GARR

The Italian Academic & Research Network

www.garr.it

University of Rome "Tor Vergata"
Department of Electronics Engineering

NETWORKING GROUP

FairVPN

Fairness-oriented Overlay VPN topology construction

Francesco Saverio Proto

Giornata di incontro con i borsisti GARR, Roma, 22.06.2010

Consortium GARR

Application Scenario

- Distributed Network of 1000+ nodes
- Nodes need to communicate securely
- PHY network is unsecure
  - Internet
  - Wireless Communities
  - Wireless spontaneous networks

Overlay VPN Network

- We run a Overlay VPN when a **group of nodes** needs to communicate on a **secure network**

- Two main problems:
  - Performances (scalability)
  - Security

Network Scenario

- VPN service connecting 100-1000 end-users (*medium-scale*).
- VPN nodes are end-user devices accessing to Internet through a private PHY connection, e.g. ADSL2+.
- VPN nodes asynchronously join and leave
- VPN links are secure tunnels based on transport (e.g. DTLS) or network-layer (e.g., IPSec) secure protocols

Overlay Topology ?

Hub-and-Spoke (star)

- a node acts as hub, other nodes (spokes) have an overlay link with the hub
- trivial to maintain but the ADSL uplink bandwidth capacity of the hub node becomes an obvious bottleneck for spoke-to-spoke connections

- Full-mesh

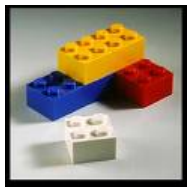
- feasible only for few tens of nodes
- because of signaling, processing and memory consumption overhead associated to the creation and maintenance of the full-mesh tunnels (maintenance of security associations, keys transfers/agreements, etc).

- (Partial) Mesh
  - a node has overlay links with a limited set of *neighbors*
  - most of the traffic relations will be routed through a multi-hop overlay path
  - need of a **routing protocol** operating on top of the overlay
  - feasible for medium-scale VPN

MESH construction

- There is a huge set of MESH topologies, thus it is challenge devising a construction strategy that singles out the optimal topology. Two approaches:

- **Clean-slate**
  - starts from given constrains and derives the optimal topology, "all-at-once"
  - integer-linear-programming
  - suitable for Virtual Service Provider network deployment, not suitable for a dynamic P2P environment, as it may imply a complete re-wiring at node joining or leaving

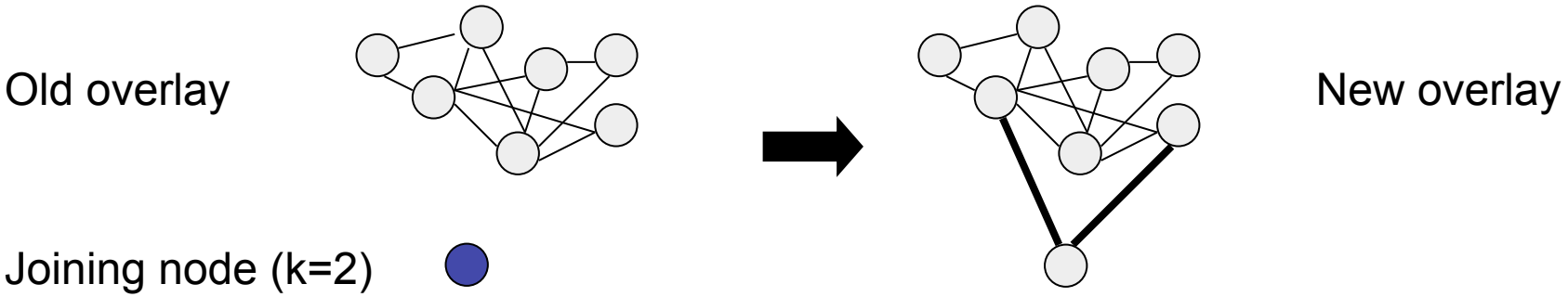


- **Incremental**
  - minimal impact on topology at node join and leave
  - overlay links established by a joining node should be retained until one of the two involved peers departs from the network
  - neighbor-selection problem: how to best set-up a given, small, number of overlay links from a just entered node toward other preexisting nodes ?

