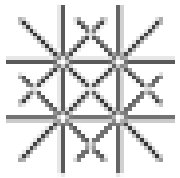
Dynamic Names and Private Address Maps: Complete Self-Configuration for MANETs

Christophe Jelger, Christian Tschudin

www.ana-project.org



UNIVERSITÄT BASEL

CoNEXT, Lisboa, Dec. 2006.

Quick overview of this presentation

- Main motivation: MANET is not Internet
  - Is IP routing and addressing the right way to go?
- Where addresses are not addresses in MANETs.
  - Why "routing by name" is a valid alternative.
- Private Address Maps:
  - A 100% NAT proposal.
- Dynamic Shortest Discriminating Names:
  - The human to network interface.

MANET (address) Autoconfiguration

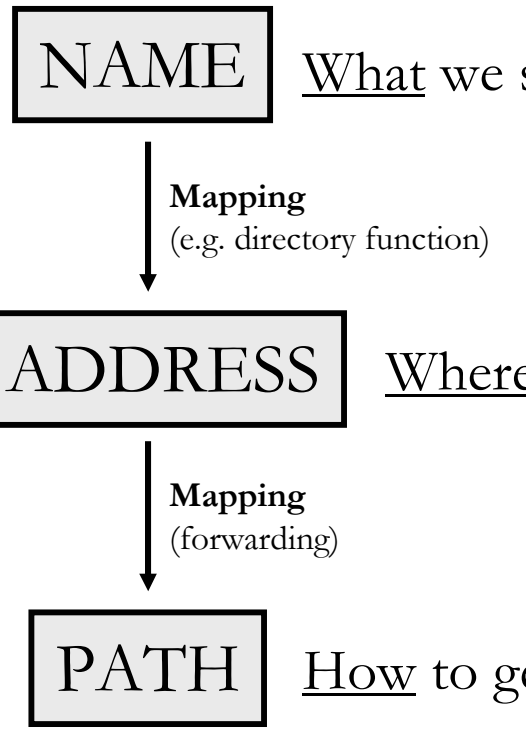
- Very active field of research for the last 5 years
  - However, no clear "winner solution" yet.
  - IETF AUTOCONF Working Group has not yet produced a single wg document !

- Focus is on addresses, addresses, and addresses …
  - Because of IP routing, each address must be unique - challenging in MANETs (no infrastructure, node mobility, network merging and partitioning, etc).
  - Core common issues of all autoconf solutions are
    - address validation ("is this address already used?") – initialization phase
    - collision detection ("is someone using my address?") – continuous phase

Objective of this work: to radically solve address autoconfiguration by "removing" addresses from the MANET networking model!

The MANET world today: "Shoch's world"

✓ Name resolution à la AODV
✗ Name autoconfiguration



✗ Address autoconfiguration

✓ MANET routing (with addresses)

