

# Fairness-oriented Overlay VPN topology construction

A. Detti, A. Caricato, G. Bianchi

Electronic Engineering Dept., University of Rome Tor Vergata, Italy  
{andrea.detti, alessandro.caricato, giuseppe.bianchi}@uniroma2.it

*Abstract* — An important issue in dynamically constructed Virtual Private Networks (VPN) is how the overlay topology is created and maintained. Classical VPN topologies, such as hub-and-spoke or full-mesh, fail to remain convenient and viable when the number of nodes grows to as little as a few tens. Convenient topology formation mechanisms should be distributed, should permit incremental and dynamic operations, and should limit the number of nodes a new entry connects with. In this work, we show that approaches devised to create “short” networks, while yielding a significant total network throughput, may be severely affected by unfairness issues, i.e., different pair of nodes may experience a widely different throughput performance. Hence, we introduce a fairness-oriented topology formation algorithm for VPN. The proposed algorithm is incremental, meaning that the addition of a new node to the overlay topology does not imply rewiring of already established overlay links. Simulation results show that our proposed approach achieves high fairness levels, as quantified in terms of well known Jain’s fairness index, meanwhile retaining satisfactory throughput performance.

## I. INTRODUCTION

An overlay Virtual Private Network (VPN) is the cheaper mean to implement a secure Intranet on top of a public Internet infrastructure. The availability of public domain VPN software running on end-user Linux, MAC OS X, or Windows computers [1], and the adoption of transport (TLS/DTLS) or application (SSH) layer secure tunneling protocols have boosted the deployment of the VPN paradigm through a wider range of scenarios, currently even including spontaneous end user communities.

The scenario targeted by this paper is that of a VPN formed by end-user devices accessing the Internet through a broadband asymmetric connection, like ADSL2+ [7]. VPN nodes asynchronously join and leave the network during its lifetime, and the number of expected VPN participants is in the medium scale order of a few tens to several hundreds of nodes. Networks of such type and scale are hardly compatible with the two topologies most frequently found in VPNs, namely, *hub-and-spoke* and *full-mesh*.

In a hub-and-spoke topology, a specific node acts as hub, whereas all the remaining nodes (spokes) are connected only to the chosen hub. Albeit trivial to maintain and dynamically update, this VPN topology is performance effective only in the very special case of traffic relations mostly addressed from/to the hub node itself. In the more general case of randomly distributed traffic (i.e. when spoke-to-spoke connections are the norm rather than the exception), the uplink bandwidth capacity of the hub node becomes an obvious bottleneck. On the opposite side, full-mesh topologies envision a dedicated

overlay link for every VPN node pair. The price to pay is the signaling, processing and memory consumption overhead associated to the creation and maintenance of the full-mesh, a burden which basically rules out this topology for networks over just a few tens of nodes. Indeed, each VPN link has to be protected through a secure tunneling protocol, and this involves state maintenance for each security association, computationally heavy VPN join procedures (every new joining node must deploy as many asymmetric cryptographic handshakes - RSA-based key transfers or Diffie-Hellman-based key agreements - as the number of remaining network nodes), rekeying overhead, etc [2].

This paper addresses the issue of creating and maintaining suitable VPN topologies [3] where a node has an overlay link with a (possibly small) subset of other nodes [4][5]. The lack of a full-mesh availability implies that most of the traffic relations will not find a direct overlay link available, and will be routed through a multi-hop overlay path between source and destination (e.g. the shortest path available). The deployed overlay topology has a strong impact on traffic performance, and the choice of a performance effective overlay topology is a complex problem, especially when, as in the dynamic scenarios targeted by our work which envision nodes asynchronously joining or leaving, supplementary *practical* conditions are posed on the process of topology creation and maintenance.

Specifically, our goal is to devise “*incremental*” connectivity management mechanisms (unlike clean-slate approaches based on integer linear programming, e.g. [14] [16]), characterized by the property that the overlay links established by a joining node should be retained throughout the network lifetime until one of the two involved peers departs from the network. Permitting “re-wiring” of the overlay topology would in fact be detrimental for several practical reasons, including overlay routing protocol instability, supplementary processing burden in re-establishing security association, and so on. The envisioned incremental operation hence resorts to a problem of *neighbor-selection*, i.e. how to best set-up a given, small, number of overlay links from a just entered node toward other preexisting nodes.

### A. Our contribution

When we first started tackling this issue, our natural goal was to design a neighbor-selection algorithm devised to maximize the network throughput. As discussed in what follows, the maximization of the network throughput involves the minimization of the overlay average path length. However, we observed that a short network was achieved only at the price of a severe unfairness in the throughput achieved between different pairs of nodes (see numerical results presented and discussed in section III).

Motivated by such a fairness issue, in this paper we devise a neighbor-selection strategy pursuing a fair and efficient VPN topology. Perfect fairness is achieved when, in case of uniform traffic, all end-to-end connections experience the same throughput. Perfect efficiency is achieved when the total network throughput (i.e. the sum of throughputs of all connections) is the highest possible. Our proposed incremental approach accomplishes a very high level of fairness (quantified in terms of the topmost known Jain's index) meanwhile retaining an effective total network throughput. To the best of our knowledge, ours is the first work which explicitly brings fairness requirements into the design of an incremental overlay connectivity management mechanism.

The remainder of this paper is organized as follows. System and simulation models are described in Section II. Section III debates the unfairness issue consequent to hop-length minimization mechanisms. Section IV describes the proposed fairness-oriented topology creation and maintenance approach. Its performance are assessed in section V. Finally, a brief summary of related works and conclusions are drawn.

