On the way to Enslave Unsociable Networks

A short defense for regular structures

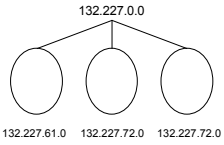
Julien Ridoux
Kavé Salamatian
Anne Fladenmuller
Yannis Viniotis

Introduction

- Next-step in Wireless: Self-Organized Networks (SON)
  - Wireless (most of the cases)
  - Multi-Hop
  - From Scratch
- Gathers different Research Problems Areas:
  - Ad-Hoc or Peer To Peer Networks
  - Routing, Addressing …
- Focus and Claim :
  - "Good" Addressing Space implies "Good" Routing Protocol

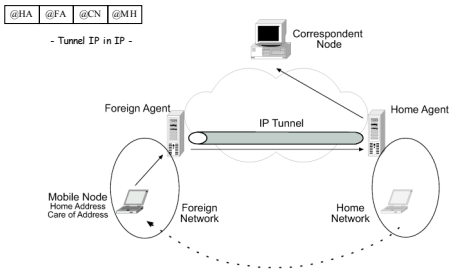
Context IP

- IP addressing : list of integers
- But theoretically organized hierarchical
  - Underlying topology built in the same way



- IP address contains :
  - Identification
  - Location
- Good for routes aggregation

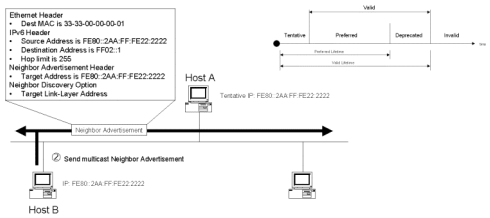
Macro-Mobilité - Mobile IP



Mobile IP – Analyse

- Mobile IP aims at maintain the association between localization and indentity.
  - Home-Address keeps on identifying the node
  - Care-of-Address ensures a coherent addressing map
- In front of mobility, Mobile IP is a hack on the network architecture
  - Tunnels = more resources needed
  - Sub-Optimal Routing

Ad Hoc Networks and IP



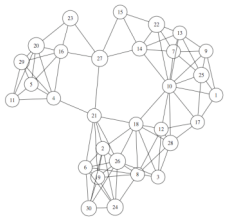
- A generalization of the DAD for wired Network
  - High Overhead
  - Not scalable
  - Not reliable (does not take into account the medium characteristics)

IP Model - conclusion

- The IP Model :
  - Is Hierarchical
  - Correlate Identity and Localization
  - Leads to non-scalable solutions
- Adaptation to Mobility :
  - Tunnels to keep the route aggregation and routing protocols with Mobile IP
  - DAD only create broadcast storms

Self-Organized Networks

- Self-Organized networks are wild !
  - No Infrastructure and Flat Topology
  - No more correlation between ID and Location
  - No particular hierarchy
  - "Classical" IP addressing organization is not compulsory



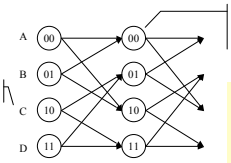
Regular Structure

- No Infrastructure : never mind !
  - Let's impose our own virtual one
  - Tree ? NO !!
- Need of a regular structure:
  - Redundant
  - Recursive
  - Robust toward mobility (if possible)
  - With equivalence between Routing and Location
    - Routes computed thanks to the structure
- A good candidate: Trellises !
  - A good tool from lower layers
  - Algebraic Structure
  - Generated by a Finite State Machine

Example 1

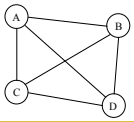
Virtual Structure

Universal ID



Node Relative Address in the Trellis

Physical Topology



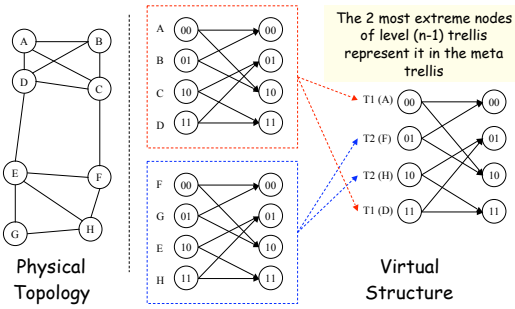
This regular structure allows redundancy :

Each node is connected to multiple receivers

The Trellis is repeated as necessary

Identification and Location are not correlated !!!

Example 2



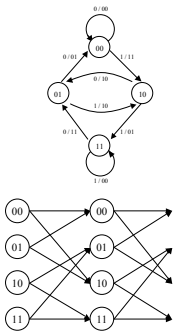
The 2 most extreme nodes of level (n-1) trellis represent it in the meta trellis

Physical Topology

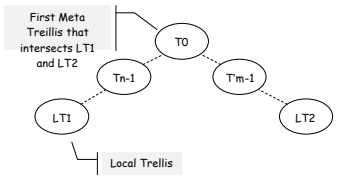
Virtual Structure

State Machine

- The structure defines the routing protocol
  - Routing with the Virtual Regular Structure
  - The state machine representation gives the equivalence between address and route
- Routing sub-optimal, but we want robustness toward mobility