On addresses and paths in MANETs

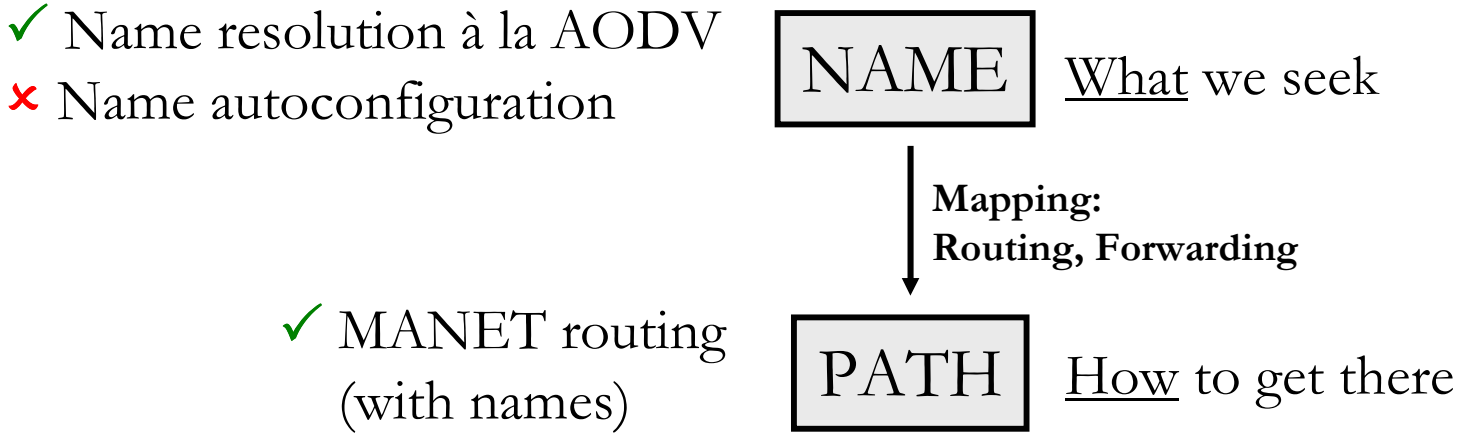
- In Shoch's model, an address indicates where a resource is located.
  - In the Internet, an address has a topological meaning: it indicates a point of attachment.
  - That is, L3 mobility implies an address change: an address is not portable.

- But in a MANET, an address indicates how to get to a node.
  - An address has no topological meaning: routing is typically flat (i.e. a route identifies a unique node).
  - L3 mobility is handled via routing: addresses do not change.
  - Addressing is totally topology independent.
  - Does this sound like Shoch's definition of an address? Our answer: **No**.

Is there an alternative to the name-address-path networking model ?

Revisiting MANETs: removing addresses

The "Routing by name" [Hauzeur86] networking model.



No more issues with addresses:
We "just" need to solve name autoconfiguration
and use a clever way to label paths.

Challenges to achieve routing by name

- Name autoconfiguration.
  - In MANETs, names (FQDNs) should not be assigned statically.
  - Dynamic names? hey this is non-sense ... ;-)
  - Actually a user is interested by what is identified by the name, not by the name itself as long as it "does the job" (i.e. identify the target resource).
- However, today's networking world works in an IP-by-default mode.
  - We cannot get rid of IP addresses without introducing backwards compatibility problems (e.g. existing applications expect DNS+IP).
- Actually the main issue faced by address autoconfiguration is address uniqueness: for IP routing to work, each address must be (globally) unique.
  - BUT we already know how to solve this with NAT.

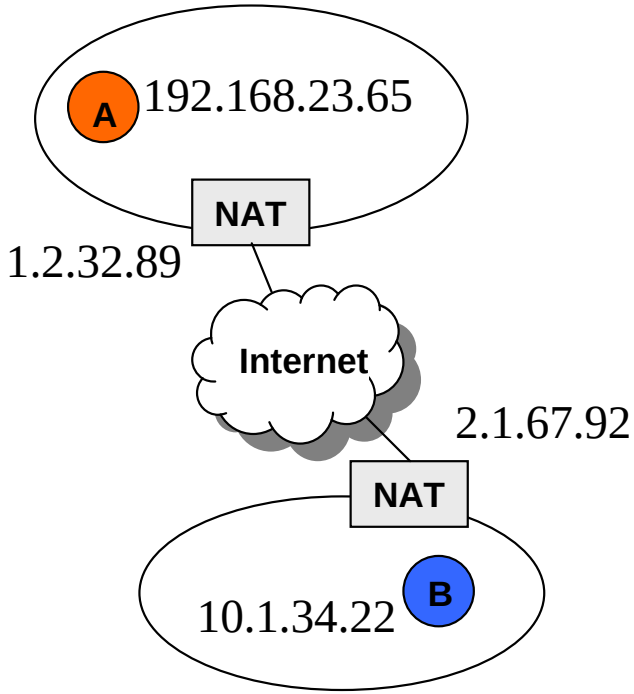
Our proposal: an overview

- Address (no) autoconfiguration.
  - **Private Address Maps**.
  - Based on an extreme use of NAT.
  - Mainly to guarantee backwards compatibility with the IP world.