## II. SYSTEM MODEL

### A. Underlay Topology and Traffic assumptions

We consider a randomly generated underlay network topology composed of Internet routers and hosts. First, we generate the topology among routers through the GT-ITM [15] package, based on the Transit-Stub model [8] (see Figure 1). Then, we randomly connect hosts to stub routers, and we pick a subset of hosts as those composing the VPN<sup>1</sup>. We assume over-provisioned, large capacity, backbone links [6], whereas we assume that the capacity bottleneck is represented by access links. Following ADSL2+ typical values [7], we set the uplink access capacity of each node to 1 Mbit/sec, and the downlink access capacity to 24 Mbit/sec. Although we assume a homogeneous access, the conceptual findings drawn in this paper are effective even for heterogeneous accesses. Moreover, we will preliminary suggest in what extend the proposed algorithm to cope heterogeneity (see next footnote 6).

Traffic relations are assumed to be uniformly distributed and greedy across the whole VPN, meaning that every node in the VPN attempts to transmit the greatest possible amount of traffic to each of the remaining network nodes. We also assume that all connections crossing a same access link fairly share the available access bandwidth (as it would occur if they were regulated by TCP). Consequently, we evaluate the throughput of connections through a max-min fair rate allocation algorithm, only accounting of access bandwidth constraints [8].

In the presence of a full-mesh topology (best case), these assumption would imply that the throughput between any pair of nodes would be  $1/(N-1)$  [Mbit/s], with  $N$  being the number of VPN nodes, and that the total network-throughput would be  $N$  [Mbit/sec], since every nodes would generate the traffic amount needed to saturate its uplink capacity.

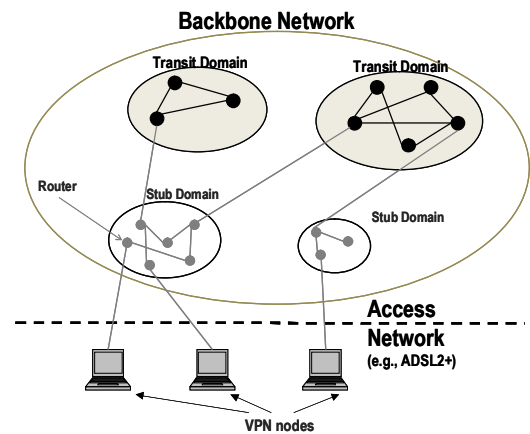


Figure 1 - Underlay network topology

### B. Routing assumptions

As anticipated above, any non-full-mesh topology in general requires traffic to be routed through a multi-hop overlay path between source and destination. In practice, this is accomplished by running a routing protocol on the overlay. We assume a link-state routing protocol such as OSPF or OLSR to be employed; as a result, traffic between two generic VPN nodes is routed through the shortest available path. In the results presented in what follows, we consider two possible path cost metrics:

- **overlay**: the path cost is measured in terms of number of overlay hops.
- **underlay**: the path cost is measured in terms of number of underlay (physical network) hops;

In both cases, the cost of a hop (overlay or underlay, depending on the assumed metric) is assumed to be unitary, plus a very small random number randomly drawn once the link is setup. This technical arrangement ensures, in both cases, that every pair of nodes will have only a single shortest path available, and hence allows our results to remain free of “implicit” load balancing issues which would emerge with the occurrence of equal cost paths.

### C. Neighbor Selection and Overlay Evolution assumptions

The overlay is incrementally constructed as follows [4]. We start from an empty VPN and we incrementally add nodes, in random order, to it. At each node addition (join), a neighbor-selection procedure is performed, where the joining node connects to  $k$  preexisting nodes. Obviously the choice of which specific  $k$  nodes to connect to is characteristic of a given approach, and for this reason it will be described next, when presenting and comparing the considered strategies. The procedure stops when the network contains a given number of nodes.

In the proposed performance evaluation we also emulate and assess a dynamic environment. Time-varying network conditions are modeled as follows. Once the network is built, we proceed by performing ten leave/rejoin rounds: in each round, five randomly selected nodes are first removed from the network, and then these same nodes are re-joined in a different

<sup>1</sup> We spread 1500 routers on stubs and transit domains as follow: 5 transit domains has on average 6 routers; for each transit router, 7 stub domains has on average 7 routers. Connection probability is between transit routers is 0.7, while the connection probability among routers of the same stub is 0.55.

random sequence. While, in principle, a “pure” incremental operation might not involve any action when a node leaves the topology, such a lack of operation may eventually yield network disconnection. For this reason, consistent with other literature work [4], we assume that, when a node  $X$  leaves the network, any other node that, while joining, had originally selected  $X$  as its neighbor, is expected to re-establish the broken link by selecting an alternative neighbor (i.e., neighbor-selection only for last link). Note that this procedure is not performed by nodes which were selected by  $X$ .

### III. UNFAIRNESS OF “SHORT NETWORK” NEIGHBOR SELECTION STRATEGIES

In a multi-hop overlay composed of  $N$  nodes, each node is in charge to deliver two types of traffic: i) the traffic locally generated and addressed to the remaining  $N-1$  nodes, and ii) the traffic received by other nodes and relayed. The total network-throughput can be measured as the sum of only the first type of traffic (locally generated) over all network nodes. It is therefore obvious that a strategy that attempts to reduce as much as possible the amount of relayed traffic will in turns achieve a high network-throughput. Intuitively, this can be accomplished by keeping the overlay network as short as possible in terms of number of overlay hops (i.e. minimize the number of relay nodes). Clearly, this approach is effective when underlay network paths are over-provisioned, like we assume (see Section II.A). To verify this idea, we compare three different incremental neighbor-selection approaches.

