Challenges and Research Directions for the Future Internetworking

Miguel Elias M. Campista *Member, IEEE,* Marcelo G. Rubinstein *Member, IEEE,* Igor M. Moraes *Member, IEEE,* Luís Henrique M. K. Costa *Member, IEEE,* and Otto Carlos M. B. Duarte

***Abstract*—We review the main challenges and survey promising techniques for network interconnection in the Internet of the future. To this end, we first discuss the shortcomings of the Internet's current model. Among them, many are consequence of unforeseen demands on the original Internet design such as: mobility, multihoming, multipath, and network scalability. These challenges have attracted significant research efforts in the latest years because of both their relevance and complexity. In this survey, for the sake of completeness, we cover several new protocols for network interconnection spanning both incremental deployments (evolutionary approach) and radical proposals to redesign the Internet from scratch (clean-slate approach). We focus on specific proposals for future internetworking such as: Loc/ID split, flat routing, network mobility, multipath and content-based routing, path programmability, and Internet scalability. Although there is no consensus on the future internetworking approach, requirements such as security, scalability, and incremental deployment are often considered.**

***Index Terms*—Future Internet, internetworking, routing.**

I. Introduction

In the early 1980's, the Internet emerged with the adoption of the TCP/IP protocol stack. IP was designed to support requirements such as network interoperability, end-to-end connectivity, and global access [1]. These requirements were achieved by keeping the network core simple and independent of upper- and under-layer protocols. From the beginning, in addition to IP, a routing protocol was used to interconnect local networks. Although the role of a routing protocol was the same as today, the early protocols computed the shortest path between any source-destination pair, since the Internet was composed of a few nodes from non-profit institutions. At that time, the Internet was so small that GGP (Gateway-to-Gateway Protocol) [2], one of the former routing protocols, used to list the number of hops to every other network in the Internet inside update messages. The original Internet architecture is presented in Figure 1.

In the following years, the number of users and networks increased above any expectation. As a result, topology updates became more frequent and GGP evolved to a complex decentralized system. In such a scenario, there was more than one

M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte are with GTA/PEE/COPPE/DEL-Poli - Universidade Federal do Rio de Janeiro - Rio de Janeiro, Brasil, e-mail: {miguel,luish,otto}@gta.ufrj.br. Marcelo G. Rubinstein is with PEL/DETEL-FEN - Universidade do Estado do Rio de Janeiro - Rio de Janeiro, Brasil, e-mail: {rubi}@uerj.br. I. M. Moraes is with MídiaCom/IC - Universidade Federal Fluminense - Niterói, Brasil, e-mail: {igor}@ic.uff.br.

Miguel Elias M. Campista is the corresponding author.

Manuscript received XXX; revised YYY.

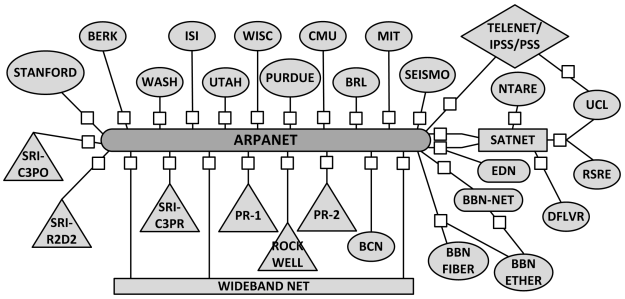


Fig. 1. Internet architecture in the early 1980s (Adapted from [3]).

version of GGP in use, which negatively impacted network interoperability. These problems culminated in the division of the Internet into multiple Autonomous Systems (ASes). An AS is defined as a set of networks and routers administered by a single institution and that do not depend on other ASes to be internally connected. From then on, each AS could select its own internal routing protocol, called intradomain protocol, while another interdomain routing protocol was used to interconnect the ASes. The interdomain protocol is the common language used by all ASes to maintain global connectivity. The first one was EGP (Exterior Gateway Protocol) [4], which had scalability issues and was later replaced by BGP (Border Gateway Protocol) [5], which is the interdomain routing protocol being used since then.

The emerged network interconnection design proved a great success. An evidence of Internet success is the astonishing growth in the number of networks. Measurements show that the current Internet has at least 207 million /24 networks, representing 93.3% of all routable prefixes in the Internet [6]. The Internet growth was about 566.4% from 2000 to 2012 [7]. Figure 2 shows the growth of the number of networks announced by BGP routers from beginning 2003 to beginning 2013 [8].

The recipe for success was the Internet popularization and the number of business opportunities it revealed. At the AS level, business opportunities have shown up with agreements among ASes for traffic relaying. The popularization of the Internet, on the other hand, is a consequence of the relatively simple creation of new applications, which do not require changes in the Internet core, and the timely way users can obtain desired information. Internet providers are promptly exploring this promising market, launching many access plans. Therefore, the demand meets the supply, leading the number of connected users to increase at the whole world. Figure 3

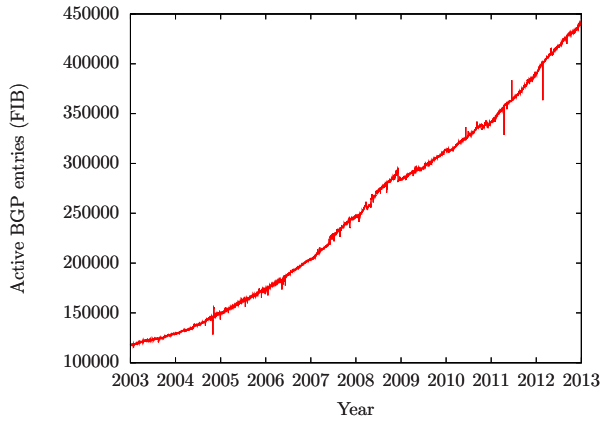


Fig. 2. Growth of the number of networks as indicated by the increase on the number of active BGP entries (Adapted from [8]).

shows the increase on the global number of subscriptions to networking services as well as the increase on the number of Internet users from 2003 to 2013 [9]. The data is from consolidated annual reports (2012 and 2013 are estimates).

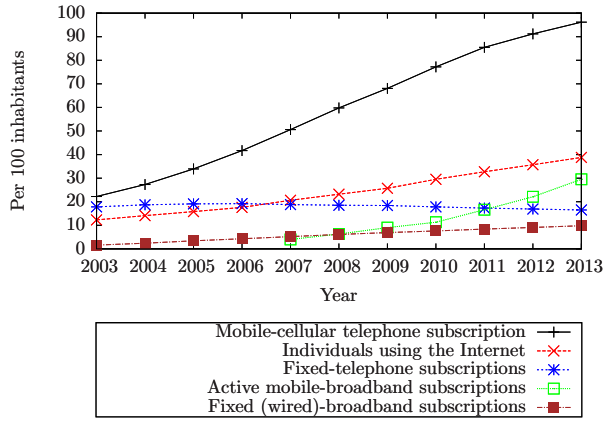


Fig. 3. Growth of the number of subscriptions to networking services and of Internet utilization (Adapted from [9]).

Similarly to what happened in the past, the Internet is reaching another inflection point where some kind of redesign must be made so that it can continue to grow. The Internet success did not arrive without consequences, such as the starvation of IP addresses or the explosion of routing tables. Hence, even consolidated protocols such as BGP and IP need to be rethought. This claim is reinforced by the four updates BGP has undergone since 1989, when it was launched, until 1995 [5], [10], and another one in 2006 [11]. Some of BGP's problems are related to the IP shortcomings discussed in this work, which did not evolve at the same pace as the Internet. Thus, besides the Internet growth, neither the initial design of IP nor the existing routing protocols were developed having in mind the possibility of changes on future application requirements. Although new proposals or extensions have flourished, e.g., IPv6 [12], mobile IP [13], multicast [14], [15], none of them has completely addressed all the problems or has been fully adopted. Another alarming issue is the lack of scalability faced by Internet routers, which is an additional obstacle to the new requirements.

Today, there is a strong debate in the networking community about the next step toward the Future Internet, but the answer is not straightforward. The research community is divided by two trends: the complete rupture with the well-known Internet model, including the design of a replacement to IP [16], or the maintenance of the current Internet architecture and the addition of extensions according to the upcoming requirements [17]. The former may lead to deep economic impact for Internet stakeholders, which already have their operational base. The latter, on the other hand, may only postpone problems we are already dealing with. An important design decision is whether new protocols must embrace all the new emerging requirements or leave some special cases aside. There is a rough consensus though that the new proposal must be scalable and prone to evolution, to cope with future Internet requirements because it would not be practical to start the same evolutionary *vs.* clean-slate contend all over again from time to time [18], [19], [20].

This paper presents the main architectural constraints for network interconnection to handle emerging requirements such as mobility, multihoming, multipath, and path customization. We discuss as well routing scalability issues and survey the recent literature to identify the main directions taken to address Internet limitations. Independently of how radical the proposal is, a new Internet architecture or another extension, it typically does not focus on all emerging requirements at once. Some of them aim at tackling them individually, but can also have an indirect impact on others. The impact sometimes represents a positive side-effect or a tradeoff to be taken into account. Therefore, we present and discuss proposals pros and cons and their resultant impacts. We propose a taxonomy to organize such proposals based on their main characteristics and on the problem they deal with. Seven broad approaches are identified: locator-identifier split, flat routing, network mobility, Internet multipath forwarding, Internet scalability, content-based routing, and programmable paths. It is worth mentioning that we do not have the ambition to make an exhaustive search, but we would like to encourage the research in the area by revealing the main challenges and how they have been tackled so far.

This work is organized as follows. Section II presents the current Internet architecture and basic definitions. Section III discusses the main challenges faced by today's Internet and also presents the constraints of its architecture. Section IV describes new directions in the area to improve network interconnection, while Section V presents the main proposals for the Future Internet. Section VI summarizes the main characteristics of the surveyed proposals and discusses additional aspects not directly addressed. Finally, Section VII concludes this work.

II. The Internet Architecture

The current Internet architecture is composed of different networks interconnected via IP, running an intradomain and an interdomain routing protocol.

A. Intradomain level

At the intradomain level, stations and routers are administrated by the same entity. This is a key advantage since it avoids problems concerning contrasting requirements from different parties, interested to configure the network according to their needs. Once the intradomain routing protocol is chosen, it is used to provide Internet access to all of the internal stations. A routing metric can be chosen to evaluate the best path in a given topology. Upon choosing the most suitable metric, choosing the best path can be seen as an optimization problem. Typically, routing protocols solve the shortest-path problem using algorithms such as Bellman-Ford and Dijkstra.

The Route Information Protocol (RIP), for instance, uses distance-vectors and a distributed version of the algorithm of Bellman-Ford to choose the best path [21]. Each router periodically sends its view of the network to its neighbors, i.e., a list of the known routes and the distance (metric) to each destination. Once a neighbor receives this distance-vector, it checks its own table, computes shortest routes if possible, and sends new distance-vectors to its own neighbors. The algorithm converges when, upon reception of distance vectors, there are no more changes in any of the routing tables. The shortcoming of RIP is observed if, for example, distance-vectors are lost. In this case, out-of-date paths install and, consequently, routing loops can appear. Some loops may only be solved when routers "count to infinity" the route metrics, leading to slow convergence.

Link-state routing protocols have been proposed as an alternative to distance-vector with the advantages of faster and loop-free route computation. These protocols are based on a topology map, which is replicated at all nodes. The topological view is maintained by nodes upon periodically flooding the state of its outgoing links to all other nodes in the network. The global view of the network permits route computation to be executed in a centralized fashion. Hence, the Dijkstra's algorithm does not result in routes with loop formation. Because there is no need to rely on counting to infinity to solve routing loops, more complex metrics can be used together with modifications of Dijkstra's algorithm. Multiple paths and load balancing can also be used, thanks to the complete map of the topology each router possesses. The Open Shortest Path First (OSPF) protocol is link-state based and is one of the most used intradomain routing protocols in the Internet [22].

Every network receives a block of IP addresses, which is used to allocate IP addresses to routers and stations. In practice, the shared IP address prefix and each IP address can identify an access network and a station, respectively.

B. Interdomain level

The Internet architecture evolved to a complex hierarchical AS topology [23], [24], [25]. At the interdomain level, the same simplicity of intradomain networks does not hold because ASes are administered by independent organizations. Therefore, the interconnection problem becomes far more challenging.

The Internet topology can be represented by a graph with ASes as nodes and AS connections as links. Such links are established according to the relationship between the different ASes and their position within the AS topology graph. ASes can be classified using a combination of customer-to-provider and peer-to-peer relationships, which correspond to different business models. ASes are considered customers if they have their traffic relayed by other ASes, which play the role of providers. In this case, the customer AS pays the transit provider for the service. On the other hand, backbone ASes typically agree on relaying traffic without financial costs, following the peer-to-peer model. The business model can also be derived from the position of the AS within the Internet topology. ASes at Internet edges, named AS_{Edge} in Figure 4, are directly connected to users and are typically customers of other ASes[1]. These ASes are called stub because they do not relay traffic on behalf of other ASes. Stub ASes or stub networks typically have a single default route to the external world, internally announced by border routers.

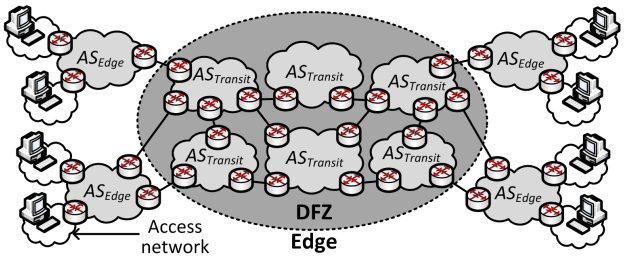


Fig. 4. Today's Internet architecture. The network is composed at the edges of access networks and customer ASes, and at the DFZ it is composed of transit ASes.

Transit ASes ($AS_{Transit}$ in Figure 4), which are not directly connected to users, form the Internet core. Their main role is to relay traffic to and from Internet edges, considering that traffic sources and destinations are placed at the network edges. Transit ASes may charge their neighboring ASes depending on the established agreement and the amount of traffic injected by the neighbor within the infrastructure of an AS. Hence, they can also play the role of customer, if they use a neighbor AS to relay traffic; or they can only have peer-to-peer relationships, if they do not have their traffic relayed by neighbors. Transit ASes usually do not use default routes, composing the Internet Default Free Zone (DFZ), and therefore, it is assumed that these ASes have a global view of the Internet. Organizations that are responsible for at least one AS, customer or transit, are called ISPs (Internet Service Providers). Figure 4 depicts the current Internet architecture, where access networks are connected to customer ASes.