- Name autoconfiguration.
  - **Dynamic Shortest Discriminating Names**.
  - User specifies a set of ordered keywords used to dynamically create a unique name (inside the MANET).

Private Address Maps

- Inspired by NAT:



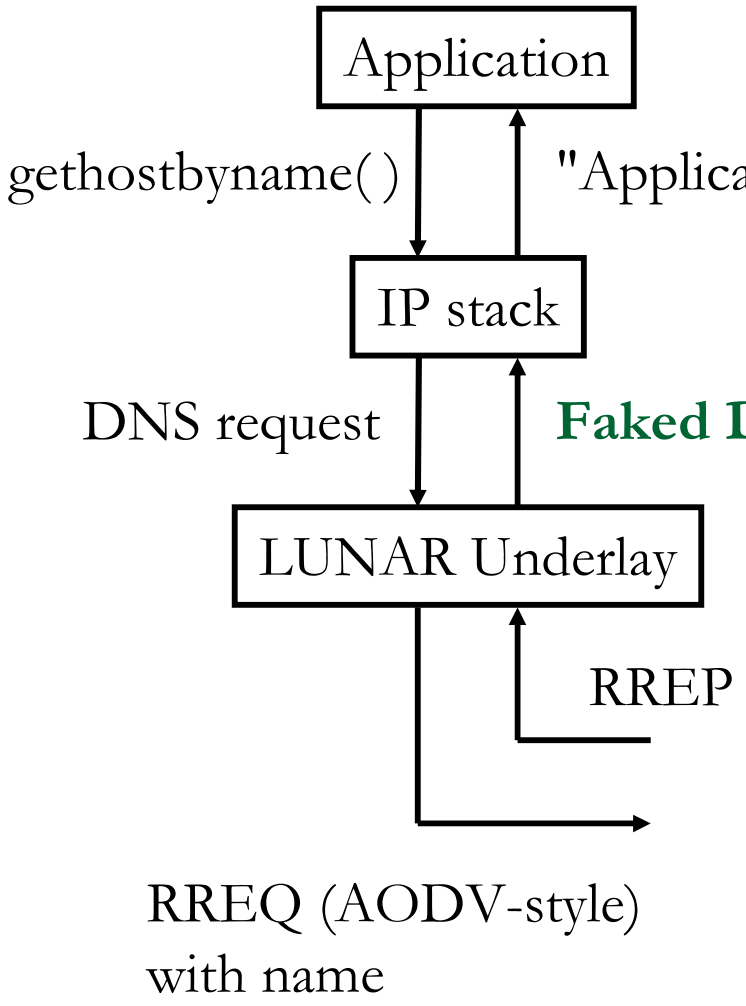
Node A sees:
A = 192.168.23.65
B = 2.1.67.92

≠ ≠

Node B sees:
B = 10.1.34.22
A = 1.2.32.89

Private Address Maps = extension of this scenario to all the MANET nodes (100% NAT)

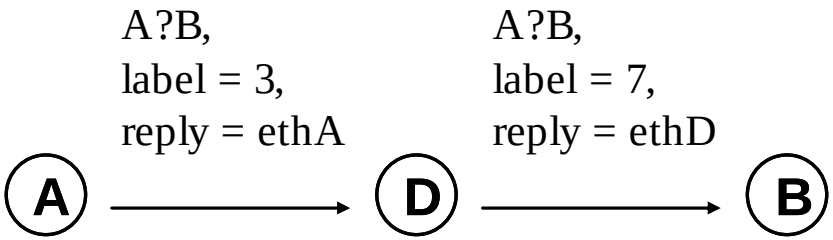
Private Address Maps: let me choose your address



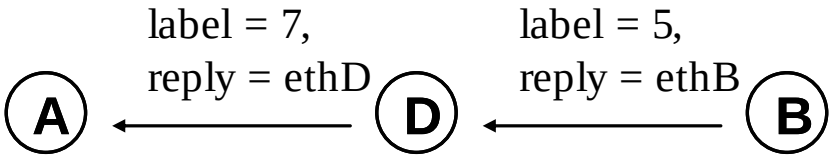
Forwarding cannot be done with IP address (addresses have no global meaning).