**Short-Overlay** (briefly named *short*). According to this strategy, a node  $n$  that aims to join the VPN retrieves information about the current overlay topology<sup>2</sup> and derives the distance-matrix  $M(i,j)$ , measured in number of overlay hops, between any node  $i$  and node  $j$ . After that, the joining node *sequentially* selects the best  $k$  neighbors by “selfishly” operating as follows [5]. At the  $h$ th step ( $1 \leq h \leq k$ ) of the sequence, the node  $n$  selects as next neighbor the VPN node that, after adding the corresponding overlay link to the topology, minimizes *its* average distance toward *all* the VPN nodes, also considering the previously selected  $h-1$  neighbors. The average path length  $apl(h,p)$  that, at the  $h$ th step, the node  $n$  would obtain selecting the node  $p$  as next neighbor can be efficiently evaluated by applying minimum expectation operations<sup>3</sup> upon the matrix  $M$ . We recall that the above suggested sequential approach is sub-optimal with respect to the selfish approach proposed in [4] which selects the  $k$  overlay neighbors all-at-once and not one-at-a-time, like we do. However, our goal is here limited to show that such type of distance-based selection strategy is affected by fairness issues, and for this reason the above outlined one-at-a-time approach allows us to readily raise this issue.

<sup>2</sup> This is in practice accomplished by having the joining node contacting the VPN through a bootstrap node. The bootstrap node forwards to the joining-node the link-state messages of the routing protocol; hence, after some time, the joining node retrieves the overlay topology [4].

<sup>3</sup> For instance, if the node  $n$  has already selected the neighbor  $x$ , then the current average path length toward all the remaining node is the average of the  $x$ th row of  $M$  (i.e.,  $\text{mean}(M(x,:))$ ) plus 1; the value 1 accounts for the overlay distance between neighbors  $n$  and  $x$ . Now, if  $n$  selected as next neighbor node  $y$ , then the average path length would become  $\text{mean}(\min(M(x,:), M(y,:)))+1$ .

	Average overlay path length	Network-throughput (Mbit/s)
Short	1.85	76
Short-Underlay	1.9	71
Random	1.95	56

TABLE I - RESULTS WITH  $N=100$  NODES,  $K=8$

**Short-Underlay.** This procedure is executed as above, with the only difference that the distance-matrix is based on the number of underlay hops.

**Random** (briefly named *rand*). This procedure is executed as *short* one, with the only difference that the  $k$  neighbors are randomly selected.

TABLE I reports results obtained for a network of  $N=100$  nodes, and using  $k=8$  as “fan-out” parameter, i.e. number of neighbors each joining node connects to. Two performance metrics are shown: average overlay path length, measured in average number of overlay hops, and network throughput, measured in Mbit/sec (we recall that 100 Mbit/sec would be the best possible throughput performance achievable by a full-mesh topology). As expected, the *short* algorithm is the best performing strategy, as it minimizes by construction the parameter (the overlay distance) which directly affects the network-throughput.

#### A. The emergence of severe unfairness issues

While the above results were expected, the quantitative findings discussed hereafter are less intuitive and, we believe, particularly critical. Figure 2 analyzes the throughput obtained by single connections of the VPN (connection-throughput) in case of *short* neighbor-selection. The values of connection-throughputs are depicted in the lower plot of Figure 2 (the connection ID reported in the x-axis being conveniently summarized as  $ID=100i+j$  to indicate the throughput experienced by the traffic generated by node  $i$  and addressed towards node  $j$ ). From the figure, it appears that a *strong unfairness exists among connection-throughputs*. This is quantitatively confirmed by the normalized histogram reported in the upper plot of Figure 2. We note that as much as 68% of the traffic relations obtains a poor throughput (below 1 kbit/sec), whereas less than 10% of the traffic relations exhibits a throughput higher than 20 kbit/sec (with peaks up to more than 100 kbit/sec) and as such gather most of the total network-throughput (i.e. 76 Mbit/s).

The motivation of such an unfairness is that a common behavior of all incremental models aiming to generate short networks is *preferential-attachment*: the probability of receiving new edges increases with the node’s degree (i.e., number of overlay links incident on the node) [9][10]. Preferential-attachment provides “hub-and-spoke”-like topologies, where a wide set of nodes (spokes) select their neighbors among a limited set of nodes (the hubs). In such a topology, on the one hand a generic overlay path likely has a length lower than two hops (i.e., a valuable performance) but, on the other hand, the uplink capacity of the hubs becomes a critical bottleneck for all the node-to-node relations which use the so generated hubs as relays. Only the few node-to-hub connections achieve high throughput, as the hubs’ downlink capacity is not throttled.

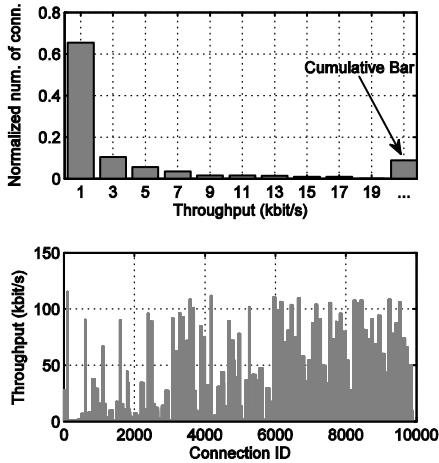


Figure 2 - Histogram (upper) and absolute values (lower) of connection-throughput in case of 100 nodes, fan-out 8, *short* neighbor selection strategy.