C. Router architecture

As already mentioned, ASes are composed by networks and routers and it is important to understand the basic functioning

[1]ASes can also be classified based on the nature of their connections to other ASes. Informally, ASes can be ranked into three levels of tiers. ASes connected to users are ranked as Tier-3. In opposition, ASes not connected to users can be ranked as Tier-2 or 1, depending if they play the role of customer ASes (Tier-2) or not (Tier-1).

of routers. They have two tables with distinct functions in packet routing: the Forwarding Information Base (FIB) and the Routing Information Base (RIB). The FIB stores the next hop router and the corresponding outgoing interface to all known destinations. Therefore, the FIB is looked up whenever a router receives a data packet and, if there is a match, the router forwards the packet to the corresponding outgoing interface. Otherwise, a default route can be used. The RIB, on the other hand, is the data base used to compute the FIB route entries. For instance, using OSPF, the FIB is built according to the routes found by the Dijkstra's algorithm, after running it over the network topology map stored in the RIB. To be efficient, FIBs are constructed to speed up lookups whereas RIBs are built to permit safe and quick updates. Figure 5 illustrates a basic router scheme, highlighting the path taken by control and data packets.

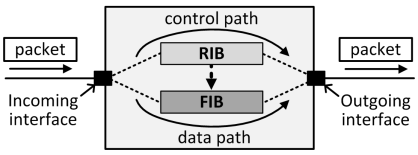


Fig. 5. Basic router scheme. The Routing Information Base (RIB) stores control information, which is used to compute the Forwarding Information Base (FIB). FIB contains all required information for packet forwarding.

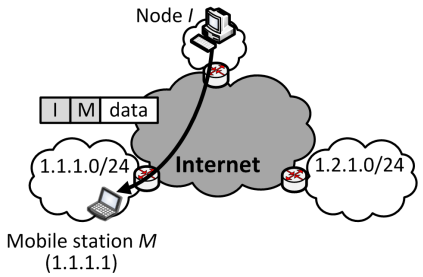
III. Challenges in Future Internetworking

Most challenges faced by the Internet today are a consequence of the architectural design decisions and the unexpected fast growth. Requirements such as mobility, multihoming, multipath, and path customization, were not anticipated in the original design of the Internet. Similarly, at the beginning, even the most optimistic would never foresee the great Internet success. In this section, we discuss in more detail the emerging challenges of network interconnection in the Internet.

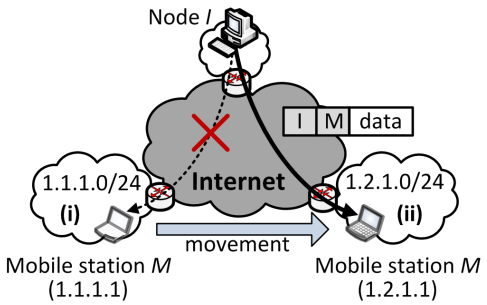
A. Mobility

Wireless networks and node mobility are two of the main future Internet challenges. One classical example is the TCP operation in wireless environments. Because it was developed for the wired Internet, TCP performs poorly with frequent transmission errors [26]. TCP assumes that large delays or packet losses are a consequence of network congestion and thus reduces transmission rates to avoid network collapse. In wireless networks, however, the same principle does not hold because transmission errors are nothing but negligible. Thus, reducing the transmission rate represents a waste of network resources. Another problem is nodes' mobility which introduces challenges in addressing and routing. Internet addressing is hierarchically organized to permit address aggregation and therefore to improve routing scalability. For example, prefixes used by access networks connected to an edge AS can be aggregated and announced together as a unique network. As a consequence, the number of entries in FIBs is reduced and route lookups become faster.

The hierarchical structure of the Internet has led to a geographical organization of IP addresses. The IP address is associated with the station location in the Internet topology. At the same time, the identifier of an Internet station is obtained from a name mapped into an IP address by the DNS (Domain Name System). The IP address is then used to locate and also to identify a station. This characteristic is called the "overloaded semantics" of IP addresses, which is a fundamental obstacle to mobility. As a node moves, it changes its location, and consequently, it must change its IP address. Therefore, a station must always reconfigure its IP address to one topologically coherent with the visited network. Because the current Internet architecture assumes invariant IP addresses, upon receiving a new address, all connections established using the prior address are lost. Figure 6(a) shows a mobile station M in its home network. Note that the packets sent from an Internet node I can be correctly delivered to M. In opposition, in Figure 6(b), after changing its access network, the IP address of M must be reconfigured, requiring a new connection with the Internet node I.



(a) Mobile station M directly receives packets from node I in the Internet, before changing the access network.



(b) Address reconfiguration: (i) station M leaves its access network and, consequently, loses all packets destined to it; (ii) upon arriving at the new access network, it reconfigures its IP address and reestablishes its previous connections.

Fig. 6. Station mobility.

Mobile nodes are frequently associated to wireless networking. Nevertheless, there are other types of mobility which do not necessarily include wireless environments. In cloud networking, virtual machines can migrate from a network to another, changing their topological position. This operation introduces flexibility since a virtual machine can still be operational even though it changes its physical location. This possibility is interesting for traffic engineering, network

maintenance, and green networking, to cite a few. Although the problem is similar to classical node mobility in wireless environments, virtual machine migration can also have additional challenges not addressed in wireless networking. Live migration is an example of node mobility with no duality in wireless networking [27], [28]. Live migration requires preserving the current state of all processes in execution. This incurs in saving the current memory state of all processes in execution and copying this state to the new physical host before migrating the virtual machine. The complete procedure allows resuming all the processes of the migrated virtual machine without disrupting its services. This issue opens venue for work on operating systems, which is not the focus of this work.

***Current Internet status**:* Mobile IP [13] is one of the former proposals to handle mobility in the Internet. The idea is to maintain the association between the mobile station and its original IP address even after the mobile node changes the network of attachment. To this end, the original network must always be informed about the mobile node's current location. Packets sent from a corresponding node to this mobile node can use the address of the mobile node in the original network as destination address in IP packets. As the original network keeps track of the current mobile station address, this node can have its packets forwarded to the visited, or foreign, network. The address in the foreign network is tracked in the original network by a home agent, which is a machine in charge of receiving and forwarding all traffic destined to the mobile station. The home agent encapsulates the packets to the mobile station and forwards them through a tunnel established with a machine in the foreign network, called foreign agent. The foreign agent assigns valid IP addresses (Care-of-Addresses - CoAs) to visiting mobile stations and forwards the received packets from the tunnel to the final destination. The operation of Mobile IP is illustrated in Figure 7. All packets sent from the corresponding node I to the mobile station M are received by the home agent H, which encapsulates and forwards the packets through a tunnel to the foreign agent F, as seen in Figure 7. The foreign agent F decapsulates the packets and forwards them to the mobile node M. The traffic in the reverse direction is originated by M and sent to the foreign agent F, which encapsulates and sends the packets to the home agent H through the tunnel. Home agent H receives, decapsulates, and sends the packets to the Internet. Although such procedure enables roaming of mobile stations, it results in indirect forwarding, reducing Internet routing efficiency.

Mobile IPv6 [29] does not require foreign agents. Hence, the tunnel is established between the home agent and the mobile station. Besides, mobile IPv6 permits packet forwarding from the Internet to the mobile station without passing through the home network, using IPv6 routing header extensions. First, mobile stations use the IP address of the foreign network as the source address. The home network address is listed in the header extension and is used by Internet nodes to recognize the roaming station with a new address. Then the new mobile station address is used as the destination of the following packets. It is worth mentioning that this improvement is only possible in fully routable IPv6 networks. Because the required

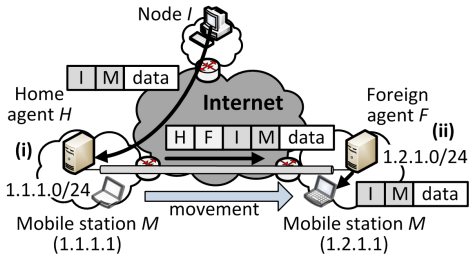


Fig. 7. Mobile IP operation: (i) station M leaves its access network and, as a consequence of Mobile IP, it still receives the packets destined to it; (ii) packets are forwarded by the home agent via tunneling towards the station M at the new access network.

header extension does not exist in IPv4 packet header format, the only possibility is the classical triangular routing if IPv4 routers are also present. Even considering the IPv6 header extension, mobile IPv6 still has problems with roaming at high speeds since it still involves address updates when changing to another visited network. In addition, IPv6 has not been fully applied in practice, requiring other alternative solutions to maintain roaming station connections.

A more general problem has been investigated to support an entire mobile network instead of a single station. This is the purpose of the NEMO (NEtwork MObility) working group [30], which aims at extending the connectivity to mobile nodes, even if they do not have direct access to interconnecting points. The NEMO approach is based on the Mobile IP operation, either IPv4 or IPv6, since it also relies on an architecture composed of home and foreign agents. NEMO is addressed in more detail in Section V.

B. Multihoming

Multihomed stations or multihomed networks are connected to the Internet via more than one access network and therefore may have multiple IP addresses from disjoint address prefixes. These addresses can be obtained from different ISPs or from the same ISP, introducing some challenges from the routing viewpoint. Figure 8 illustrates a multihomed access network (site multihoming) and a multihomed station (host multihoming). Multihomed stations have network interfaces configured with different IP addresses whereas multihomed networks are connected to border routers providing Internet access via different ISPs. Stations within an access network may have private IP addresses or IP addresses from both ISPs (AS_A or AS_B). Alternatively, multihoming can also be used to avoid address reconfiguration. In that case, the multiple address prefixes are not assigned by the directly-connected ISP, i.e. not provider assigned, but instead are assigned by a Regional Internet Register (RIR). As a consequence, the multihomed network is independent of the ISP, or provider independent, and can undergo an ISP change without address reconfiguration.

***Current Internet status**:* Multihoming advantages are many-fold: it increases communication reliability [31], [32] because an ISP problem can be circumvented by using another one, as soon as the configurations permit failure isolation; it

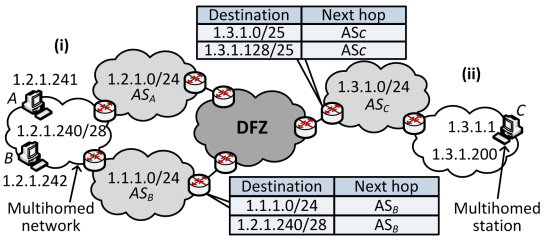


Fig. 8. Example of a (i) multihomed access network and a (ii) multihomed station. In multihomed access network, the different IP address ranges are inherited from different ASes (AS_A and AS_B).

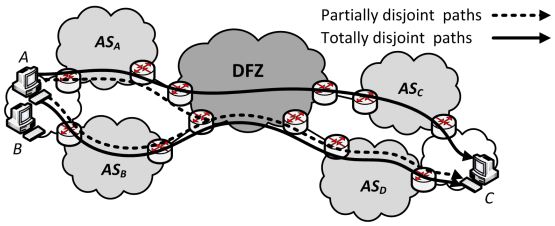


Fig. 9. Partially and totally disjoint multiple paths. In the first one, the multiple paths partially overlap; whereas in the second one, they are totally disjoint.

allows traffic engineering, throughput maximization, and cost reduction (e.g., more expensive ISPs are used for sensitive traffic whereas cheaper ones are used otherwise); and it avoids network address reconfiguration after an ISP change. Those advantages motivate multihoming employment despite its direct impact on scalability. Because non-aggregated prefixes must be announced to all ASes in the Internet, the premise of hierarchical address organization is violated. Considering that each provider has pre-established address blocks, announcing non-aggregated prefixes leads to scalability issues with respect to BGP table growth. In the last few years, the number of BGP entries has exponentially increased and is still moving up. CIDR numbers show that from January 2012 to January 2013 approximately 50,000 new active BGP entries have shown up [33]. Therefore, providing all the advantages of multihoming with negligible impact on Internet scalability requires further investigation.

C. Multipath

Originally, the Internet was designed based on single-path routing algorithms. Thus, routing protocols typically announce a single alternative for each destination. Along the years, multipath techniques, i.e., the use of several alternative paths to reach destination, appeared to better use the ever richer Internet topology.

Multipath techniques allow spreading traffic among different paths. Main multipath advantages are: it increases the available bandwidth, and so provides shorter delays; it increases fault tolerance, by allowing the use of other routes when one or more routes become unavailable; and it enables traffic engineering and load balancing. This last advantage allows the selection of a path according to application requirements, e.g., paths with larger bandwidth or lower delay, and avoid congested paths [34]. Figure 9 illustrates multipath forwarding involving paths from node A to C. Unlike multihoming, a data flow simultaneously uses the multiple paths for load balancing. Figure 9 illustrates an example of totally- and partially-disjoint paths. Although Figure 9 depicts end-to-end paths, proposals to multipath deployment can be focused on at least three key Internet levels: at the access, at the edge, and at the core.

Multipath forwarding is a possibility not fully exploited in the Internet, i.e., at the edge and core levels, despite its recognized potential [35]. Measurements show that there is an alternative path with lower loss rate and lower delay than the chosen one in 30 to 80% of the time [34]. Using alternative paths could significantly improve packet forwarding performance, but this capacity has been left behind because multipath routing has an impact on the upper layers. For instance, packets from one TCP connection may arrive out of order if path diversity is used at a fine-grained packet level, instead of at the flow level. At the access level, intradomain routing protocols usually choose only one path to interconnect any source-destination pair. Protocols such as OSPF and RIP (Routing Information Protocol) maintain in their RIBs only the next hop toward known destinations [36]. OSPF, however, is sometimes used with variants that are able to compute multiple paths. Even in this case, the multiple paths are not necessarily used for simultaneous transmissions and can be maintained for backup purposes. This characteristic is an obstacle to reliability and flexibility, easily achieved by splitting the traffic among the multiple paths.

Current Internet status: Although the Internet topology provides multiple paths because of the multiple AS connections, problems concerning scalability, commercial agreements among ISPs, and the BGP design make the deployment of such technique a distant reality in the interdomain level [34]. Storing multiple paths to every other destination in the network may impact scalability, and in addition, computing multiple paths requires the execution of more complex algorithms, further increasing the number of entries in FIBs [34]. Moreover, to become fully exploited, multiple paths must be able to pass through networks of different ISPs which do not necessarily have mutual agreements. Finally, although BGP may receive multiple announces toward a destination network, similarly to intradomain protocols, BGP was developed to provide only one path to each Internet network prefix. Each ISP announces to its neighbors a single active path based on his own policies. Some vendors support storing multiple paths in the routing table to allow load sharing, but this feature has never been standardized and experiments observing Internet AS-level routes toward thousands of destinations revealed that BGP multipath is practically not used today [37]. Moreover, if more than one path were offered by ISPs to its neighbors, part of the traffic control could be lost because they would not be able to anticipate how their neighbors will distribute their traffic. This knowledge is important for ISPs to perform traffic engineering inside their own network infrastructure.