Routing creates a label-switched forwarding path à la MPLS (i.e. with local labels).

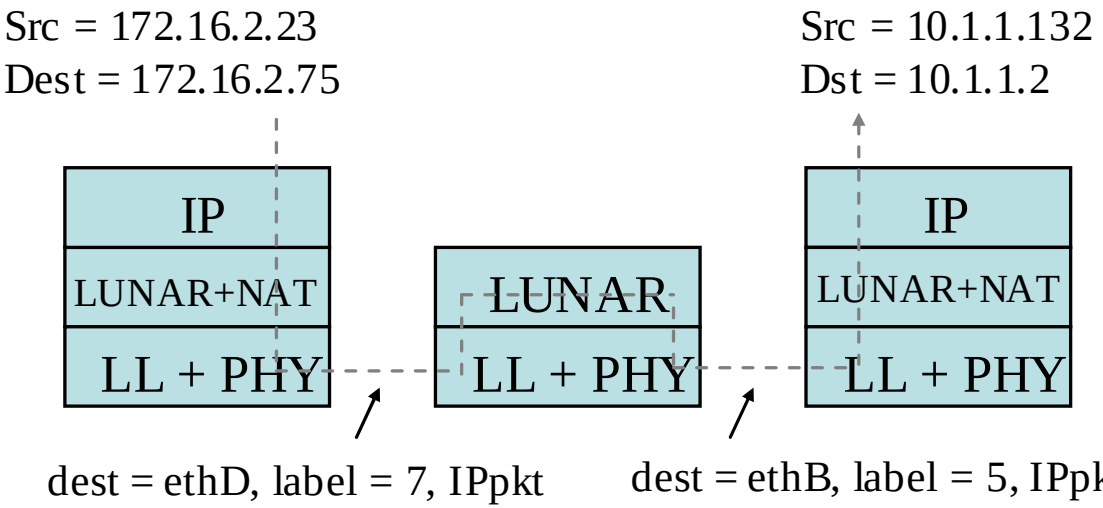
Private Address Maps: lookup + path setup