Figure 3 plots the number of links incident of every node (i.e., the node *degree*) as resulting from the execution of the *short* neighbor-selection strategy. This figure clearly raises the occurrence of preferential-attachment. Indeed, few nodes have a very high degree (i.e., they act as hubs), while the most of nodes has a degree in the order of the fan-out (i.e., they act as spokes). In addition, while Figure 2 has shown up unfairness among connection-throughputs, Figure 3 reveals another form of unfairness: *management-load unfairness*. A limited set of nodes, indeed, is heavily loaded by the burden of managing a high number of overlay links, while the majority of nodes only manages a small number of overlay links.

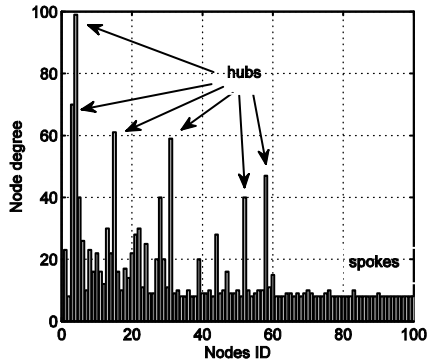


Figure 3 - Node degree in case of 100 nodes, fan-out 8, *short* neighbor selection strategy

Finally, for ease of illustration, in the left graph depicted in Figure 4 we visualize the topology obtained by the *short* algorithm in case of a network with 50 nodes and fan-out 2. We clearly observe two hub nodes at the center of the graph. On the contrary, the right graph in Figure 4 visualizes the topology achieved by our *fair* algorithm (discussed later); we observe a “fair” role of nodes.

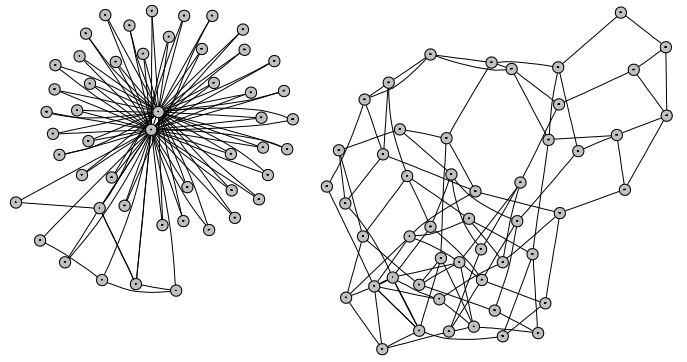


Figure 4 - VPN topology in case of *short* algorithm (left) and *fair* algorithm (right), fan-out 2, 50 nodes

#### IV. NEIGHBOR SELECTION METHODOLOGY FOR FAIR OVERLAY

In this section we introduce a neighbor-selection algorithm devised to incrementally construct an overlay network which exhibits a high level of fairness in terms of connection-throughput<sup>4</sup>, meanwhile retaining a satisfactory total network throughput performance.

##### A. Motivation and insights

Our proposed approach stems from the following two remarks.

**Observation 1:** *An overlay topology where each uplink access channel supports the same number of connections (locally generated or forwarded), would yield perfect fairness.*

**Observation 2:** *The shorter the overlay, the greater the network throughput would be.*

An overlay path can be seen as a succession of bottlenecks, namely the traversed uplinks. The connection-throughput is equal to the amount of bit-rate obtained in the worst case bottleneck. Therefore, if for all connections the amounts of bit-rate obtained in the worst case bottleneck were the same, then each connection would achieve the same throughput. In our model, each uplink has the same capacity, fairly shared among supported connections (locally generated and relayed); hence, the worst case bottleneck of a connection is the uplink access channel supporting the largest number of connections. Consequently, if all uplinks were supporting the same number of connections, then the bit-rates granted to the connections would be equal (observation 1).

Let us now assume to have found a perfectly-fair overlay network, formed by  $N$  nodes and with an average path length equal to  $H$ . In this case, we are able to prove the second observation. The total number of connections supported by all uplinks is  $N(N-1)H$  and each uplink exactly conveys  $(N-1)H$  connections. By assuming an unitary uplink capacity, the amount of bit-rate obtained by a connection on each crossed uplink is  $1/((N-1)H)$ , the end-to-end throughput of each

<sup>4</sup> In this paper, we directly focus on connection-throughput fairness; nevertheless, performance evaluation will show that we also achieve a high fairness in terms of management-load, i.e. node degrees are similar.

connection is  $1/((N-1)H)$ , too, and the overall network-throughput is  $N/H$ <sup>5</sup>. Therefore, the lower the average path length  $H$ , the higher the network-throughput (observation 2).