At the upper layers, multiple paths are also appealing even considering all the obstacles from the network layer. For instance, TCP has a multipath version named MultiPath TCP (MPTCP) [38], [39]. In MPTCP, the sender divides a single

flow into multiple subflows and uses additional TCP options for data reordering at the receiver. As the original congestion control mechanism was designed for single-flows, MPTCP links the congestion control mechanism to the multiple paths, concentrating traffic on subflows traversing less congested paths. The drawback of MPTCP is its requirement of existence of multiple paths and of the discovery of them by the underlying routing protocol.

D. Programmable paths

One possibility to decouple Internet paths from routing protocols and agreements of ISPs is to empower users with the ability to configure Internet paths themselves. The best path could be chosen according to user-level requirements, such as quality of service parameters; or could be dynamically chosen by intelligent agents [1]. To accomplish that, the Internet must handle user-level metrics and the current ISPs would have to be adapted to provide this new customized service. This possibility, however, has to deal with the best-effort service currently provided by IP. A programmable Internet must support differentiated services and provide support for users or agents to choose paths at their own will, independently of providers' agreements. This is not unrealistic, if we consider that the Internet intelligence is maintained at the network edges and there is no one better than the user to know whether an application performance is acceptable [1].

The freedom for users to choose paths can stimulate competition and, hence, lead to a reduction on the access costs. One question that remains unanswered regards the absence of end-to-end quality of service in the Internet. One of the claimed reasons for that is the lack of competition among ISPs. In other words, because this service is not offered by any ISP, there is no reason for anyone to apply. This scenario will be transformed if one pioneer ISP decides to offer path programmability. In this case, users with special requirements would choose on demand the most appropriate path for their application. This service could guarantee ISPs a privileged place at the market share because they would attract users who are waiting for such opportunity. Other ISPs would have to take countermeasures to regain competitiveness [1]. If the first ISPs do not show up, incentives could be offered to motivate first initiatives.

The potential disadvantages of user or agent freedom is the increased complexity each one must deal with. For instance, users or agents would need wide network knowledge and, even in this case, they could lead to flaws because of misuse or malicious behavior. Moreover, choosing paths may require nodes maintaining multiple customized paths, further impacting scalability. All these challenges could be tackled if decisions were centralized and the network was limited to a smaller number of switches or routers. Indeed, this is an alternative for local area networks with a limited number of forwarding elements [40], which can be controlled by a single entity. Although it does not solve path programmability at Internet scale because of scalability issues, it can already be considered an evolution towards path programmability.

Current Internet status: Currently, packets follow the path selected by routing protocols. Intradomain routing protocols, on the one hand, choose the best path between source-destination pairs based on pre-established metrics, e.g., hop count and delay. Interdomain routing protocols, on the other hand, choose neighboring ISPs for relaying traffic based on policies. In this case, traffic is forwarded depending on agreements that could limit network services such as the maximum throughput. BGP, the *de-facto* interdomain protocol, requires many manual configurations from network administrators, which is totally opposite to programmable paths at the network edges or intelligence at the network core.

In a more limited scope, software-defined networking (SDN) [40] is emerging as an alternative to enable programmability by decoupling the forwarding and control functions of a network element. Different from active networks, SDN neither radically changes the forwarding elements nor the contents of a packet. In a software-defined network, a controller manages the forwarding table of a switch via a programmable interface. Hence, although switches are still responsible for packet forwarding, control decisions are taken by an external controller regarding path programmability.

E. Router scalability

Scalability is one of the most critical challenges in the short term for the Future Internet design, according to reports from the Internet Architecture Board (IAB) [41]. Routers have memory and processing limitations. Hence, the unlimited increase on the number of entries in routing tables (FIBs) and data bases (RIBs) may impact the performance of packet forwarding. Each router needs a certain amount of time, which must be as small as possible, to store a packet, check for the best outgoing interface based on the destination address, and forward the packet. Moreover, the control overhead produced when the number of stations increases may become more and more relevant compared with the available bandwidth. Measurements from CIDR show that since the beginning of the 1990's the number of active BGP entries in FIBs has increased from a few hundreds to approximately 450,000 entries [33]. Other CIDR numbers show that about only 42.4% of the total number of Internet networks use aggregated prefixes [42], demonstrating the problem magnitude. Partial address aggregation, as well as many Internet challenges, can culminate in longer routing tables and in more control overhead. These side-effects are obstacles to the Internet growth considering that address lookup and message processing can be computationally expensive for routers. Proposing new algorithms to improve router scalability is, therefore, a major challenge to the new Internet architecture.

Current Internet status: Scalability issues could have lower impact if the original Internet architecture was strictly followed. The assumption of hierarchical organization and the consequent address aggregation aim at routing scalability [43], [44]. Nevertheless, the emerging requirements hinder address aggregation on FIBs and increase the network control overhead. The support for multihoming requires the association of disjoint prefixes to the same network and the

support for multipath requires the association of more than one path to the same destination. User mobility increases the amount of control information exchanged to maintain a mobile station connected and, additionally, is incompatible with the correlation between geographical location and identification used for FIB aggregation. At last, path programming incurs in more routing information stored at the network edges. An indirect effect that could also impact scalability is the use of address spaces larger than the one provided by IPv4. The jump from 32 to 128 bit address space with IPv6 means that an unprecedented number of users and networks are possible in the Internet, which can further add more routes in FIBs and control overhead in the network.

In practice, scalability problems reflect on the routing table size of BGP, mainly because of partial address aggregation. Coming back to Figure 8, the edge router of AS_B announces a network prefix not aggregated with its own. The edge router from AS_C, on the other hand, halves its network prefix and announces them disaggregated because of multihomed stations. This is often interesting for service providers because it allows traffic engineering and facilitates differentiated services deployment, but, in opposition, increases BGP table sizes.

IV. Research Directions for the Future Internetworking

In this work, we focus on the five Internet routing challenges listed in Section III: mobility, multihoming, multipath, path programmability, and scalability. Although the list is not exhaustive, it brings up a number of insights regarding what can be expected for the future Internet. This list is based on requirements that have emerged along the first 25+ years of Internet utilization.

A suitable classification for all proposals focused on future internetworking issues is a challenging task. Therefore, complementary classifications or orthogonal ones can be proposed without loss of generality. The classification can be a consequence of either a more general study or a more focused one. In this work, we surveyed the literature seeking for contributions that directly or indirectly tackle at least one of the highlighted challenges. The result is organized in seven distinct classes as follows:

- Loc/ID split
- Flat routing
- Network mobility
- Multiple paths
- Content-based routing
- Programmable paths
- Internet scalability

The first four classes are somehow a consequence of the "ossified" structure of IP routing and addressing, which was not designed for mobility, multihoming, and multipath routing. The following two classes, content-based routing and programmable paths, are two approaches that can potentially disrupt the current Internet architecture. Both are concerned with the Internet service at the users' point of view. Nevertheless, whereas the first one is based on users dealing with the Internet as a black box, i.e., they are only interested in receiving the content requested no matter where it comes from; the latter expands the interface between the users and the Internet, allowing users to further interfere on routing issues. The last class, scalability, is a requirement that has been present in any list of Internet challenges. This section introduces the rationale behind our classification, whereas Section V provides details of the main proposals of each class.

A. Locator-identifier split (Loc/ID split)

One of the main proposals to incorporate mobility and multihoming to the Internet is to decouple station topological locator from its identifier. This is the idea of Loc/ID split proposals, which break the overloaded semantics of the IP address. In Loc/ID split proposals, multihoming becomes easier because one ID can be associated to more than one locator. Therefore, reaching the same node is possible by using any of the correlated locators. Mobility is also simplified by using the same approach. Connections can be established using identifiers, instead of topology-based addresses. Hence, changing the topological position does not lead to connection interruptions.

Loc/ID split proposals can be further divided into three subclasses, namely indirect forwarding, network-, and host-based approaches. The first one uses an intermediate system to maintain end-host's identification and corresponding location. This system resolves node's location or intermediates forwarding procedures. This last possibility is also known as triangular routing. The second approach uses a network locator for forwarding procedures within the DFZ and an end-host identifier within a local scope at the network borders. This kind of approach uses border routers in charge of mapping locators to identifiers. The third approach, on the other hand, uses global end-host identifiers to establish end-to-end communications. Host-based approaches can also use mapping systems during connection establishment. Network- and host-based approaches can be combined in a mixed approach using mapping systems as well as end-to-end identifiers.

B. Flat routing

Flat routing is another possibility to circumvent the IP address overloaded semantics. It extrapolates the concept of Loc/ID split by ignoring the locator and performing packet forwarding solely based on identifiers. This requires identifier uniqueness and imposes the total decoupling of node identifier from network topology. Flat identifiers can be typically obtained using DHTs (Distributed Hash Tables), which can uniformly distribute the workload and the functionality, avoiding overwhelmed nodes. In peer-to-peer applications, for instance, it is desirable to have a balanced distribution of data throughout the network. Flat routing can then take the DHT idea and use it for routing instead of content distribution. Once communications are based on identifiers, and not on addresses related to the network topology, nodes can roam without losing connections.

Flat routing proposals must consider intra and interdomain communications. ROFL (Routing On Flat Labels) [45], for instance, adapts peer-to-peer protocols to the routing case: it

uses Chord [46] in intradomain and Canon [47] in interdomain level. Other proposals can either use DHT-based approaches, hash-based identifiers or indirect forwarding, as will be seen in next section. Flat routing must be investigated even considering the Internet premise of keeping structured information of node location on packet header.

C. Network mobility

IP addressing is the main reason mobility is very difficult in the Internet. If we consider that neither Loc/ID split nor flat routing are practical short-term solutions, it would be interesting to improve Mobile IP. For instance, Mobile IP builds upon a basic architecture composed of a foreign and a home agent, a corresponding node, and a single mobile station. This simple architecture has quickly become a limiting factor with the increasing popularity of wireless networking, e.g., mobile ad hoc networks, vehicular networks, and delay tolerant networks. All of them require mobility support to cases where the entire network moves and not only a single station.

Network mobility allows stations from the same network to roam together and share the same Internet access. The goal is to extend connectivity even to mobile nodes without direct access to interconnecting points even after they change their home network. Hence, in opposition to infrastructure wireless networks, where only nodes within range of access points can have Internet access, the entire mobile network will do. This is the problem tackled by network mobility proposals, which follows the NEtwork MObility (NEMO) architecture [30]. Following NEMO, other proposals aim at improving its performance, reducing the number of triangulations in packet forwarding. Source routing, for instance, is a straightforward solution to avoid the participation of multiple home agents in packet forwarding.

D. Multiple paths

Although there are known algorithms to compute multiple paths, this approach applied to the Internet, i.e., at transit and core levels, often faces problems regarding the inertia for changes and the huge number of complex policies. Even though there are such obstacles, proposals for multipath routing typically rely on source routing or on overlay networking. The first one defines the multiple paths at the source; whereas the latter abstracts networking constraints by setting different paths at upper layers. In both cases, ISPs can still lose part of the network control because when using source routing the path is chosen at the source node whereas when using overlay networks decisions may not consider internal policies of ISPs. The overlay possibility, however, does not require modifications to the network core. Therefore, the multiple paths can be chosen at the overlay layer and packet forwarding can use traditional IP routing. Tunneling is another alternative that can create virtual links from multiple underlying physical ones. The topology can then be changed for multipath purposes.

E. Content-based routing

Content-based routing is based on the premise that users are now valuing the Internet for "what" content it makes available. The IP communication paradigm, however, is based on "where" this content is located [48], [49], [50]. Content-Based Networking (CBN) (also referred to as Content-Centric Networking, Data-Oriented Networking, Information-Centric Networking, and Name-Oriented Networking) is emerging as a new communication paradigm to achieve scalable and efficient content distribution [48], [51], [52]. CBN aims at finding specific data regardless of its location. Users then can consider the Internet as a black box where they retrieve content no matter where it comes from. To accomplish that, the basic CBN primitive is content identification, i.e., packets are forwarded based only on the content name and not on the destination address [53], [54]. The main advantage is favoring applications based on content distribution by increasing the efficiency of content location and delivery. Dealing with user mobility is also less complex because CBN decouples content identification and location.

CBN routing is different from IP routing. In CBN, the routing protocol has to acquire information about the contents available in the network to forward requests towards potential sources. Routing decisions are made according to content names. Content-based routing mechanisms can be divided into structured and non-structured mechanisms [54]. Non-structured routing protocols for CBNs consider that there is no dedicated infrastructure to maintain routing information and routers are not hierarchically organized. Thus, routing information is propagated among nodes, and paths between content sources and users are computed similarly to the traditional routing protocols in the current Internet [51], [48]. With structured routing protocols, on the other hand, specific nodes are responsible for maintaining routing information. Structured CBN protocols are based on hierarchical trees [52] or on DHTs [55], [56], similarly to Loc/ID split proposals. By using hierarchical trees, for example, a router maintains information of all the contents stored by nodes below it in the tree. Thus, the publication of a new content or the modification of an existing one generates control messages that are only propagated up along the tree until it reaches a router that has the corresponding routing entry. On the other hand, hierarchical protocols may also experience scalability problems depending on the naming approach adopted.

F. Programmable paths

Programmable paths emerged from the need for service differentiation in the Internet. Providing quality of service in the Internet involves many different application requirements. Hence, researchers believe that new services, other than the traditional "best effort", would require Internet add-ons for network topology awareness as well as sophisticated methods for new applications design. These new methods could be used to move end-to-end functionalities to the network core [1]. For instance, monitoring the network performance and giving support to path customization could be an alternative to change the Internet *status quo*. Path customization could be supported by introducing intelligence at the network core, which would require knowledge acquisition and network global view. The first requirement can be supported by software agents acting

on the network using the available knowledge, such as the one gathered in a Knowledge Plane [57]. These agents would be able to take autonomous decisions about the best path for a given application. Besides using intelligent agents, the path choice can also be made by the user himself. To this end, architectures empowering users to choose the path followed by their own traffic at the AS level must be implemented, instead of only offering them default paths without any kind of interaction [18]. Therefore, paths must be visible to the user as well as their main characteristics. It is worth mentioning that path customization in the general sense means influencing the path choice. It can be considered the same as traffic engineering from the point of view of the network operator, but can also mean giving the end user some level of choice in the path his packets take in the network, not necessarily taking traffic engineering parameters into account. Path programmability can also be achieved by employing software-defined networking. In this paradigm, a controller node exists and has a global view of the network. Since the network is limited to a region, the controller can compute the path traversed by a flow in a centralized fashion and add the corresponding entry on the forwarding table of each switch or router in the path so as to be followed by all packets from the same flow. The path is computed according to specific rules defined by the controller to satisfy requirements in terms of bandwidth and delay per flow, for example [58]. Since path programmability can be based on the utilization of agents, users, or controllers, proposals from this class can be further classified as agent-oriented, user-oriented, or controller-oriented. Although controller-oriented proposals can be programmed by software based on user-level rules, we prefer to analyze them as a separate case given their importance.