Meta Trellis Architecture

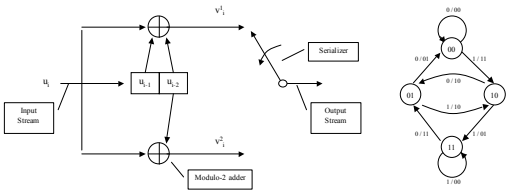


Recursive process used to bind all the local trellis of the network

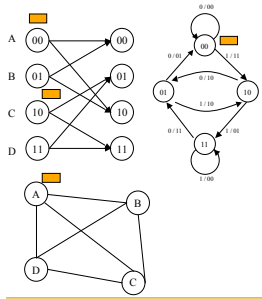
n ≠ m : not obviously a well-balanced tree

Convolutional Code

- Code : (n,k,L) = (2,1,2)
  - L shift register memory, n outputs, k entries



Routing

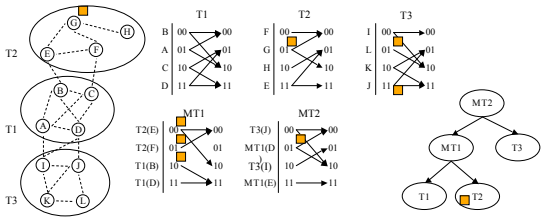


- Path : Source : A (00), Destination : B (01)
- State Machine :
  - Initial State: 00
  - Encode : 01
- The destination address can be encoded
  - Gives a route
- Routing is not optimal !!!

Recursive Routing

- We defined a recursive structure
- The same routing function can be applied at each level of recursion
- No need of a route as we define it
  - Concatenation of nodes' addresses to route through the trellis
  - Route is computed locally at each level

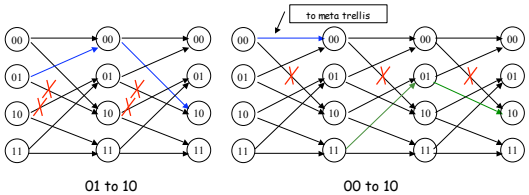
Recursive Routing



- G-> K: Suppose F knows K's relative address
  - Concatenation of addresses at each trellis level
  - Each node receiving the packet is able to route it in its trellis

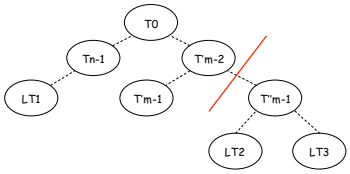
Individual Mobility

- Broken link when a node has moved – INDIVIDUAL MOBILITY
  - 1st case : we can find a longer path in the Trellis to the destination
  - 2nd case : we need to use higher level of hierarchy to reach the destination



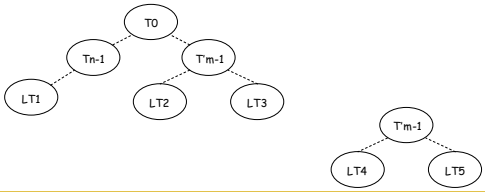
Group Mobility - Split

- A Split in the routing topology leads to a cut split at a specific recursion level
  - At worse : local trellis reconstruction
  - No change in nodes relative addresses



Group Mobility - Merge

- Recursive structure allow us to merge different independent networks
  - Add a level of recursion to bind the trellis



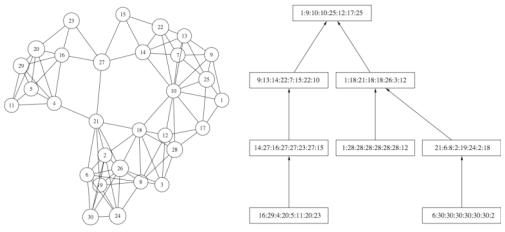
Relative Address Retrieval

- The structure allows reactive approach as well as proactive one
  - These are the 2 extreme cases :
  - Reactive :
    - Each node sends a request in the structure to retrieve the destination address
    - Load of control traffic ?
  - Proactive :
    - Each trellis head sends its neighbors information to the higher trellis
    - When reaches the highest level : go down to all nodes
    - Each node maintains information all nodes : scalability ?
- Chose both of them :
  - Proactive registration of nodes to the top of the structure
  - Reactive Route Discovery on demand

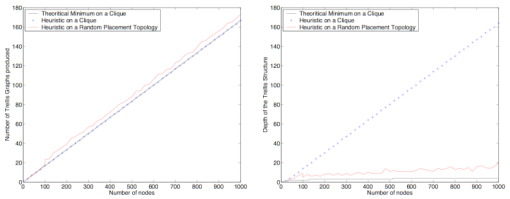
Comments

- The structure is defined locally
  - Limits the exchange of control message to the neighbors
- The structure defines the routing protocol
  - Routing without the need to exchange routing tables
  - Routing sub-optimal, but we want robustness
- Difficulty: node placement to fit at best the physical topology
  - NP-Complete problem for a given topology
  - Heuristic, based on sequential node arrival

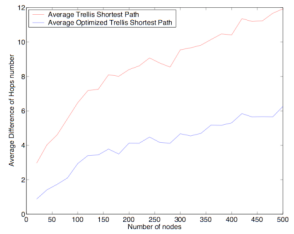
Example



Results - Construction



Results - Routing



Conclusion

- Possible to use Virtual Regular Structures
  - The tree is not compulsory !
  - Trellis are a possible implementation
- Possible to approximate the optimal VRS to any topology
- Routing becomes trivial
- The size of the addressing space fits the number of present nodes
- Local definition
- No flooding
- No exchange of routing tables
- Routing table size : constant !