Motivated by these two observations, we infer that reasonable heuristic for neighbor-selection algorithm consists in *i) selecting the set of neighbors that better equalizes the number of connections over each uplink, and ii) simultaneously limits the average overlay path length.*

Regarding the first goal, we observe that we have a single connection for each couple of nodes  $(i,j)$  and this connection goes on the unique shortest-path  $(i,j)$  (see Section II.A). Therefore, the number of connections relayed by the generic  $p$ th uplink is equal to the number of shortest-paths traversing the  $p$ th node, i.e. the *betweenness-centrality*  $BC(p)$  [11][13]. Moreover, the total number of connections  $NC(p)$  supported by the  $p$ th uplink is equal to  $BC(p)+(N-1)$ , where  $N-1$  is the number of locally generated connections. When a joining node selects as neighbor the  $p$ th node, then the  $p$ th node is used by the joining node to reach (and to be reached by) other VPN nodes; consequently,  $BC(p)$  increases and the number of connections supported by the  $p$ th uplink increases too. Therefore, we conclude that a straightforward way to equalize the number of connections supported by uplinks consists in *favoring the selection of nodes with lower number of connections.*

### B. The proposed “fair” neighbor selection algorithm

When a node joins the VPN, it retrieves the current overlay topology, then derives the distance-matrix  $M$  (based on overlay hops) and corresponding betweenness-centrality [11]  $BC$  of each node. Subsequently, the joining node iteratively selects  $k$  neighbors, one-at-a-time. At the  $h$ th step,  $(1 \leq h \leq k)$ , the joining-node  $n$  selects as neighbor the node  $p$  that minimizes the following *heuristic* cost function:

$$\text{cost}(h, p) = \text{apl}(h, p) \text{fc}(p) \quad (1)$$

Where the value  $\text{apl}(h, p)$  is the average path length that, at the  $h$ th step, the node  $n$  would obtain selecting the node  $p$  as next neighbor, also considering the previously selected  $h-1$  neighbors. This component of the cost function drives the neighbor-selection toward the achievement of a short overlay network.

The value of  $\text{fc}(p)$  (fairness-cost) is reported in Eq (2), where  $N$  is the number of nodes that are already in the VPN. This component of the cost function drives the neighbor-selection toward the achievement of a fair overlay, i.e. a network formed by nodes supporting a similar number of connections. It is a function inversely proportional to the number of connections supported by the  $p$ th node: a lower

<sup>5</sup> We stress that the value of  $N/H$  is strictly valid in case of a perfect-fair overlay, where each uplink exactly supports  $(N-1)H$  connections. Moreover, we infer that in case of an unfair overlay, the value of  $N/H$  can be considered as a lower-bound. This happens because in an unfair overlay, connections with longer path are penalized. Since for a given amount of connection-throughput, a connection with a longer path consumes more uplink resources (it crosses more uplink accesses), then by penalizing longer connections we are using more efficiently uplink resources and, in turn, we obtain a greater network-throughput.

number of supported connections involves a lower value of  $\text{fc}(p)$ ; i.e., a greater possibility of getting the minimum value of  $\text{cost}(h, p)$  and of being selected as neighbor. The function  $\text{fc}(p)$  is a well-known approximation of the Heaviside step-function (Figure 5 upper) versus the number of connections  $NC(p)$ , centered on the average value of  $NC$  ( $\text{mean}(NC)$ ). The design of  $\text{fc}(p)$  enforces  $NC$  values to being very concentrated around their average value; i.e. very similar  $NC$  values<sup>6</sup>.

$$\text{fc}(p) = \left( \frac{1}{1 + \exp\left(-\frac{(NC(p) - \text{mean}(NC))}{\alpha}\right)} \right) \quad (2)$$

$$NC(p) = BC(p) + (N - 1)$$

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

For the cost function  $\text{cost}(h, p)$ , the “weight” of the fairness-cost ( $\text{fc}$ ) component, respect to the weight of average path length ( $\text{apl}$ ) component, depends on parameter  $\alpha$ . By decreasing  $\alpha$ , the function  $\text{fc}$  tends to be more and more close to the step-function, and this gets stronger the fairness achievement, respect to the reduction of average path length. Conversely, by increasing  $\alpha$  the function  $\text{fc}$  tends to be a constant, so that fairness achievement is relaxed. We dynamically adapt the parameter  $\alpha$  to the level of fairness provided by the overlay at the node joining. In case of an overlay with an high level of fairness  $\alpha$  is high, and vice-versa. (Figure 5 lower). We measure the level of fairness through the Jain’s index [12] of  $NC$  values. Jain’s index is equal to 1 when all considered values are equal and the more we move away from this perfect-fair condition, the more the index tends to the minimum value, i.e. the inverse of the number of elements.

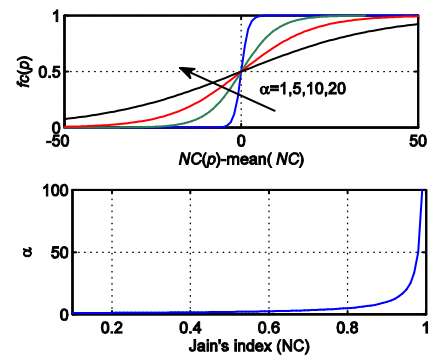


Figure 5 - Fairness-cost (upper), parameter  $\alpha$  (lower)

<sup>6</sup> In case of heterogeneous uplink capacities, if the  $p$ th node has an uplink capacity equal to  $u(p)$  units of bandwidth (e.g., 256 kbit/s being a bandwidth-unit), then in  $\text{fc}(p)$  and  $\alpha$  of Eq (2) the value  $NC$  has to be replaced by its *normalized* version,  $NC(p)/u(p)$ .



## V. PERFORMANCE EVALUATION

We have carried out two campaigns of performance evaluation. In the first campaign we examine performance versus the number of VPN nodes in case of fan-out 8. In the second campaign we evaluate the impact of fan-out on a VPN formed by 100 nodes. Although not reported, at each node joining we monitored the evolution of  $\alpha$  parameter (Eq. (2)) and, for all the simulations hereafter reported, we observed a convergence trend, evidencing a stable behavior of the *fair* neighbor-selection algorithm.