G. Internet scalability

Most of the proposals for future Internet interconnection have an impact on the routing table size. Multiple paths require more than one entry per destination; multihoming requires networks to be identified by non-aggregated address ranges; the utilization of flat identifiers, such as used in Loc/ID split and flat routing, makes aggregation of route entries more difficult since the address space may not be hierarchically organized; content-based routing may also add states to routers using its "bread crumbs" approach; and so on. Besides routing table size, another problem that affects scalability is the number of routing control messages sent mainly by edge ASes. These messages are disseminated to all other ASes and the number of them can grow with misconfigurations or malicious actions [43], [44].

The fast growth of routing tables significantly impacts the memory of current routers. This can cause routing table inconsistencies, and even worse, can lead to failures of network equipment [59]. The problem of an exacerbated number of control messages can also lead to traffic overhead and routing table instabilities. Proposals with different levels of complexity exist to reduce routing table sizes. Trivial approaches simply argue that the utilization of disjoint address ranges must be discouraged. More complex approaches can use an idea similar to Loc/ID split, advocating that by removing edge addresses from the core network, we can significantly reduce routing table sizes. Other ideas are concerned with changing the addressing scheme to permit more aggregatable structures. In this case, the IP address can still be used, but a mapping system becomes required. Control messages, on the other hand, can be controlled by using multicast transmissions. The tradeoff, in this case, would be the higher number of states and the higher complexity in both core and edge routers, which is the typical multicast problem. Finally, smarter strategies for storing routing tables were proposed. Memory can be better organized by selecting the most appropriate entries to be cached. This can potentially reduce routing table sizes without losing information. All these proposals will be detailed in the next section.

V. Proposals for the Future Internetworking

This section introduces representative proposals from both competing approaches, namely evolutionary and clean-slate. The first are committed to deploy incremental changes to the Internet, whereas the latter completely disrupt with the original Internet architecture. Independent of the approach, however, the proposals typically address more than one of the challenges described in Section III.

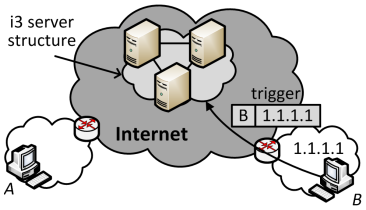
A. Locator-identifier split (Loc/ID Split)

In this section, we further divide Loc/ID split proposals into three subclasses, namely indirect forwarding, network-, and host-based approaches. In addition, we present a mixed proposal combining network- and host-based approaches. We also discuss the main approaches used for mapping systems.

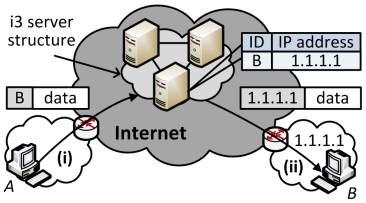
*1) **Indirect forwarding**:* The **Internet Indirection Infrastructure (i3)** [60] is one of the first Loc/ID split proposals. The rationale behind is it to rely on an indirect addressing scheme already used in practice in services such as mobility and multicast, which route packets via home agents and group addresses, respectively. The i3 then proposes an overlay structure for indirection, providing support for services using the same principle.

In i3, a source station sends packets to a destination identifier, instead of to the corresponding IP address. The destination, on the other hand, sends triggers containing its identifier and its IP address to the i3 server structure. Hence, the identifier-to-IP-address mapping from potential receivers is stored in the i3 structure. Before forwarding a packet, the i3 structure replaces the destination identifier with the destination IP address. In Figure 10(a), station B sends a trigger to the i3 server structure. Therefore, upon receiving a packet destined to B coming from A, for instance, i3 forwards the packet to the corresponding destination IP address (Figure 10(b)). Multihoming is supported by using the same identifier associated to different IP addresses or to a network prefix.

Regarding mobility, connections are maintained using the source and the destination identifiers. Nevertheless, if a mobile node changes its access network, it has also to change its IP address and update the i3 structure. Because the connection



(a) Trigger emission. A destination sends a trigger containing its identifier and IP address to the i3 server structure.



(b) Packet forwarding: (i) upon receiving a packet destined to B coming from A, (ii) i3 forwards the packet to the corresponding destination IP address.

Fig. 10. Example of indirect packet forwarding in i3.

is established based on identifiers, it is not reset. Although security was not originally the main focus, efforts have also been made in this direction [61].

*2) **Network-based approaches**:* The **Locator/Id Split Protocol (LISP)** [31], [62], [63] focuses on Internet multihoming, also using indirect routing. Unlike i3, however, LISP divides the address space into local, composed of Internet edge networks; and interdomain, composed of routers in the DFZ. Routers from edge ASes have the interfaces that are connected to the DFZ configured with locators, called Routing LOCators (RLOCs). Stations, on the other hand, are assigned an identifier, called Endpoint IDentifier (EID), with a scope limited to their access networks. As end-to-end packet forwarding includes both EIDs and RLOCs, a mapping system is required. Multihoming becomes easier because one EID can be associated to more than one RLOC or even to a local network address prefix. Figure 11 shows station EID_A using a single locator $RLOC_A$, whereas station EID_B uses two locators, namely $RLOC_{B1}$ and $RLOC_{B2}$, for multihoming.

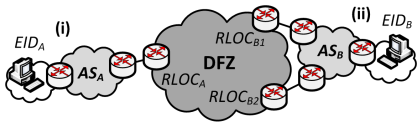
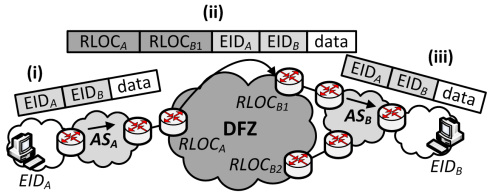
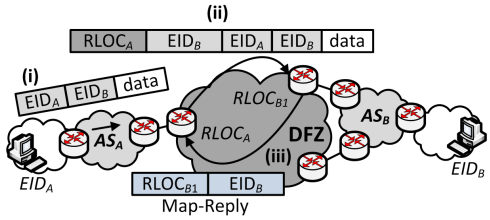


Fig. 11. LISP addressing: (i) EID_A uses a single locator $RLOC_A$, whereas (ii) EID_B uses two locators, $RLOC_A$ and $RLOC_B$, for multihoming.

In LISP, packets are forwarded in the DFZ using RLOCs as if they were their source and destination addresses. Nevertheless, between stations and their respective border routers, source and destination EIDs are used for addressing. Hence, each packet received from the source station at its border router has an extra header added using source and destination RLOCs. This extra header is removed at the destination border router before forwarding the packet to the final destination. The procedure of EID-RLOC mapping and packet encapsulation is called Map & Encap. The whole operation is transparent to access networks and avoids modifications to the Internet structure. Figure 12(a) shows how the LISP packet header changes along the path. If the EID-RLOC mapping for a given destination does not exist, LISP uses the Mapping Distribution Protocol for mapping updates.



(a) Direct forwarding: (i) the packet is sent from EID_A to $RLOC_A$ using stations' addresses; (ii) the packet is encapsulated to be forwarded within the DFZ; (iii) the packet is decapsulated upon leaving the DFZ.



(b) Map-Reply: (i) the packet is sent from EID_A to $RLOC_A$ using stations' addresses; (ii) the packet is encapsulated to be forwarded within the DFZ using the station address as the destination since $RLOC_A$ does not know the correct mapping; (iii) a map-reply is sent back to $RLOC_A$ for the following packets.

Fig. 12. LISP packet forwarding.

LISP assumes for packet forwarding that each source station knows its IP address and the destination EID. Nevertheless, the EID-RLOC mapping procedure depends on the LISP version. In the earliest ones (v.1 and v.1.5), the EID could be routed in the Internet and then, the source border router had to encapsulate the packet using its RLOC and the destination EID as the source and destination addresses, respectively. The packet was forwarded in the DFZ until it reached a border router able to perform the destination EID-RLOC mapping. The corresponding EID-RLOC mapping was sent back as a reply message to the source border router in a Map-Reply, as shown in Figure 12(b). Newer LISP versions no longer consider EIDs as routable in the DFZ. Therefore, a Mapping Distribution Protocol was developed to explicitly request an EID-RLOC mapping (Map-Request).

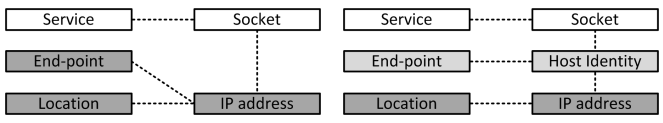
LISP has an emerging variant to support traffic engineering that is also attracting much attention [64], [65]. In the typical approach, each source border router simply tunnels the packet toward the destination border router. Hence, all routing decisions within the DFZ are handled by the underlying routing protocol. In LISP-TE (LISP-Traffic Engineering), in opposition, intermediate routers are chosen by the source border router according to parameters such as path congestion, failure

recovery capability, and multiple shared paths. To this end, LISP-TE introduces an Explicit Locator Path (ELP) locator encoding, which explicitly lists all the intermediate routers. Each one of these routers, called Reencapsulating Tunnel Routers (RTRs), receives a packet and reencapsulates it before sending to the next RTR along the path. These multiple Map & Encap procedures allow defining a complete path from source to destination. Another improvement regards security, which , as well as in i3, has been investigated for LISP [62].

The **Identifier-Locator Network Protocol (ILNP)** [66] splits the IP address into Identifier and Locator. The first is used as a node identity, employed by TCP for connection establishment and also for pseudo-header calculation. The Locator, a network IP address, is used for routing purposes at the network layer. Because the Identifier-Locator mapping is required, the DNS is used for name resolution. As a consequence, it must be aware of all Identifier-Locator mapping in the network.

ILNP is concerned with backward compatibility and, although it defines a new protocol, it does not change packet header formats. Considering its IPv6 implementation, it redesigns the address field, assigning 64 bits for the Identifier and 64 bits for the Locator. In IPv4, the implementation is more complex because of header constraints. Therefore, Identifiers are carried in additional option headers. Independent if using IPv4 pr IPv6, unicast routing can proceed without modifications using the Locator for packet forwarding. Upon reaching the destination network, a cache is used to find the corresponding node. It is worth mentioning that the DNS can be dynamically updated, which is important for mobility and also for provider-independent addresses. Similar to LISP, multihoming can be set by assigning multiple locators to the same identifier. In addition, identifiers can also be encrypted for privacy reasons.

*3) **Host-based approaches**:* The **Host Identity Protocol (HIP)** [67], [68] formally proposes a new architecture for Loc/ID split using the concepts of identity, an abstraction to identify the node; and identifier, a binary sequence used in the identification process. For example, the identity is a station name whereas the identifier is its public cryptographic key. Although any kind of identifier could be used, stations are authenticated using a cryptographic key. Figure 13 shows the current Internet architecture, in which the IP address embodies the dual identifier and locator roles, and the HIP architecture. Note that the same figure could not be directly applied to LISP because there is not an end-to-end identity space.



(a) Address in today's Internet. (b) Address in HIP.

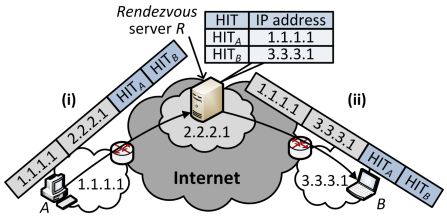
Fig. 13. Different addressing architectures.

Instead of an unformatted identifier, HIP uses the hash of the identifier to make addressing independent of cryptographic algorithms and to use a fixed identifier size. The hash output is called the Host Identity Tag (HIT), which must be globally unique. As HIP is concerned with security from design, a communication starts with authentication and key exchange procedures, using the Base Exchange Protocol (BEX). The authentication involves the exchange of cryptographic challenges, which are updated whenever a cryptographic key expires or one of the stations roams. The transport layer establishes end-to-end connections using the source and destination HITs, provided by the identification layer. Therefore, the identification layer replaces the HIT by the corresponding IP address before sending the packet to the Internet. At the source, this procedure is straightforward but requires a name resolution system to resolve the destination HIT. After obtaining the destination IP address, the communication starts. If the destination roams, the roaming station must inform the other communicating peer about its new IP address and must update the name resolution system.

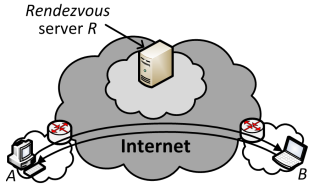
HIP defines a static *rendezvous* server to register mobile stations HITs and their current IP addresses. Once registered, the mobile station records its *rendezvous* server IP address in its DNS entry. Hence, when a source station requests the IP address of the mobile station, it obtains from the DNS the IP address of the *rendezvous* server where the mobile station is registered. With this IP address, the source station starts the authentication and key exchange procedure using the *rendezvous* server as the destination. Upon finding the corresponding HIT-IP address mapping, the server forwards the first packet received to the corresponding mobile station, which responds directly to the source. The communication continues without the *rendezvous* intermediation, which only comes again into scene if a station changes its IP address. Figure 14 illustrates the operation of the *rendezvous* server. In Figure 14(a), the source station A sends the first packet to the mobile station B using the IP address of the *rendezvous* server R as destination. The *rendezvous* server then forwards the packet to the mobile station, which responds directly to A, as seen in Figure 14(b). Although HIP and LISP do not forward packets in the same fashion, HIP can also provide multihoming support because a single HIT can be mapped into multiple IP addresses [69].

Shim6 [32] focuses on providing fault tolerance to multihomed stations in IPv6 domains. Shim6 provides resilient communications in case of locator failures by setting up a state at the end-points for later recovery. Thus, similar to HIP, Shim6 assigns a special name to identifiers, called Upper Layer IDentifiers (ULIDs), and uses the IPv6 address as locator. Unlike previous proposals, the IPv6 address is also used as ULID, not requiring another name space. If a locator changes as a consequence of mobility or a communication failure, the ULID remains invariant whereas a new locator is chosen. As connections are based on ULIDs, locator changes do not result in failures. Renewing locators takes into account available addresses from the multihoming options, which are previously agreed before communication starts between source and destination. Moreover, Shim6 can use hash-based addresses to avoid address spoofing [32].

Currently, a variant of LISP to support node mobility



(a) *Rendezvous* server forwarding: (i) the source station A sends the first packet to the mobile station B using the IP address of the *rendezvous* server R as destination; (ii) the *rendezvous* server R then forwards the packet to the mobile station B.



(b) Direct forwarding: the response for the first packet and all following packets are directly exchanged between source A and destination B.

Fig. 14. Packet forwarding in HIP.

is under standardization. **LISP Mobile Node (LISP-MN)** performs EID-RLOC mapping at the mobile nodes and not at their respective border routers [70]. As a consequence, a roaming node is in charge of updating its EID-RLOC mapping and must update Map-Servers about its new locator. This makes LISP to behave as a host-based approach instead of a network based. LISP-MN maintains established connections because mobile nodes inform their peers about new RLOCs. A drawback is the signaling issues raised to verify whether the mobile node is behind a Network Address Translation (NAT) box.