- We dealt with incremental approach

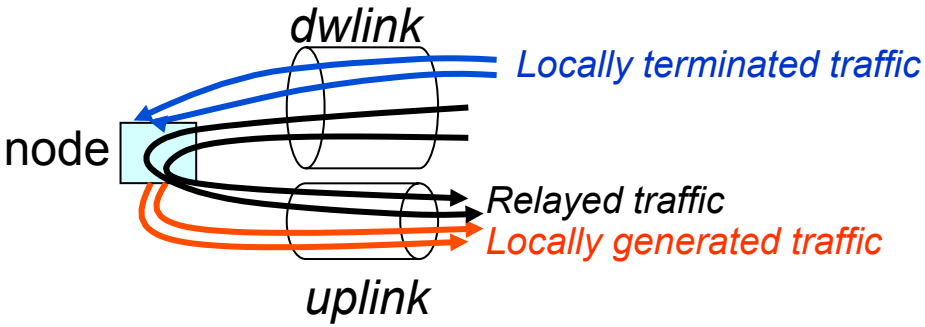
System model

- Neighbor-selection approach
  - a joining node select *k* (*fan-out*) preexisting nodes to connect to
  - when a node *X* leaves VPN, nodes that selected *X* as neighbor perform re-selection to reestablish broken links
  - selection strategy drives overlay topology toward a **specific performance goal**



Maximization of network-throughput

- **Initially** we design a neighbor-selection algorithm devised to maximize network-throughput
- Network-throughput is the sum of connections' throughput
- **The shorter the network (overlay hops), the higher the network-throughput**
- Each node is in charge to deliver (uplink) two types of traffic:
  - the locally generated traffic addressed to the remaining *N*-1 nodes
  - the traffic received by other nodes and relayed

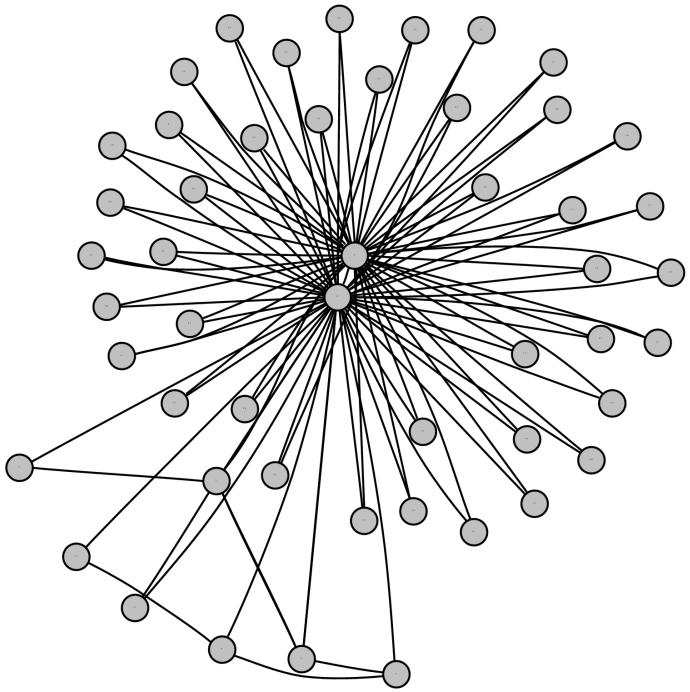


- Network-throughput = sum of locally generated traffic over all network nodes (otherwise sum of locally terminated traffic)
- The lower the relayed traffic, the higher the network-throughput