### A. Analysis versus the number of nodes.

Figure 6 reports the normalized histogram (upper) and absolute value (lower) of the throughput obtained by the different connections, in case of 100 nodes, fan-out 8 and *fair* neighbor-selection algorithm. The majority of connections gets about 5 kbit/s of throughput (we remark that 10 kbit/s would be the expected throughput in a full-mesh network), and the remaining 10% of connections receives a throughput slightly greater (and in no case greater than 13 kbit/s). By comparing the *fair* algorithm with the *short* one, we also notice a significant throughput improvement for most of the connections (while, with *short*, 50% of connections do not overcome 0.9 kbit/s, for a fair overlay we get 5.2 kbit/s).

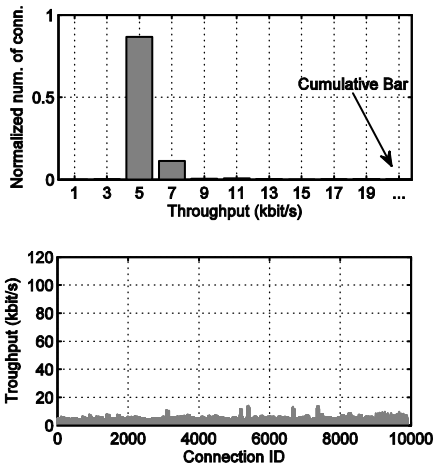


Figure 6 - Histogram (upper) and absolute values (lower) of connection-throughput in case of 100 nodes, fan-out 8, *fair* algorithm

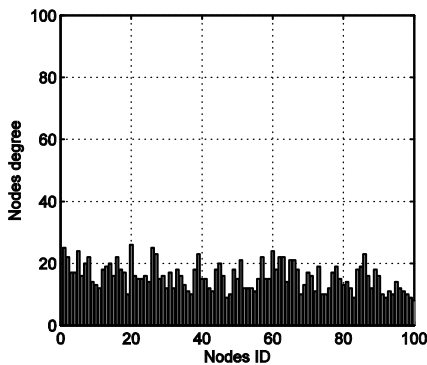


Figure 7 - Node degree in case of 100 nodes, fan-out = 8, *fair* algorithm

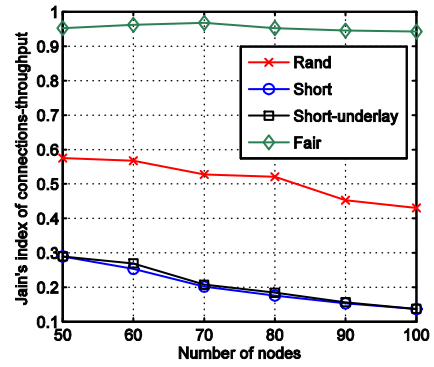


Figure 8 - Jain's index of connection-throughput versus number of nodes with fan-out 8

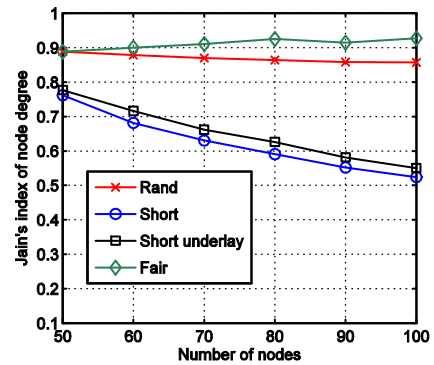


Figure 9 - Jain's index of node-degree versus number of VPN nodes with fan-out 8

Figure 7 shows that the values of node degree (i.e., number of overlay links per node) are very similar. This result reveals that an indirect-effect coming out from the equalization of connection-throughput is the equalization of node degree; i.e., valuable management-load fairness.

Figure 8 reports the Jain's index of connection-throughput. We observe that short networks (overlay or underlay) provide values of the Jain's index close to the lower bound,  $1/N$ ; therefore, there is a strong unfairness, which gets worse as the number of nodes grows. The network based on random neighbor-selection does not suffer of preferential-attachment phenomenon and, hence, tends to be more fair than short networks. *Fair* neighbor-selection succeeds in enforcing connection-throughput fairness (Jain's index being close to 1).

Figure 9 reports the Jain's index computed for the node degree. We observe that short networks (overlay or underlay) provide the lowest performance, as network links are concentrated on a small set of preferential nodes. In case of random neighbor-selection, there are no preferential nodes and the node degree tends to be similar. *Fair* neighbor-selection provides a slightly higher performance than *random* one.

Figure 10 reports the overlay average path length. The *short* algorithm yields the lower overlay average path length, while the *random* algorithm yields the worst one. The *short-underlay* algorithm yields a performance between *rand* and *short* algorithms; indeed, a longer underlay path, likely (but not

surely) corresponds to a longer path and the minimization of underlay path length is close to the overlay path length minimization<sup>7</sup>. For small number of nodes, *fair* has a path length comparable to that of *short*; nevertheless, by increasing the number of nodes, *fair* tends to get closer to *rand*'s performance. This is motivated because, as the number of nodes increase, it becomes more difficult to maintain fairness; consequently, the *fair* algorithm tends to enforce fairness-cost component of Eq. (1) (i.e., tends to use lower values of  $\alpha$ ), with respect to the average path length component.