*4) **Mixed approach**:* The **Global Locator, Local Locator, and Identifier Split (GLI-Split)** [71] framework divides the IP address into an identifier and two independent locators: one to be used within the DFZ and another to be used within a local scope at the network borders. In previous proposals, assuming that each border network receives a set of identifiers with local scope, a roaming user would have to change its identifier when moving to another network. This shortcoming would lead to a connection disruption since end-to-end identifiers are used instead of IP addresses. To circumvent such limitation, GLI-Split adds of a new locator level to yield identifier maintenance even if a node changes to another network. The second locator level is used at the network border to locate a node instead of the identifier.

Similarly to other Loc/ID Split proposals, a communication is handled based on end-to-end identifiers. Therefore, a mapping system is also required to locate the nodes by using an address with topological meaning. The utilization of the two locator levels implies, as a consequence, the use of two mapping system levels: one to resolve the identifier to a DFZ locator and another to resolve the identifier to a network border locator. The DFZ mapping system of GLI-Split is similar to the one used by LISP, whereas the network border mapping system is in charge of only resolving identifiers to locators at the network borders. Note that, at the one hand, identifiers are used to establish end-to-end connections, configuring a host-based approach. At the other hand, a mapping system is still required to deal with the two-locator levels, configuring a network-based approach. This combined configuration gives room to a mixed approach.

GLI-Split supports traffic engineering by selecting the most suitable gateway to the Internet. The multiple gateways can be connected to different ISPs providing differentiated services, chosen accordingly to application requirements or the type of traffic. The multiple gateways can also be used to support multipath forwarding, although it does not mean that GLI-Split neither supports simultaneous transmissions, nor totally disjoint paths.

*5) **Mapping systems**:* Untying identifier from locator requires a mapping system because the Internet still needs topological-based addresses for routing. Therefore, similarly to DNS, which associates names to IP addresses, the Loc/ID split requires a mapping system to associate identifiers to locators [72], [73], [74]. Mapping systems may use PUSH or PULL strategies. In the first one, central elements are in charge of the mapping system, proactively updating border routers [75]. In the second strategy, requests are sent whenever a border router cannot map an identifier into its locator. In this case, a request is sent to central elements, operating in a DNS or a DHT fashion [73], [74], [76]. Hybrid strategies are also possible such as LISP-ALT [77]. In these systems, some central routers exchange information using the PUSH strategy whereas responses to border routers use the PULL strategy.

FIRMS aims at improving the mapping system scalability assuming that a set of EIDs is assigned to a provider via prefix blocks [78]. As a consequence, this provider can participate of the mapping system by resolving its own EIDs. Each provider then sends pointers to its own mapping base to a global entity, which consolidates the same information from all providers before sending a copy to them. Hence, whenever a border router is willing to resolve an EID, it can indirectly look up the EID-RLOC mapping at the mapping base of the prefix owner.

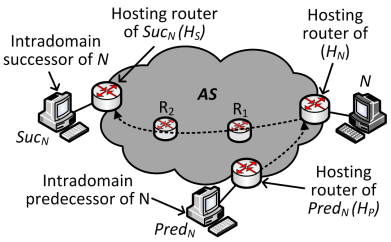
B. Flat routing

In this section, we present ROFL [45], a pioneer work on flat routing for the Internet. Following, we present other flat routing proposals.

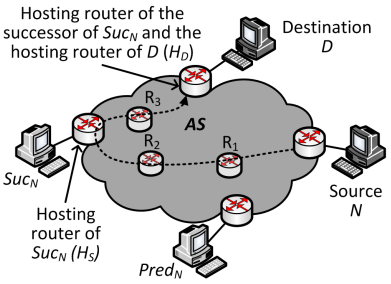
*1) **Routing On Flat Labels (ROFL)**:* It uses labels based on DHTs to identify nodes [45] and uses DHT-based protocols in intradomain (Chord [46]) and in interdomain (Canon [47]).

Similarly to Chord, ROFL defines a circular identifier space where every node has a successor and a predecessor in its intradomain network. Nevertheless, ROFL considers nodes within a local access network, named guest nodes, connected via an interconnection router, called host router. Host routers store the complete path from them to the successors and

predecessors host routers of their guest nodes. Equivalently, routers hosting successors and predecessors also store the complete path in the reverse direction. This path is defined as a hop-by-hop sequence of physical router identifiers, used by source routing. Figure 15(a) illustrates a node N with identifier ID_N hosted by router H_N, the successor of N (Suc_N) with identifier ID_S hosted by H_S, and the predecessor of N ($Pred_N$) with identifier ID_P hosted by H_P. Note that the paths stored for node N start and finish in hosting routers. The paths for the successor and predecessor node are, respectively, $\langle H_N, R_1, R_2, H_S \rangle$ and $\langle H_N, H_P \rangle$.



(a) Successor and predecessor nodes.

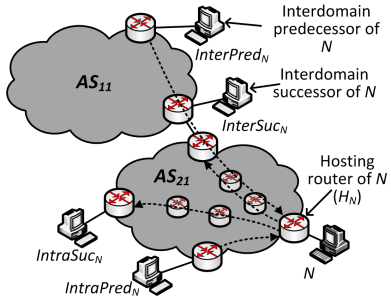


(b) Hop-by-hop packet forwarding from source N to destination D goes through routers hosting all the required successor nodes in the path.

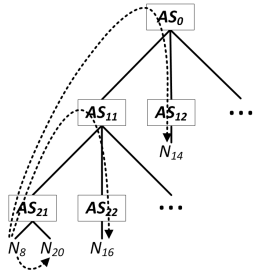
Fig. 15. Intradomain operation of ROFL.

Besides caching pointers to paths connecting their guests to their successors and predecessors, host routers also store pointers to paths that use them as intermediate routers. The first has precedence over the second because the cache has limited size, although the second type of pointers can greatly impact on forwarding performance reducing route stretch. Before forwarding a packet, the host router compares the destination identifier to the identifiers on its table to choose the one nearest to the destination as next hop. Figure 15(b) shows a hop-by-hop packet forwarding from source N to destination D, which goes through routers hosting all the required successor nodes in the path, i.e. H_S and H_D.

Because ROFL considers AS-level policies in interdomain routing, it models the network as a tree topology, where each circular identifier space (an AS) is a vertex, and the edges are the links connecting them. Communications among different ASes are possible after unifying the vertices, which is a three-phase procedure. First, each AS discovers the other immediate upper-level AS with whom it has agreements. Second, similar to Canon, the different ASes recursively unify to the ASes at the same level or below it in the AS tree using routers that share common links. In the last phase, each AS defines pointers to neighboring routers in other ASes to decrease the number of hops. A given node in an AS can be globally reachable if its host router maintains predecessors and successors for its guest nodes in each AS directly connected, as seen in Figure 16(a). Similarly to the intradomain case, in interdomain, ROFL also uses source routing at the AS level.



(a) ROFL successor nodes.



(b) ROFL hierarchy.

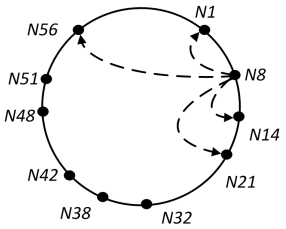
Fig. 16. Interdomain operation of ROFL.

Packet forwarding in ROFL follows an isolation property. Communication between nodes within the same AS does not use pointers to nodes out of it. Similarly, a communication between nodes in different ASes results in packet forwarding through the shortest path in the tree. Hence, the path composed between two ASes must go through the first ascendent AS in common. As observed in Figure 16(b), if N_8 wants to communicate with N_{20}, it will do it without using external pointers. Nevertheless, if N_8 wants to communicate with N_{16}, N_8 has to use its host router to find the successor of N_8 in AS_{11}. Equivalently, a communication between N_8 and N_{14} uses the uppermost AS, demonstrating that even though it uses a flat identification space, scalability and ASes organization are still open issues. The scalability issue with ROFL is related to the amount of pointers stored to other nodes, the control overhead to establish a circular identification space, the latency to join the network, and the failure recovery procedure.

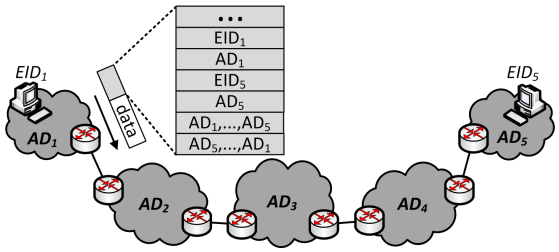
Multipath, path customization, and security are other service requirements supported by ROFL. The first two are handled by end-user negotiation when beginning a session. The destination can append to its response the set of ASes that will be used in the new session. Security, on the other hand, can be implemented by controlling the pointers built to the node and certifying that a given identity is from who it claims to be.

*2) **Other proposals**:* The **Virtual Ring Routing (VRR)** [79] is another protocol that uses a flat identification space, but limits the routing scope to a multihop wireless network [80], [81]. VRR reduces the scalability problem because it deals with fewer nodes than it would find in the Internet. VRR operation is similar to intradomain ROFL, however, each node N maintains a path table to the m closest consecutive neighbors in the circular identification space, half located in clockwise direction and half in counterclockwise direction. Figure 17(a) shows an example where N_8 has 4 neighbors, i.e., $m = 4$. Unlike in ROFL, VRR nodes store consecutive successors and predecessors in their FIBs, and

also use physical information to establish neighborhood. Because VRR was developed for wireless networking, avoiding neighbors connected through low quality links is a desirable feature.



(a) Neighbors in the circular identification space of VRR.



(b) AIP forwarding.

Fig. 17. Other flat routing protocols.

The **Accountable Internet Protocol (AIP)** [82] is another proposal that avoids the use of prefixes and classless interdomain addresses. To this end, it returns to the original address structure of the Internet that concatenates network and station identifiers. This structure has two hierarchical levels, one for routers and another for stations. Unlike ROFL and VRR, which use only one level, in AIP, routers are organized in Accountability Domains (ADs) and stations are identified by Endpoint IDentifiers (EIDs). These two levels increase scalability, but do not completely decouple node identifier from topology locator. One challenge faced by AIP in today's Internet is the correlation between an AS and an accountability domain. ASes identifiers and their announced address prefixes originate from different name spaces. In AIP, the identifier of the accountability domain is used in both cases instead, preventing the best prefix match procedure.

The AIP address structure is a concatenation of the public key hash of the domain identifier (AD) hosting the node and the hash of the station public key. TCP connections are then established using node identifiers, allowing mobility. The single requirement is to guarantee identifier uniqueness, which is obtained by concatenating part of the station MAC address with the node identifier. Packet forwarding in AIP is done using source routing at the domain level because of the non-hierarchical organization of domains. Figure 17(b) shows the header of a packet from EID_1 to EID_5, which includes the source node and its domain identifier, the destination node and its domain identifier, and all the other domains between source and destination in both direct and reverse directions. If an administrator is willing to use multiple paths, he can split an AD into multiple AD-interface combinations and announce them with different routes.

The **Distributed Compact Routing (Disco)** [83], unlike previous protocols, claims that scalable routing on flat names is feasible and the number of bits per network node in routing tables can be logarithmically upper-bounded, instead of linearly lower-bounded as in traditional routing protocols for an increasing number of nodes. In addition, Disco is also concerned with path length. Authors argue that by using DHTs (e.g., as in ROFL and VRR), the forwarding procedure may incur in longer paths even if the destination is physically close to the source. To avoid that, Disco assumes that each source knows the address of the destination and also assumes the presence of special nodes called landmarks, which are randomly selected based on local decision. These nodes play an essential role since all other nodes know the shortest path to them.

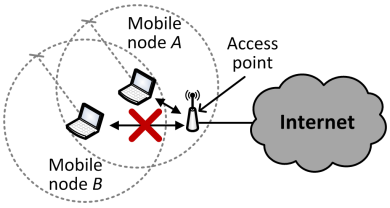
Nodes obtain the notion of vicinity and landmarks based on a standard path-vector routing protocol. Thus, each node stores shortest path information about a given number of closest neighbors and about all network landmarks. Because the node address is a combination of the identifier of the closest landmark and the labels of all nodes within the shortest path between them, the address is used for routing. Therefore, unless the destination is within the source node vicinity or is one of the landmarks, the path chosen goes through the closest landmark. Obtaining information of such landmark is straightforward because Disco assumes that all nodes know the destination address. Out of it, they can extract the identifier of the closest landmark to the destination, which is the solely information missing for packet forwarding since the path toward any landmark is already known.

C. Network mobility

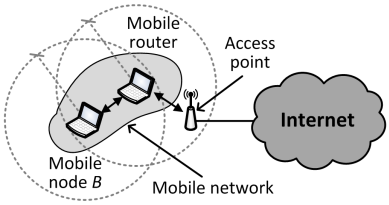
In this section, we describe NEMO [30] and its enhancements.

*1) **NEtwork MObility (NEMO)**:* In this architecture [30], each network selects one mobile station to operate as a Mobile Router (MR) to provide backhaul access to other stations. Figure 18(a) shows the infrastructure mode, where only nodes directly connected to an access point can have Internet access, whereas Figure 18(b) shows the NEMO basic operation, where mobile node A plays the role of a network MR, providing backhaul access to station B.

NEMO is subject of an IETF working group [84]. Its basic architecture concentrates into the MR all the operations executed by stations running legacy Mobile IP. Updates, such as a new IP address (CoA) obtained from a foreign network and the consequent tunnel establishment, become transparent to all other mobile nodes. In the NEMO architecture, a MR also sends update messages to its home agent, which associates the MR IP address in the foreign network (CoA) to a network prefix, unlike Mobile IP. Thus, all packets destined to one of the mobile network nodes are encapsulated and tunneled by the home agent to the MR. The MR decapsulates the packets received and forwards them to the destination within the mobile network. In addition, the MR forwards all the packets



(a) Classic Mobile IP mode: nodes out of the range of the access point cannot have network access.



(b) Network mobility: nodes can access the Internet even out of the range of the access point via an intermediate wireless node.

Fig. 18. Connectivity in mobile wireless networks.

originated in its mobile network to its home agent, or sends these packets directly to the Internet if the foreign network does not perform egress filtering. Figure 19 illustrates the path followed by packets from corresponding node C to mobile node N using NEMO. The first protocol implementation following the NEMO architecture was called NEMO Basic Support Protocol (NEMO BS), which is preferably used over IPv6 because of its routing header extension [85]. NEMO BS still has to deal with challenges regarding security. For instance, the source address of a tunneled packet must be checked by the MR and by its home agent to certify its authenticity.