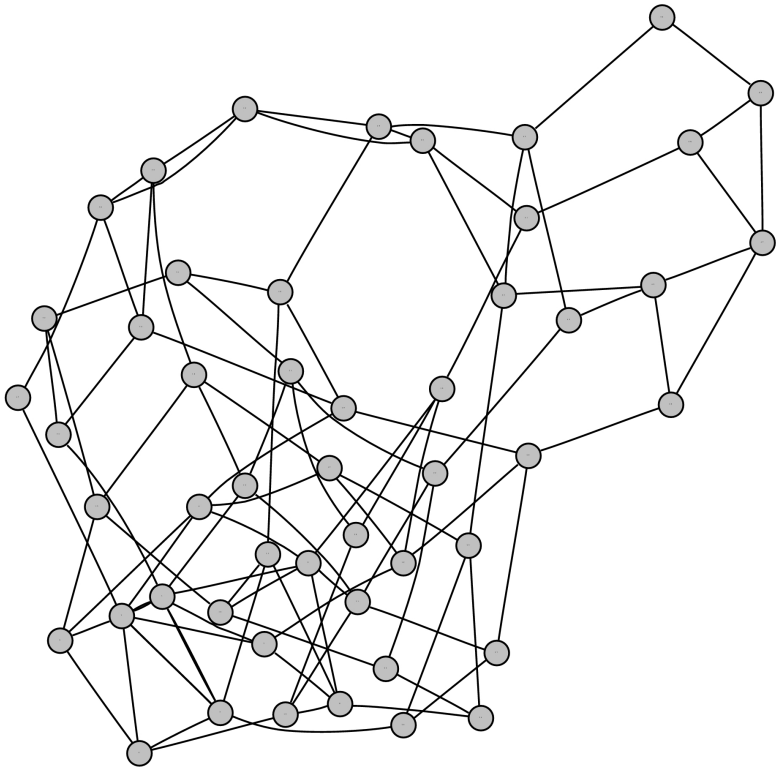
Neighbor-selection strategies

- Short-Overlay
  - A joining node *n* retrieves the current overlay topology (by a bootstrap node)
  - It derives the distance-matrix *M(i,j)* (measured in number of overlay hops)
  - It sequentially selects the best *k* neighbors by **"selfishly"** operating as follows:
    - at the *h*th step (1≤h≤k), node *n* selects as next neighbor the node that minimizes its average distance toward all the VPN nodes, also considering the previously selected *h*-1 neighbors. We recall that, in this way, the *k* neighbors are selected **one-at-a-time**.
- Short-Underlay
  - Like short-overlay, with the only difference that the distance-matrix is based on the number of underlay hops.
- Random
  - Like short-overlay, with the only difference that the *k* neighbors are randomly selected.

Visualizing preferential-attachment phenomenon



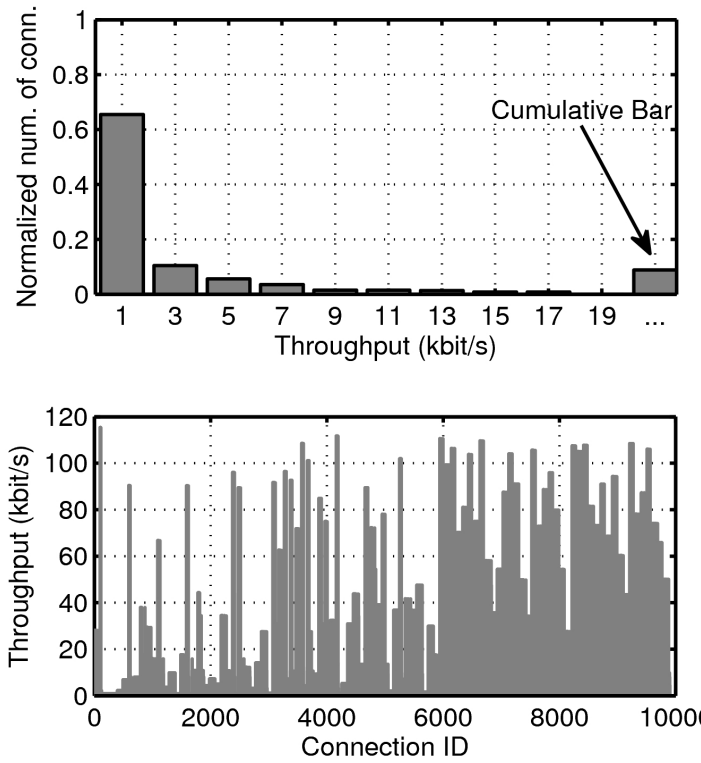
Short neighbor-selection (fan-out 2)