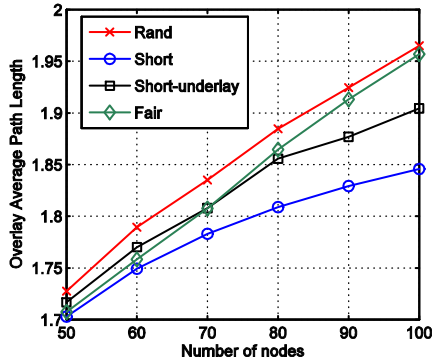


Figure 10 - Overlay average path length versus number of nodes, fan-out 8

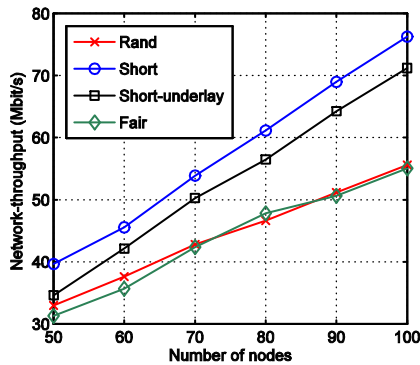


Figure 11 - Network-throughput (Mbit/s) versus no. of nodes with fan-out 8

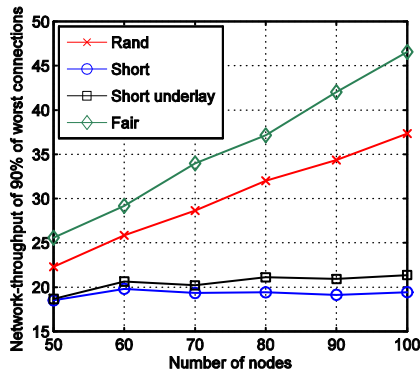


Figure 12 - Amount of network-throughput (Mbit/s) provided by the 90% of worst connections versus number of nodes, fan-out 8

<sup>7</sup> Although not reported, we also measured the average underlay path length. The short-underlay algorithm yields the shortest underlay network, random yields the worst performance and short is in the middle.

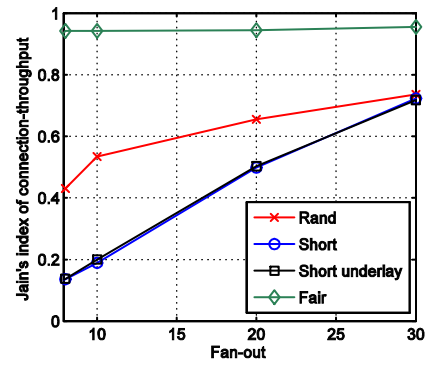


Figure 13 - Jain's index of connection-throughput versus number of fan-out in case of 100 nodes

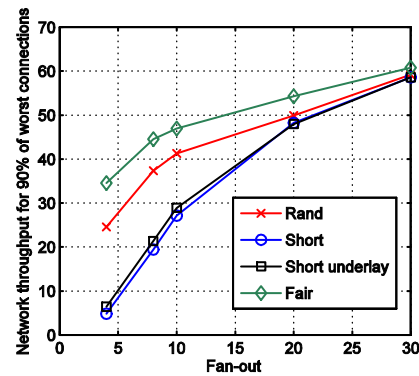


Figure 14 - Amount of network-throughput (Mbit/s) provided by the 90% of worst connections versus fan-out, in case of 100 nodes.

It could be argued that the proposed *fair* algorithm yields a lower network-throughput than the *short* approach which, by construction, attempts to maximize the network-throughput (i.e., sum of connection-throughputs). This is indeed expected, and confirmed by the results presented in Figure 11, which show that the network-throughput of *fair* is comparable to the *random* neighbor selection approach. However, it is worth to remark that the overall network-throughput is a single performance metric which somewhat “hides” the actual throughput performance perceived by the majority of connections.

This is made evident by the results presented in Figure 12, which reports the amount of network-throughput exclusively provided by the 90% of worst connections (named 90-network-throughput)<sup>8</sup>. Not only we observe that *fair* neighbor-selection yields the highest 90-network-throughput, while *short* yields the lowest one, but also we remark the significant absolute difference emerging when the network size grows.

These results are readily explained as follows. In the case of *fair*, most of the connections receives the same throughput, and therefore the 90-network-throughput is quite close with the 90% of the overall network-throughput (Figure 11). Conversely, in the case of *short*, the network-throughput is

<sup>8</sup> To evaluate the 90-network-throughput we sort in ascending order the values of connection-throughputs, then sum the first 90% of elements. Obviously, the 100-network-throughput is equal to the overall network-throughput.

“boosted” by very few connections, whereas the majority of connections (spoke-to-spoke) retains a marginal throughput performance. For this reason, the sum of the throughput contributed by as much as 90% of the worst case connections remains very low.

### B. Analysis versus fan-out.

For completeness, in this section we briefly discuss the performance results obtained by considering different fan-out values. Obviously, the higher the fan-out, the shorter the average path length. Moreover, an higher fan-out naturally provides more fairness; indeed, full-mesh is a perfect fair topology. Figure 13 reports the Jain’s index of connection-throughput in case of 100 nodes. We observe that *fair* algorithm succeeds in obtaining a higher Jain’s index in any considered cases, whereas the other algorithms improve fairness only with very large fan-out values. Finally, Figure 14 plots the 90-network-throughput for increasing fan-out values. Performance tends to converge for the obvious reason that the topology, in all the considered algorithms, gets closer to the full-mesh case.