a) Route request sent by A looking for B

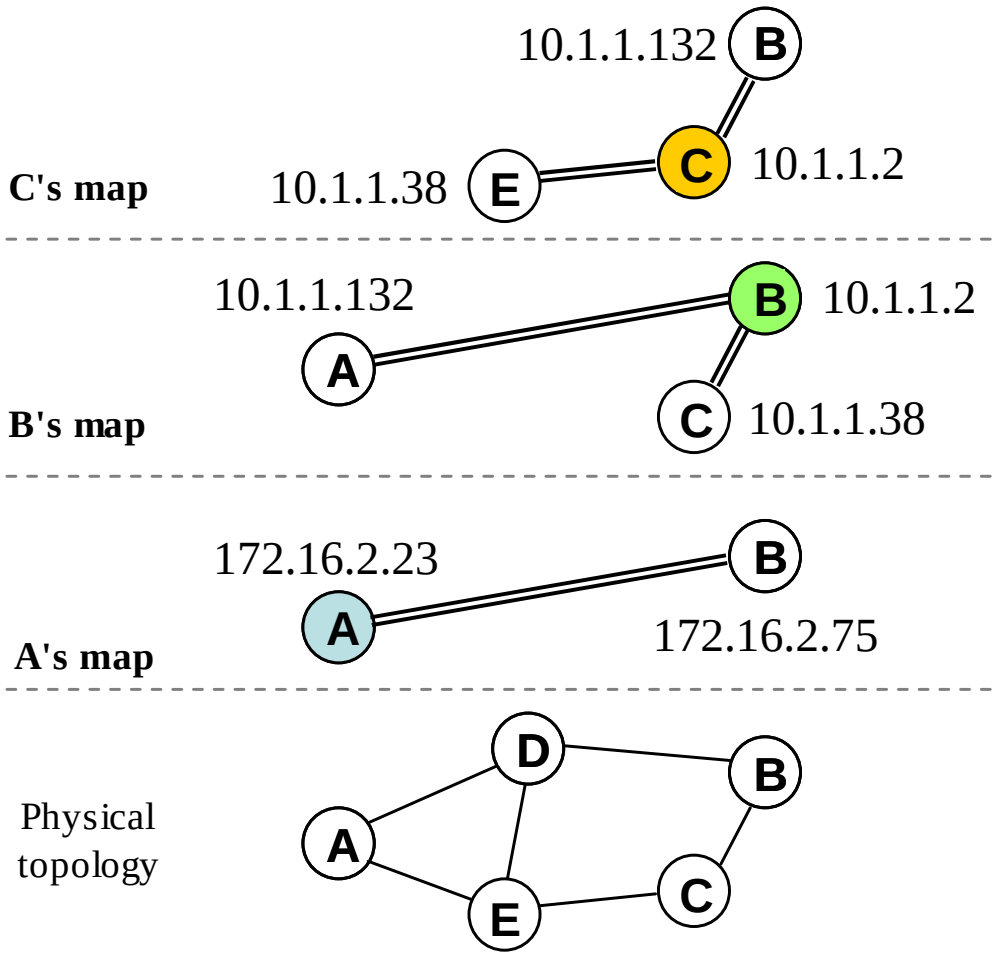


b) Route reply sent by B towards A



Private Address Maps

- ✓ No need to maintain unique addresses.
- ✗ Only suitable for flat networks with host routes.



Dynamic Shortest Discriminating Names (DSDNs)

- Each user specifies a set of ordered keywords
  - e.g. {John, Doe, UniBasel, CSDept, Switzerland}
  - and {Paul, Doe, UniBasel, BioDept}

- LUNAR dynamically creates FQDN with minimum number of keywords (first 2 are always used).
  - e.g. john-doe.net.lunar
  - and paul-doe.net.lunar

- If a name conflict is detected, names are extended
  - e.g. {John, Doe, UniBasel, CSDept, Switzerland}
  - and {John, Doe, UniBasel, BioDept}
  - → john-doe-csdept.net.lunar
    john-doe-biodept.net.lunar

DSDNs: identifying nodes

- Names can change!
  - How does LUNAR maintain paths?

- To identify nodes in the long term, each LUNAR node has a Host Identity Tag or HIT (see [HIP]).
  - Unique 128-bit string.
  - Once known, the HIT is always used to identify a LUNAR node.
  - That is, name changes do not break connections.

- Hence for each of its active peers, a LUNAR node maintains:
  - a static HIT for long term identification.
  - a dynamic name for human to network interface (i.e. yellow pages).
  - an address, to make IP stack + applications happy.

So where are we now? What's coming next?

- Complete self-configuring MANET:
  - User specifies a set of keywords without having to care about uniqueness.
  - Unique name is generated by LUNAR.
  - Any address can be used: no need to be unique.
  - Implemented as a Linux kernel module (with home-made NAT).

- Note that:
  - Private address maps can be used with any naming/routing scheme that can perform a lookup and create virtual circuits (MPLS-style).
  - I.e., DSDN + Reactive-MPLS-style routing is just one way of generating names and performing routing by name.

- Potential extensions:
  - NAT ALGs for applications using names in payload (e.g. HTML).

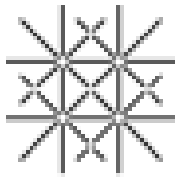
Thank you

Questions ?

Want to learn more?
… I invite you to read our CoNEXT and Wons06 papers.

Want to try?
Google {LUNARng download}

www.ana-project.org



UNIVERSITÄT BASEL