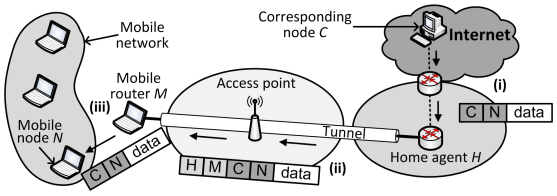


Fig. 19. NEMO packet forwarding: (i) the corresponding node C sends a packet to the mobile station N using the IP address of N in its home network; (ii) the home agent H encapsulates the packet and sends it to the network visited by N; (iii) the mobile router M receives the packet and, similarly to a foreign agent, decapsulates the packet and sends it to N.

As already mentioned, if a mobile network node wants to communicate to an Internet node, packets may be first sent to the home agent. The home agent then forwards those packets to the Internet, making routing inefficient. This problem is more severe if we consider that a single MR can have associated to it several mobile networks. Fundamentally, coexistent mobile networks are possible and, as a consequence, a MR can use another one from a different mobile network to reach the Internet. In this case, NEMO is known as Nested NEMO [86] and the communications can be very inefficient considering that all traffic from a mobile router must be first sent to its home agent (Figure 20).

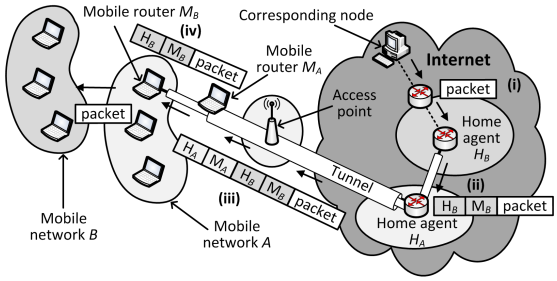


Fig. 20. Nested NEMO problem. Mobile networks have Internet access via nested mobile routers: (i) the corresponding node sends a packet to a station within mobile network B; (ii) the home agent H_B of the mobile router M_B encapsulates the packet, as well as (iii) the home agent H_A of the mobile router M_A; (iv) the packet starts getting decapsulated upon arriving at the mobile routers.

*2) **NEMO enhancements**:* **NEMO+** [86] proposes three protocols to enhance communications. The Tree Discovery (TD) protocol assists MRs to choose, in their neighborhood, the MR that has the best path to the Internet. TD uses the IPv6 neighboring discovery to obtain complete paths used by other nodes to reach the Internet. These paths are obtained through ICMPv6 messages and are represented as tree branches starting at the Internet gateway (root) and finishing at the MRs (leafs). Upon receiving such announcements, the MR chooses the best tree branch available. The Network In Node Advertisement (NINA) protocol announces to the MRs located near an interconnection point, the prefixes of the subnetworks associated to each MR. These announcements allow communications between mobile nodes from different networks without the participation of home agents, but connected to the same tree. Figure 21 shows the path taken by packets from node B in the mobile network B to node A in the mobile network A. The solid and the dotted lines show the path with and without NINA, respectively. The last protocol, called Reverse Routing Header (RRH), improves packet forwarding to the Internet by always using the last MR IP address as the source address of all packets. The previous source IP address is stored in a list in the packet header, which contains the IP addresses of all previous MRs that have forwarded the packet. As the source IP address received by the interconnection point is the IP address of the last MR, the packet is only forwarded to the home agent of this last router before being sent to the Internet. The list of all IP addresses of the MRs is added by the home agent in the reverse direction, when it receives the packet from the Internet. Hence, the packet is forwarded to the originating mobile node.

Light-NEMO+ [87] uses an idea similar to RRH. In Light-NEMO+, the top level MR stores an identification (called cookie) of the originating node CoA and forwards the packet to its corresponding node in the Internet. The corresponding node sends back the same cookie, which allows the top level MR to map the communication without using source routing.

The **MAnet NEMO (MANEMO)** [88] protocol aims at guaranteeing that an ad hoc mobile node will always be

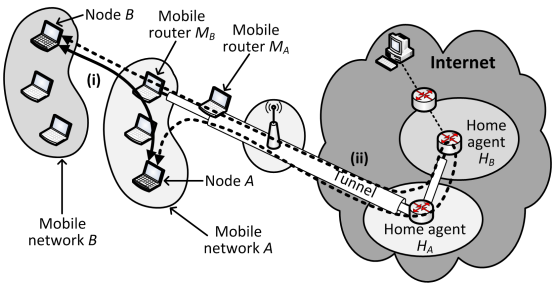


Fig. 21. Impact of NINA utilization: (i) packets are sent without passing through mobile routers' home agents; (ii) inefficient path passing through mobile routers' home agents.

reachable from the Internet. The MR running the MANEMO protocol has one network interface configured according to the foreign network and another configured according to the ad hoc network it belongs to. The latter interface runs an ad hoc routing protocol, e.g., OLSR [89], configured to announce itself as an Internet gateway. The Unified MANEMO Architecture (UMA) [30] proposes the unification of all NEMO protocols to guarantee interoperability. UMA also defines how the different mobile nodes must connect to the Internet, either via direct access to interconnection points or other mobile networks, and how tunnels between different home agents must be established.

Since NEMO BS, security includes authentication of MRs before binding them to a network prefix. Security can be further enhanced by using IPSec. In addition, NEMO can deal with multihoming, assigning to a single MR multiple home agents. Nevertheless, security and multihoming are considered fragile and cumbersome because of the broadcast nature of the wireless medium and all the overhead involved. Therefore, proposals to unify NEMO to HIP are emerging to improve security and also mobility and multihoming [90]. As mentioned before, HIP uses authentication procedures before establishing connections, which is a step ahead toward secure communications. Other proposals also tackle these issues by providing mechanisms to maintain sessions even after a network roams (e.g., SeNEMO [91]).

D. Multiple paths

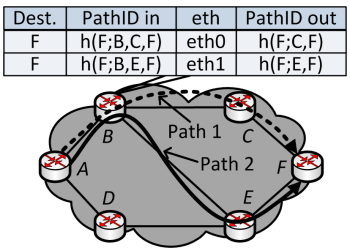
This section presents proposals for multipath routing based on source routing, overlay, and tunneling approaches. We focus on transit and core level proposals, although we also present multipath forwarding at the access network level when the proposal also handles this case.

*1) **Source routing**:* The **BANANAS framework** [92] uses source routing for multipath forwarding, assigning an identifier called PathID to each path. A PathID is composed by the hash of all vertices and link identifiers in a path between any pair of nodes. The vertices identifiers, e.g., IP addresses, as well as the link and AS identifiers are globally known and so is the PathID. The concept of "globally known" depends on the scope: for intradomain, it means knowing vertices and links identifiers; whereas for interdomain, it means knowing AS identifiers. In the intradomain case, the global knowledge is obtained by link-state routing protocols; whereas in the interdomain, it is obtained by BGP path vectors. To avoid collisions among PathIDs, each one is extended by a tuple composed of the destination IP address and the hash result. PathIDs are added to all forwarded packets.

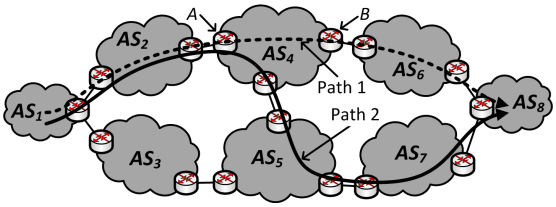
BANANAS is incrementally deployable. To this end, routing computation is performed only considering the routers running BANANAS. In intradomain, the source router sends packets through multiple paths, as depicted in Figure 22(a), which also shows the forwarding table of router B. Packets carry the PathID computed by hash functions, denoted by $h(.)$ in the figure, according to the path followed. An intermediate router uses the tuple destination address prefix, input PathID, outgoing interface, and output PathID for packet forwarding, instead of the tuple destination address prefix, next hop, and outgoing interface. The input PathID is a hash of all router identifiers from the current router to the destination whereas the output PathID is the hash of all identifiers from the next hop router to the destination. Upon receiving a packet, a router looks up the corresponding entry in its FIB based on the destination prefix address and input PathID. Before forwarding the packet, the router replaces the packet PathID with the output PathID, procedure repeated by every BANANAS router in the path. Every router executes an algorithm to compute multiple paths since the framework was proposed with this goal. Even the intermediate routers must know all possible paths from itself to all the possible destinations in the network to forward packets according to the path chosen at the source. The side-effect is the number of routing entries stored, which leads to scalability issues, but these problems are tackled by the use of coding techniques to compact information [92].

The interdomain routing is an abstraction of the intradomain version but using ASes. The role played by the ingress and egress AS border routers are equivalent to the incoming and outgoing router interfaces in the intradomain. An e-PathID is added to packets, where AS numbers are used as identifiers. An AS ingress border router examines the destination address prefix of the received packets and forwards them to the corresponding egress border router. Thus, the IP address of the egress border router is used as the packet destination address and the original one is stored in a stack, where the IP address of the egress border router is at the top. The egress border router pops its address from the stack and changes the destination address to the original one. The ingress border router in the next AS repeats the same procedure and so on. The PathID used inside each AS depends on the intradomain path chosen. On the other hand, the utilization of multiple interdomain paths must deal with issues related to AS-level agreements. Figure 22(b) illustrates multiple interdomain paths between sources in AS_1 and destinations in AS_8. Routers A and B represent ingress and egress border routers, respectively, of AS_4 concerning the flow in Path 1. Thus, a packet following Path 1 uses the address of router B as destination IP address after being forwarded by router A.

Path splicing is another possibility to use alternative paths that builds multiple shortest-path trees from the source to all other network destinations [93]. To find different trees, controlled link weight perturbations are added to slightly



(a) Intradomain routing. The sequence of routers is chosen at the source and is followed by each intermediate router using the PathID.



(b) Interdomain routing.

Fig. 22. Packet forwarding in BANANAS. Sequence of ASes is chosen at the source and is followed according to the e-PathID.

change the network topology. From each modified topology a new tree maximizing the disjunction of paths to the same destination is computed, without considerably affecting path length. Whenever the shortest-path algorithm is run, a new FIB is built. In the end, considering n computed FIBs, each destination becomes reachable by n different entries. Each tree is named a slice, and if an intermediate router decides to change the slice followed by a packet, it splices the path with another slice. An intermediate router can decide then whether the packet will be forwarded through the same slice. Alternatively, the packet can carry on its header a sequence of bits (splice bits) to determine the slices followed hop-by-hop. These bits are used to index the corresponding FIB containing the chosen next hop. Because these bits are selected at the source, the user can influence routing decisions. If a router does not recognize the splice bits, it can send the packet using a default route. Path splicing can be used for intra and interdomain routing. The difference is where a splice takes place, if it is at the router or AS level. Considering the intradomain case, Figure 23 illustrates two slices computed by router A. Slices 1 and 2 are represented by two different dotted lines. Upon receiving a packet from A to F, node B looks up the next hop at the different FIBs. Node B then switches the path followed by the packet from Slice 1 to 2, as shown by the solid arrow. Path splicing improves failure recovery at the cost of adding more states per router.

Slick packets are used by a stateless protocol that relies on source routing to improve failure recovery without adding states per routers [94]. These packets avoid large packet headers and strict path definition which come along with source routing. Slick packets add to the packet header an encoded directed acyclic graph (forwarding subgraph) toward the required destination. The encoded subgraph avoids large packet headers and, upon receiving a packet, an intermediate

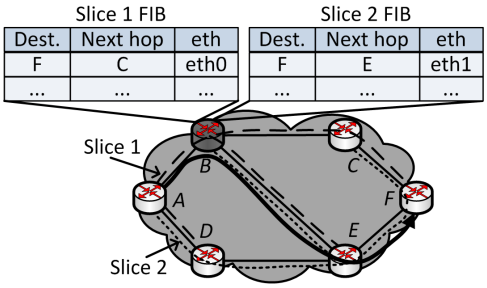


Fig. 23. Intradomain path splicing from Slice 1 to 2 at node B. The two different dotted lines show the two slices.

router can choose the next hop among its neighbors according to the received forwarding subgraph, further avoiding strict routing. A preferential path can be established at the source, but one of the alternative paths can be chosen on the fly, if needed. Hence, if a failure occurs, it is neither needed to go back to the source node nor to store additional paths into the network. Based on the forwarding subgraph, an intermediate router can locally take a countermeasure to circumvent a link failure, increasing network responsiveness.

*2) **Overlay networking and tunneling**:* The **Resilient Overlay Network (RON)** [95] architecture, initially proposed for failure recovery, can also be used for multipath forwarding. Nevertheless, one of its main drawbacks is the reduction of the control by the transit ASes over the chosen paths, which is also a side-effect of source routing.

The **Multipath Interdomain ROuting (MIRO)** [96] protocol neither uses overlay networking nor source routing. MIRO is based on negotiation between neighboring ASes for multipath utilization, improving forwarding performance and security, if a next hop AS is not performing as expected. MIRO assumes that although multiple paths likely exist, each AS only announces the most convenient path because of adopted policies, protocol implementation, and scalability. Nevertheless, MIRO argues that a given AS may ask for multiple paths, as long as it is interested in. Hence, scalability issues can be controlled and the implementation throughout the Internet can be incremental. An AS not using MIRO simply does not respond to alternative path requests. Figure 24 shows the negotiation between routers A and B for alternative paths. Router A belongs to AS_2 and does not want to forward its traffic through the default path announced by AS_4, which is through AS_6. AS_4 offers an alternative path through AS_5, which is accepted by AS_2. ASes can send requests to more than one neighbor to obtain other alternative paths or they can proactively announce a known alternative path, if it suits a received request.

In MIRO, the paths discovered by regular BGP are still used and the packets sent through alternative paths are tunneled [96] between neighboring ASes to guarantee the utilization of the alternative path. The tunnel identifier is sent by the AS closest to the destination to the AS closest to the source after the negotiation procedure. Figure 24 shows a tunnel established between routers A and B to forward packets through the alternative path. The tunnel ID is used to identify

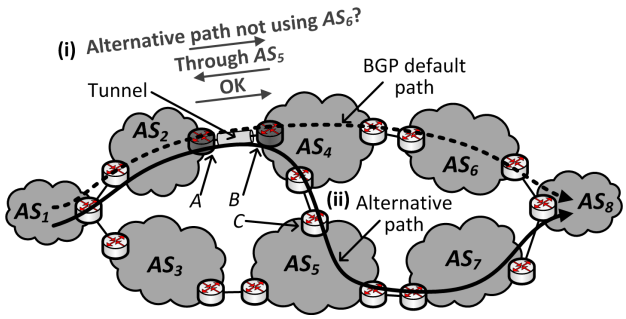


Fig. 24. MIRO operation: (i) negotiation to discover alternative paths; (ii) alternative path utilization.

the alternative path via AS_5.

MIRO uses a multihop approach, since it selects or avoids specific intermediate routers or paths. Such selections, as well as the utilization of other techniques, can be individually set without taking into account their impact on other ISPs. The unilaterally decision of all ISPs can destructively contribute to the overall performance. Therefore, coordinated efforts are important to allow ISPs to explicitly exchange information about their routing decisions. Hence, upon negotiation, they can agree on better choices toward overall performance gains [97], [98].