## VI. BRIEF SUMMARY OF RELATED WORK

Our work concerns overlay networks devised for unicast connections among nodes. We might roughly classify related works on the basis of the adopted approach, namely: *clean-slate* and *incremental*. A clean-slate approach starts from given constrains (e.g., the set of nodes to connect to, costs of overlay links, service revenues etc.) and derives the optimal overlay topology, “all-at-once”. The analytical instrument to face this calculus is integer-linear-programming [14][16]; nevertheless, even the heuristic algorithms are proposed [3]. Obviously, a clean-slate approach better fits the case of a virtual service provider that needs to deploy its overlay network, but it is not suitable for a dynamic P2P environment, as it may imply a complete re-wiring at node joining. Incremental approaches are the ones dealt in this paper and better fit with P2P scenarios. In this area, relevant works are [4][5], where the authors formulate a game-theoretic analysis and analyze performance of a best-response (BR) neighbor-selection strategy. According to the BR principle, a node chooses the set of  $k$  neighbors that averagely yields the minimum shortest-path-cost, from itself toward all other nodes. The cost of a shortest path is the sum of the cost of each traversed overlay link; different “classical” link-cost functions are examined (e.g. unitary, link delay, etc). Besides BR strategy selects the  $k$  neighbors all-at-once whereas our algorithm selects them one-at-a-time, the main difference with [4][5] is that we reveal the fairness issue and specify a methodology and a cost function to cope with.

## VII. CONCLUSION

In this work, we have addressed the issue of how to incrementally construct overlay network topologies for VPN application. This translates into the problem of defining an appropriate neighbor-selection strategy that every node joining the VPN should employ.

We have quantitatively shown that approaches devised to create “short” overlay networks, while yielding a significant

total network-throughput, may be severely affected by unfairness issues, i.e., different pair of nodes may experience a widely different throughput performance.

To face this issue, it is important to control the fairness properties of the being constructed overlay directly in the neighbor-selection strategy. We have therefore proposed an heuristic approach which combines in a single cost function both the fairness enforcement (through the control of the dispersion of the betweenness centrality values of the network nodes) as well as the average path length. Albeit preliminary and open to possible improvements, numerical results show that our proposed heuristic guarantees excellent fairness performance and satisfactory throughput.

## ACKNOWLEDGMENT

This work has been funded by the Italian research program PRIN 2007 within the “SORPASSO” project.

## REFERENCES

- [1] Khanvilkar, S.; Khokhar, A., "Virtual private networks: an overview with performance evaluation," Communications Magazine, IEEE , vol.42, no.10, pp. 146-154, Oct. 2004
- [2] De Clercq, J.; Paridaens, O., "Scalability implications of virtual private networks," Communications Magazine, IEEE , vol.40, no.5, pp.151-157, May 2002
- [3] Z. Li and P. Mohapatra. "Impact of Topology On Overlay Routing Service," in proc. of IEEE INFOCOM '04.
- [4] G. Smaragdakis, V. Lekakis, N. Laoutaris, A. Bestavros, J. W. Byers, M. Roussopoulos, "EGOIST: Overlay Routing using Selfish Neighbor Selection" ACM CoNEXT 2008.
- [5] N. Laoutaris, G. Smaragdakis, A. Bestavros, J. W. Byers, "Implications of Selfish Neighbor Selection in Overlay Networks", IEEE INFOCOM 2007.
- [6] Fraleigh, C.; Moon, S.; Lyles, B.; Cotton, C.; Khan, M.; Moll, D.; Rockell, R.; Seely, T.; Diot, S.C., "Packet-level traffic measurements from the Sprint IP backbone," Network, IEEE , vol.17, no.6, pp. 6-16, Nov.-Dec. 2003
- [7] ITU-T recommendation G.992.5, "Asymmetric digital subscriber line (ADSL) transceivers - Extended bandwidth ADSL2 (ADSL2plus)"
- [8] L. Bracciale, F. Lo Piccolo, D. Luzzi, S. Salsano "OPSS: an Overlay Peer-to-peer Streaming Simulator for large-scale networks", ACM SIGMETRICS Performance Evaluation Review, Volume 35, Issue 3 (December 2007) ISSN:0163-5999 (pdf)
- [9] R.V. Sole and R. Ferrer i Cancho, "Statistical Mechanics of Complex Networks", Lecture Notes in Physics vol. 625, Springer, Berlin (2003).
- [10] Markus Brede, Bert J.M. de Vries, "Networks that optimize a trade-off between efficiency and dynamical resilience", Physics Letters A, Volume 373, Issue 43, 19 October 2009, Pages 3910-3914
- [11] U. Brandes, "A Faster Algorithm for Betweenness Centrality", Journal of Mathematical Sociology, 2001, vol. 25, pp. 163—177
- [12] J. R. Jain, W. Hawe, D. Chiu, "A Quantitative measure of fairness and discrimination for resource allocation in Shared Computer Systems," DEC-TR-301, September 26, 1984, <http://www.cis.ohio-state.edu/~jain/papers/fairness.htm>
- [13] Silvia Gago Alvarez, Hurajová, Tomas Jana Madaras, "Notes on betweenness centrality of a graph," external research report available at <http://upcommons.upc.edu/e-prints/bitstream/2117/2891/1/GaHuMa2008v2.pdf>
- [14] A.Capone, J.Elias, F.Martignon, "Routing and resource optimization in service overlay networks," Elsevier Computer Network, Volume 53, Issue 2, 13 February 2009, pp 180-190
- [15] "GT-ITM: Modeling topology of large internetworks," <http://www.cc.gatech.edu/projects/gtitm/>.
- [16] M.Youssef, C. Scoglio, "On graph-based characteristics of optimal overlay topologies," Elsevier Computer. Network, Volume 53, Issue 7, 13 May 2009, pp. 913-925.