Fair neighbor-selection (fan-out 2)

Throughput unfairness of short overlay

- Strong unfairness exists among connection-throughputs
- 68% of traffic relations gets poor throughput (below 1 kbps), 10% exhibits a throughput higher than 20 kbps
- The unfairness is due to the *preferential-attachment* typical of incremental models for short networks
  - Probability of being selected as neighbor increases with the node degree
- "hub-and-spoke"-like topologies



Neighbor-selection for Throughput Fair overlay: insights

- Two fundamental observations:
  - An overlay topology where each uplink access channel supports the same number of connections (locally generated or relayed), would yield perfect fairness
  - The shorter the overlay, the greater the network throughput would be

- So a reasonable heuristic is:
  - selecting the set of neighbors that better equalizes the number of connections over each uplink and, simultaneously limiting the overlay average path length

Neighbor-selection for Throughput Fair overlay: the algorithm

- At the *h*th step, the joining-node *n* selects as next neighbor the node *p* that minimizes the cost function:

$$cost(h,p) = apl(h,p)\,fc(p)$$

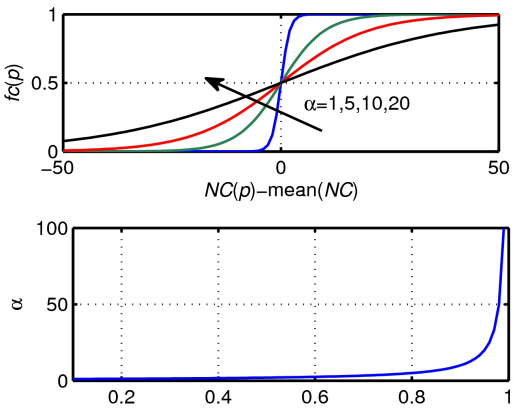
  - *apl(h,p)*: average overlay path length that the node *n* would obtain selecting the node *p*, also considering the previously selected *h-1* nodes
  - *fc(p)* (fairness-cost) is an approximation of the Heaviside step-function versus NC(*p*). The Lower the number of supported connections, the lower the fairness-cost, the higher the probability to be selected

- The "weight" of *fc(p)* in *cost(h,p)* depends on the fairness level (Jain's index $1/K \div 1$) on NCs
  - Modulation of α. The lower the current fairness level on *NCs*, the closest *fc(p)* to the ideal step function, the higher the weigth

$$fc(p) = \left(\frac{1}{1 + \exp\left(\frac{(NC(p) - \text{mean}(NC))}{\alpha} \right)} \right)$$

$$\alpha = \frac{1}{1 - \text{Jain's\_index}(NC)}$$

$$\text{Jain's\_index}(x) = \frac{\left(\sum_{k=1}^{K} x_k \right)^2}{K \sum_{k=1}^{K} x_k^{\,2}}$$

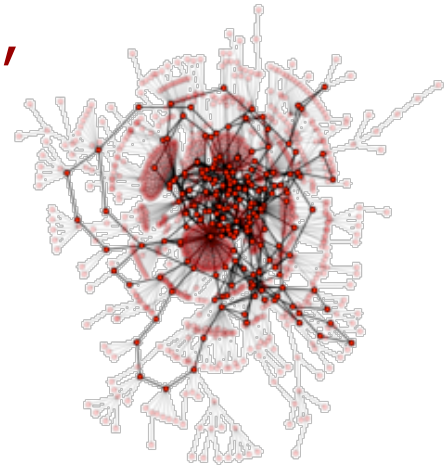


Implementation

- First implementation available
  - http://minerva.netgroup.uniroma2.it/fairvpn
  - Tested on emulated network with Netkit
    - Just ~10 nodes to test implementation
  - Testing ongoing on Planet-lab
    - Around 1000 nodes to test scalability and performance

Implementation

- Python implementation
  - Wrapper around the tinc-vpn VPN software
  - Networkx
    - NetworkX is a Python package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks.



