ mobility model is important
  - Should match the reality as close as possible
  - Mobility model has a large impact on the node distribution and therefore also on the network topology

- Example: Network topolgy for Random Waypoint (left) and a vehicular mobility model(right)





Open problem

- Even systems like MLS still make way too many simplifying assumptions. So there is the obvious question about a location service which is practical.

- Essentially a good location service system needs to
  1. work in dynamic environments
  2. give acceptable memory and communication loads
  3. provide stretch guarantees
  4. neither make funny assumptions about node distributions …
  5. … nor about mobility patterns
  6. be secure