Tunneling procedures can also be used to forward packets via deflection points [99], [34], which are routers to whom the path is deviated. Figure 24 shows router C being used as a deflection point (conventional BGP path goes through router B in AS_4). In this case, however, the tunnel would be established until router C. After the deflection point, the traffic follows the conventional BGP path assuming there is no other alternative. Deflection points can also be used in overlay networks. In extreme cases, a user station in another AS could participate of the routing and this station can be the deflection point. The user participation impacts network scalability but avoids network modifications. The deflection point can be chosen by the source node, but independently of the situation, the tunnel end-point must be aware of the ongoing procedure to forward the traffic to the real final destination. Although MIRO uses tunneling procedures and can involve end users as deflection points, running the protocol at the AS-level is an attempt to guarantee scalability at the cost of fine-grained topology visibility.

E. Content-based routing

In this section, we describe two different subclasses of CBN architectures, namely structured and non-structured [100].

*1) **Structured architecture**:* The **Data-Oriented Network Architecture (DONA)** employs flat naming and hierarchical routing [52]. DONA employs flat and self-certifying names generated by a principal. This entity is associated with public-private key pairs, which are used to identify contents. Names are formed by a pair $P : L$, where P is the hash of the principal's public key and L is a label randomly chosen by the principal, which ensures name uniqueness in its domain. A principal P acts as a content publisher and administrator because only nodes authorized by P are able to provide access to contents named as $P : L$.

Upon requesting content $P : L$, users receive the corresponding data, public key P, label L, metadata, and a signature of the requested content [53]. Upon receiving this information, they can verify the authenticity of the publisher by checking if the hash of the public key sent is P and also that P was used to sign the content received. Thus, DONA provides content authentication by protecting the content and not the connections over which the content is sent. DONA also assumes that users are able to find names by using external mechanisms, such as search engines.

The main disadvantage of flat naming is the lack of name aggregation, which may compromise scalability. To avoid this problem, DONA employs a hierarchical tree algorithm that reduces the amount of information routers have to maintain and also reduces the control overhead. In DONA, routers are referred to as Register Handlers (RHs), which are organized in different hierarchical levels called tiers, as shown in Figure 25. The root node is placed in Tier-1 and RHs in the same level are called peers. RHs have explicit routes to peers and thus a RH is able to forward content requests to nodes in the same level directly. On the other hand, if a RH does not know a route to a given content, it forwards the request to its immediate upper-level node in the tree. Thus, RHs do not have to maintain a complete view of the network topology. A RH only maintains routing information about its descendants.

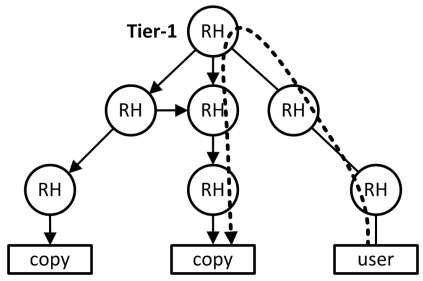


Fig. 25. An example of the hierarchical tree used by DONA (Adapted from [52]).

Routing in DONA is based on FIND and REGISTER messages. Users send FIND($P : L$) messages to the local RH to find a given $P : L$ content. The local RH then forwards these messages to nearby peers that maintain a copy of the requested content. If there is no peer that has a copy, the local RH forwards the content request to the RH above it in the tree. REGISTER(P:L) messages are sent by nodes that have copies of a content and thus these messages establish the state needed by RHs to efficiently forward FIND messages. Figure 25 shows the registration state, represented by solid arrows, in RHs after copies have registered themselves. In this example, the dashed arrow represents the route traversed by the FIND message sent by the user from the local RH to a nearby RH that maintains a copy of the content.

DONA also introduces caching mechanisms in RHs to increase content availability and QoS experienced by users. On the other hand, as mentioned, DONA uses non-aggregatable

identifiers that results in one entry per content in routing tables, which may compromise routing scalability. DONA can be incrementally deployed over the current IP layer, considering the present addressing and routing mechanisms [52]. FIND messages, for example, are inserted as a shim layer between IP and transport headers. These messages resolve names and also initiate the transport exchange. For that matter, transport protocols should bind to names instead of addresses without complex changes. DONA does not require modifications of the IP infrastructure to exchange messages after a FIND has been received, because these messages are not handled by RHs. Messages are forwarded to the destination based on the current IP routing and forwarding mechanisms. Multihoming and mobility support are also natively provided. A multihomed node can send FIND messages to more than one local RH and thus multiple paths are defined to receive the content. Mobility support is assured by the content register protocol that is based on REGISTER and UNREGISTER messages. Before the location changes, a publisher sends an UNREGISTER message to its current local RH. In the new location, the publisher only has to send a REGISTER message to its local RH to make the content available. All FIND messages can be then forwarded to the new location.

The **Network of Information (NetInf)** [55] is another CBN architecture that employs flat naming and structured routing based on DHTs. Similarly to DONA, NetInf considers names composed by the concatenation of the hash of the content owner's public key, P, and a label chosen by the owner L. NetInf also proposes to store a digital signature in meta-data to allow content integrity verification. In addition, owner authenticity and identification is determined from public key chaining information stored in meta-data.

NetInf employs a hierarchical name resolution service called Multi-level Distributed Hash Table (MDHT) [101] to improve the scalability of existing name resolution systems. With MDHT, contents are retrieved based on a two-step process similarly to other name resolution systems. First, in the resolution phase, name is resolved into a list of locators that indicates copies of the desired content. In the next step, called content forwarding phase, a set of locators is selected from the list according to network conditions, for example. After that, the content is sent by one or multiple sources to the requester, which allows multipath communications.

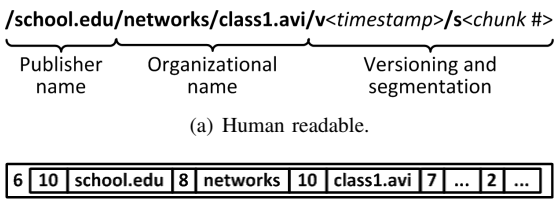
The resolution phase is the core of the MDHT system. To store name-locator bindings, MDHT employs multiple interconnected DHTs (called DHT areas), which represent a network on a different topological level. Basically, three levels are considered: the AS, the Point of Presence (POP), and the Access Node (AN). These levels are arranged to reflect the underlying network topology. For example, in the AN level, nodes work similarly to a local DNS server, receiving name requests directly from clients.

In NetInf, routing and forwarding requests are also organized in different levels. On the first one, intra-area forwarding is performed by using the DHT algorithm adopted in the area. In the second level, inter-area forwarding is performed through MDHT nodes in both levels. The authors argue that the nested hierarchical approach simplifies the deployment of the MDHT system because it can be implemented in small networks and, after that, be interconnected to build a larger system.

There are also additional proposals for specific scenarios relying on the structured architecture. For instance, SeDAX (Secure Data-centric Application eXtension platform for smart grid applications) [102] employs content-based routing to satisfy scalability and security requirements of smart grids [103].

*2) **Non-structured architecture**:* The **Content-Centric Networking (CCN)** employs hierarchical naming and non-structured routing to provide content distribution [48]. The main characteristic of CCN is to divide content in chunks that are uniquely identified and individually requested.

The CCN naming approach provides names that have hierarchical semantics and can be aggregated. In addition, CCN aims at using names partially meaningful to humans. For that, each name is a set of components that indicate content location and properties, and is also represented as an URI (Uniform Resource Identifier) for convenience. Each component is formed by an arbitrary number of octets, leading names to have variable length. Figure 26 shows the human-readable representation and the binary encoding of a given content name. Components are separated by slashes and, in this case, represent the publisher name, the organizational content name, version, and chunk numbers. Figure 27 shows the name tree derived from the example illustrated in Figure 26. In this example, a user requesting the most recent version of the content indicates the interest in "`/school.edu/networks/class1.avi` *RightmostChild*". Thus, the first chunk of the most recent version of the content is retrieved, as indicated in the figure. The following chunks can be explicitly requested (s1, s2, and s3) or by using the *LeftmostRightSibling* annotation. Names in CCN, however, are not highly persistent because if the content publisher changes, the content name must also be changed.



(a) Human readable.

(b) Binary encoding.

Fig. 26. An example of content name.

CCN routing employs only two packets: interest and data. Users request contents by broadcasting interests over the Internet. Any node storing data that satisfies an interest received can respond by sending a data packet. This packet is sent to the requesting node over the reverse path traversed by the interest. This path is defined based on "bread crumbs" left behind by interests on their way toward sources on intermediate nodes. Thus, only interest packets are routed in CCN. Both interest and data packets have a field that carries the content name, which is used to verify if a data packet satisfies a given interest.

A CCN router works similarly to an IP router. For each incoming packet, a best prefix match lookup is performed on its content name and, based on the result, an action is done.

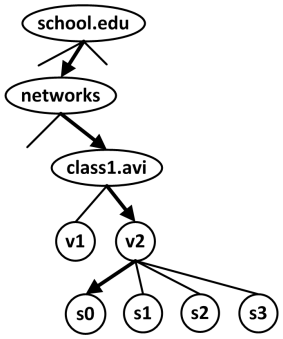


Fig. 27. An example of name tree.

Figure 28 illustrates CCN routers, which are composed of the FIB, the Content Store (CS), and the Pending Interest Table (PIT).

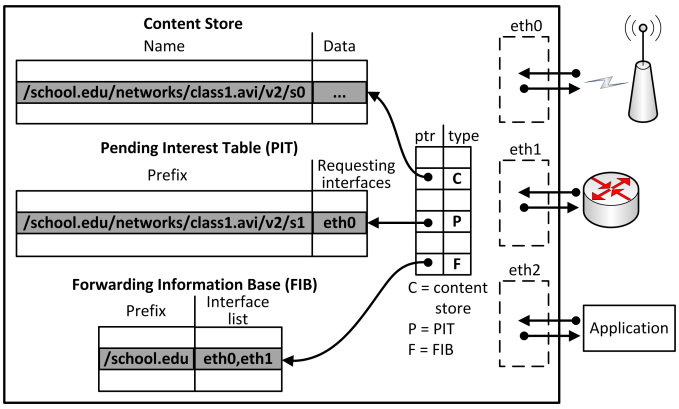


Fig. 28. The CCN router model (Adapted from [48]).

The CCN FIB is used to perform interest forwarding toward potential sources of the desired data but, rather than IP FIB, it defines a list of outgoing interfaces for each prefix. This is needed because CCN allows multiple sources per data and thus nodes must be able to send queries to all of them. The Content Store (CS), on the other hand, is similar to the buffer of an IP router but it must be able to maintain states about the arriving data packets as long as possible. Each packet in CCN is potentially useful to many users that are interested in the same content. Hence, unlike an IP router that discards a packet as soon as it is forwarded, CS employs different replacement policies such as Least Recently Used (LRU) and Least Frequently Used (LFU). Finally, PIT keeps track of each interest packet forwarded to content sources by registering its input interface. When an interest packet arrives to a router, it first verifies if there is already an entry for the name carried by this packet in the CS. If there is, the corresponding data packet is sent through the input interface of the interest packet. Otherwise, the router verifies if there is already an entry for this interest packet in PIT. In affirmative case, the input interface of the interest packet is added to the list of interfaces that are also waiting for this content and the interest packet is dropped. If there is no entry in the PIT, the interest packet is forwarded based on FIB rules and a new PIT entry is created with the input interface of this packet. Thus, interest packets leave "bread crumbs" (i.e., PIT entries) on routers as these packets are forwarded to sources. Routers are then able to forward back data packets to the clients. During the reverse forwarding, a router only admits a data packet if it has a valid entry for the content that the packet carries in its PIT. Otherwise, this data packet is discarded.

As routing in CCN is not based on the location of users, multihoming is also supported. In addition, multipath is an intrinsic characteristic of CCN because interests are sent through multiple interfaces and data chunks can be simultaneously received from multiple sources. Scalability is a shortcoming of content-based routing, which is already tackled by aggregating names in CCN. Some studies argue that current routers are able to implement in-network caching at CDN scale [104]. Nevertheless, authors claim that content-based routing can use legacy protocols and can also execute on top of IP layer. This possibilities permit the architecture to be considered incrementally deployable [48]. In contrast to DONA, CCN does not use self-certifying names. Nevertheless, it authenticates the association between names and content by signing each data packet and also allows the encryption of private content.

Non-structured architectures are also employed in specific scenarios. For example, CRoWN (Content-centRic frameWork for vehicular ad-hoc Networks) considers content-based routing as an alternative to provide efficient communication in vehicular networks [105].

F. Programmable paths

In this section, we subdivide the approaches into agent-, user-, and controller-oriented proposals. Although the first one does not have any representative candidate to the best of our knowledge, the basic concept is introduced. User- and controller-oriented proposals, in opposition, have been addressed in the literature. Because all proposals may require in-network measurements, we also dedicate a section to this topic.

*1) **Agent-oriented approach**:* These proposals are in opposite direction to the end-to-end Internet principle. The characteristic of intelligence at the edges, on the one hand, increases network simplicity and flexibility [1]. On the other hand, it complicates fault diagnosis and demands much manual configuration. This last consequence reduces network performance because it possibly results in misconfigurations and in higher delays for network recovery. Although intelligent network agents reduce human intervention, their employment must not harm one of the main Internet pillars which is core simplicity [57]. Therefore, there is a clear tradeoff, which must be balanced. The main role of the Internet agents is to improve the network autonomy, making it more independent of manual configurations. Autonomous networking, on the other hand, uses techniques for context recognition to perform tasks faster. These techniques require a high-level view of the network goals and the knowledge of their limitations to making decisions. Finally, agents must report their performance to users and network administrators with high level interfaces.

The knowledge of the current traffic status and of the users' requirements also permits agents to make decisions at the

routing layer. Such decisions are not necessarily governed by deterministic algorithms because the environment is highly dynamic, prone to conflicts of interests, and full of uncertainties. It is important to keep a database that builds, reconciles, and maintains the many aspects of the Internet behavior in a high-level view. This database can be stored in a plane called the Knowledge Plane as defined by Clark *et al.* [57]. The Knowledge Plane must be able to provide information such as the best configuration parameters given a certain scenario to other network elements. In the routing case, the Knowledge Plane is capable of changing or establishing paths according to application requirements. One challenge is to estimate how much information the Knowledge Plane must maintain and how broad the network view must be. Especially in routing, designing a Knowledge Plane with Internet global view is not feasible given the amount of information needed. Then, this plane must maintain the most useful information in a given circumstance and must use scalable distributed techniques to filter observations according to users' interests.