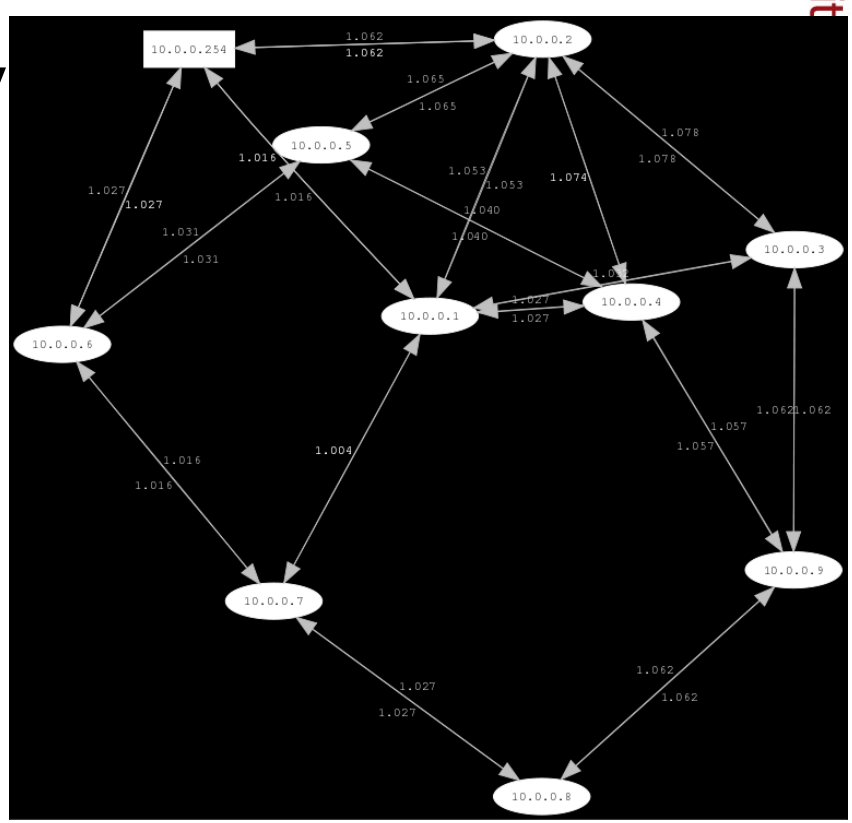
Bootstrap node

- When a node wants to attach to the VPN the bootstrap nodes provides the current topology

  - ANY Node can act as bootstrap node at any given time
  - The OLSR dot-draw plugin exports the overlay network topology
  - Mapping between underlay and overlay addresses are exported with the name-service plugin

OLSR Routing protocol

- We run OLSR on the Overlay Network
  - Link-state protocol
    - The topology is used by next joining node to run the algorithm



Validation with Netkit

- We prooved that the implementation for the costruction of the topology works
- We cannot do performance measurements in UML (emulator is not a simulator) to test that the Overlay VPN is fair in terms of throughput

Planet Lab

- PlanetLab is a group of computers available as a testbed for computer networking and distributed systems research.

- We are deploying the FairVPN implementation on a Planet-Lab slice

Future work

- Performance measurements on planet-lab
- Introduce reputation mechanism to:
  - Change select fanout nodes
  - Trigger rewiring
  - Change routing behaviour in the overlay

Questions ?

- Questions ?