*2) **User-oriented approach**:* The **New Internet Routing Architecture (NIRA)** [106], [107] is a user-oriented proposal that aims at stimulating ISP competition. NIRA proposes empowering users with the opportunity to choose paths at the AS level. Hence, the competition among ISPs can culminate in lower Internet access fees as well as stimulating novel added-value services. Among them, one could think of a network service to support end-to-end quality of service. The simplest manner for users to define the path traversed by their packets is by using source routing. NIRA uses a scheme that does not explicitly insert the path into packet headers to avoid enlarging their sizes. In NIRA, the address of a station is a concatenation of its identifier with all ASes identifiers along the way to the Internet core. The part of the address identifying the ASes must be a concatenation of all domain prefixes in the path toward the core. Figure 29 assumes there is only one domain between the access network and the Internet core. In addition, let the identifier of the core network to be also 1.1 (1.1.0.0/16), the identifier of the intermediate domain to be 1 (1.1.1.0/24), and the identifier of station A to be 1. Hence, the concatenation of all identifiers becomes 1.1.1.1, which is the address of station A. As a consequence, in an end-to-end communication, packet source and destination addresses are enough to identify the whole path. Only core ASes need to run a routing protocol to forward the received packets. In Figure 29, the routing protocol selects the best path to be followed by packets from router R_A to R_B, for example. Because the number of ASes in the core is small compared with the number of ASes in the entire Internet, this protocol does not face scalability issues. Although we use 32-bit addresses in the example of Figure 29, NIRA does not define a fixed address format since this does not affect its operation.

NIRA packet forwarding is based on both source and destination addresses and also depends on users' choices for route discovery. The first feature contrasts to IP forwarding, which is solely based on destination addresses. The second feature, however, is supported by using two path discovery protocols: the Topology Information Propagation Protocol (TIPP) and the

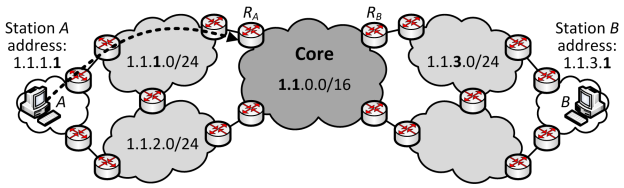


Fig. 29. NIRA addressing structure.

Name-to-Route Resolution Service (NRRS). The first protocol sends users information regarding the path from the user station to the core AS, taking into account possible AS agreements. In Figure 29, station A could choose between the IP address 1.1.1.1 and 1.1.2.1. The NRRS protocol, on the other hand, defines a name resolution service to provide stations the address of the destination. Hence, station A discovers the address of station B, 1.1.3.1 (Figure 29). The source and the destination addresses are cached to avoid consecutive searches.

Pathlet Routing [108] is another user-oriented proposal, in which ISPs announce incomplete paths (Pathlets) which can be concatenated by users. This procedure results in end-to-end paths, each one defined as a sequence of virtual nodes. Such nodes can be associated to all routes known by all network routers, or in the simplest case, to only the routes known by a single router. The path of a packet is defined in the source packet header, where each Pathlet is identified by a Forwarding IDentifier (FID) with scope limited to the Pathlet source virtual node. Once the route is established at the source, subsequent routers only change the sequence of identifiers. Figure 30 illustrates these modifications. Each dotted arrow represents a Pathlet and the rectangles below each router represent the Pathlet identifiers within the header of the packet in that router. The packet is originated at router A and destined to router E. Note that at each hop the traversed Pathlets are removed from the header (e.g., between routers A and B) and new Pathlets are inserted in the packet header if a traversed Pathlet is composed of multiple hops (e.g., between B and C since Pathlet 2 is composed of Pathlets 7 and 1).

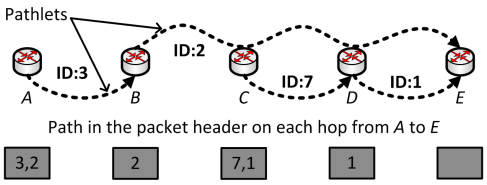


Fig. 30. Pathlet Routing.

In Pathlet Routing, a given node disseminates the Pathlets it knows, considering its own polices. Pathlets are disseminated such as BGP path vectors, which whenever received by users, allow them to choose their paths. In Figure 30, the path from router A requires five IP addresses, including source and destination, whereas the utilization of Pathlet requires at most two identifiers (FIDs). Pathlet Routing can also emulate MIRO [96] by accommodating two pathlets in a single route of a packet.

An important question is how to encourage ISPs to provide users the power to choose paths. The current Internet model is based on commercial agreements between providers where the end user has no influence. Changing the model, however, must be attractive also for ISPs because it comprises agreements between users and access providers, or between users and a subset of ISPs that could offer connectivity up to the Internet core. In the first option, providers directly connected to users make agreements with neighbors to offer different path possibilities depending on users' choices. In the second option, users have the possibility to choose all ASes in the path, which offers them more freedom. Nevertheless, scalability problems arise as each service provider may follow specific policies with the different neighbors.

*3) **Controller-oriented approach**:* The **OpenFlow protocol** [40] enables software-defined networking by providing a programming interface for switches. A controller configures the forwarding elements, which are called OpenFlow switches, to define the networking actions for a given set of packets. An OpenFlow switch is composed of a flow table and a secure channel, as illustrated in Figure 31. The secure channel is used for communications between controller and switches. Thus, the controller can manage a switch by sending messages to either modify its flow table or to acquire network statistics and equipment information. The flow table is a set of entries, where each entry is defined by a number of header fields and a set of associated actions and counters. Whenever a packet arrives at a given switch, it is compared against a set of flow entries. If there is a match, the actions defined for that entry are performed on this packet. Otherwise, the packet is forwarded to the controller through the secure channel to have the set of actions defined. Basically, the controller defines a path for the packet and adds the corresponding flow entry in all the switches along that path. OpenFlow provides methods to automatically compute paths based on bandwidth and delay requirements, for example. The following packets of the same flow, considering a flow as a sequence of packets matching the same flow entry, will no longer need to be forwarded to the controller.

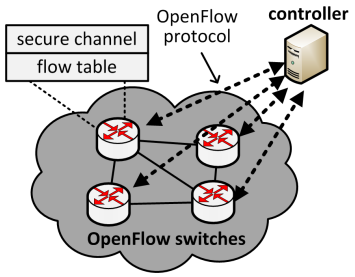


Fig. 31. An OpenFlow network: the controller manages flow tables over the secure channel.

Clearly, there is a tradeoff between simple management and scalability with OpenFlow. One node, the controller, has to program paths for all flows in a centralized fashion and has to set these paths on all switches. In this case, on the one hand, path programmability does not depend on specific vendor equipment. On the other hand, the controller must deal with traffic from all flows and can become a single point of failure.

*4) **Data acquisition**:* An obstacle for agent- and user-oriented proposals is how to obtain information from the network. From the current network status, agents and users take their decisions. For instance, VoIP applications have end-to-end delay restrictions, which is a metric difficult to measure since it requires clock synchronization between end-hosts. Besides metric acquisition, another challenge is how to consolidate information and to precisely represent the current network status for decision makers. In controller-oriented proposals, on the other hand, it is easier to acquire and consolidate information because the controller performs both tasks in a centralized fashion.

The **Complexity Oblivious Network Management (CONMan)** [109] architecture proposes an interface to simplify information acquisition. The authors argue that a major challenge for network management is that protocols and devices have many implementation details, which makes data acquisition more complex. The CONMan architecture proposes an interface providing minimum protocol- and device-specific information required to simplify data acquisition and to improve the network performance from a more effective management. CONMan provides information regarding the physical medium, the switching between different media, packet filtering, performance, and security (data integrity, authenticity, and confidentiality).

NetQuery [110], unlike CONMan, provides information regarding ISPs performance relying on trusted hardware for device-specific information. In addition, it relies on audit-based credentials for network information provided by trusted third parties. NetQuery is an implementation of a Knowledge Plane to assist autonomic decisions by network entities, such as ISPs and users. To avoid global view, each ISP maintains servers to manage their own information. The different ISPs are federated and can obtain information from others, once they are allowed to do so. The sort of information available, if any, depends on the amount of trust one ISP has on another.

G. Internet scalability

First, we describe proposals focused on networking strategies to improve scalability. Afterwards, we present proposals to enhance scalability by better organizing routing tables. Finally, we present one proposal to speedup FIB searches based on hardware optimization.

*1) **Networking proposals**:* A trivial proposal investigated to reduce scalability problems is to discourage the use of disjoint address ranges. This is done following two basic approaches: access networks can no longer use provider-independent address ranges and cannot split an address range received from one of its ISPs [44], [111]. In this proposal, multihomed stations must use IP addresses from the ISPs directly connected. This constraint allows each ISP to announce aggregated prefix addresses.

The **Globally Routable Address (GRA) and Globally Deliverable Address (GDA)** is a more flexible proposal that

separates the address space into two types [43], [44]. GRAs are composed of the addresses from DFZ routing tables, which are only reachable within the DFZ. In opposition, GDAs are Internet edge network addresses, which must be unique and reachable anywhere. Nevertheless, GDAs must not be stored in DFZ tables and, as a consequence, the absence of these addresses in DFZ routers reduces the number of entries and the number of prefixes announced by BGP. The basic assumption is that eliminating edge network prefixes from interdomain routing reduces routing table sizes and the number of updates by up to one order of magnitude [43]. Because the number of ISPs has been stable compared with the number of access networks [43], tables containing GRAs are not expected to significantly increase in the short-term. GDAs, on the other hand, are assigned to access networks, and must be unique to guarantee correct delivery. As ISPs are only aware of GRA addresses, a mapping service between the two address types is needed. Such mapping is done by ISPs edge routers connected to the access networks in charge of discovering the corresponding destination address of a received packet. An example of the mapping and encapsulating procedures [44] is depicted in Figure 32. In this figure, packets originated in station A and destined to B are tunneled from R_A to R_B. Figure 32 also shows the address space division. Note that the routers inside the GRA space do not need to know the mapping and encapsulating mechanisms. This permits the same configuration in use today, but with fewer entries in routing tables.

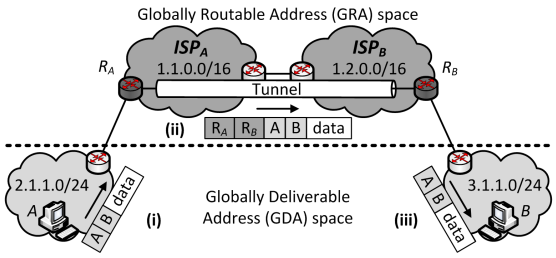


Fig. 32. Packet forwarding using Globally Deliverable Address (GDA) and Globally Reachable Address (GRA) spaces: (i) packets are sent using GDA; (ii) these packets are encapsulated upon entering in the DFZ using GRAs; (iii) upon leaving the DFZ, packets are decapsulated to be delivered to the correct destination.

Using GDAs allows each ISP to improve address aggregation. Moreover, each access network can use provider-independent addresses, as seen in Figure 32, simplifying multihoming as well as network address prefix changes. Each change, however, must be known by the edge routers in charge of the mapping service so as to maintain correctness. The mapping information updates may raise scalability issues because they are disseminated throughout the DFZ. In addition, GDA/GRA uses the same basic principle of LISP, therefore, authors claim that GDA/GRA can also be adapted to deal with mobility.

A Practical Tunneling (APT) [112] architecture is a hybrid solution that proposes that the mapping information must be spread into the DFZ. Nevertheless, in each network in DFZ, only a limited number of specialized network devices receive such information. These devices, called Default Mappers (DMs), store complete tables with all mapping information whereas edge routers cache only the most recent mapping requests. If there is a match for a packet, the edge router encapsulates the packet and proceeds with the regular forwarding operation. Otherwise, it forwards the packet to a DM, which handles the packet as if it was a map request and sends back the response to the edge router. At the same time, the DM encapsulates and forwards the packet on behalf of the requesting router to speed up the communication. This hybrid strategy reduces table sizes in edge routers and does not overload DMs by forwarding many packets. Note that a single edge prefix can be mapped into multiple egress transit routers, providing multihoming support. In addition, control messages are encrypted to avoid traffic divert, which would be possible if a malicious entity changes the information therein.

The **Six/One Router** [113] protocol maps the addresses of stations in different access networks taking into account backward compatibility. The address mapping is done at the edge router of the source station access network, which replaces the packet source address with a DFZ valid IP address. Similarly, the source edge router performs an address resolution procedure to map the packet destination address into a DFZ address. This biunivocal relationship imposes each station's address in an access network to be mapped into a unique DFZ address from its corresponding ISP. The router running the Six/One Router protocol also inserts an extension into the packet header to identify source and destination addresses in their respective access networks. Thus, when an edge router in the destination access network receives a packet, it maps the source and destination addresses into the original ones before sending the packet to the correct destination. Figure 33 illustrates the operation of the Six/One Router protocol involving a legacy network. Station A has its address mapped into an address of its ISP. Nevertheless, the address of destination B is maintained because its access network does not perform address separation. Figure 33 also shows the packet header extension inserted by router R_A. Note that whereas station A access network can use an address outside its provider (ISP_A) range, station B must use an address from its provider (ISP_B) range. In the reverse direction, if station B originates a packet, the destination address would be the address of the destination in the DFZ (R_A). Relying on the biunivocal relationship, A is the only station associated to R_A address. Even though Six/One Router does not directly tackle security issues, it assumes that the mapping resolution system is secure.

The **Hierarchical Architecture for Internet Routing (HAIR)** [114] is another proposal to reduce the number of entries in DFZ routing tables. Different from two-level address separation proposals, HAIR divides the different ASes in more hierarchical levels with core Internet transit ASes at the top and access networks at the bottom. HAIR also uses Loc/ID split techniques to support mobile stations. Packet forwarding is then based on source and destination *identifiers*, which are mapped into *addresses* by the source station before sending the packet. The source address encodes the identifiers of all edge routers traversed by the packet from source to

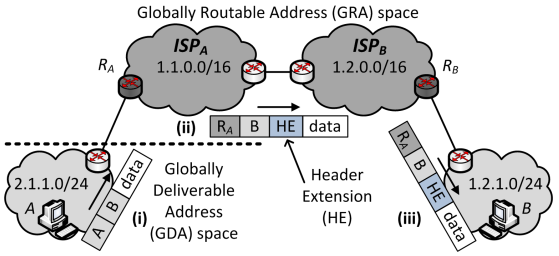


Fig. 33. Six/One Router operation in case of a legacy access network: (i) station A sends a packet destined to station B; (ii) station A has its address mapped into an address of its ISP, whereas station is maintained because its access network does not perform address separation; (iii) packet from A is delivered to B, which is responsible for decapsulating the packet.

the top level. Similarly, the destination address encodes edge router identifiers from the top level to the destination. The source station knows the destination identifier and, on demand, uses a resolving service to receive the corresponding address. Figure 34 shows the source and destination addresses used. Note that the source address is encoded by a function f having as an input the identifiers of station A, of the first edge router at the hierarchical level just above (R_A), and of the core edge router (R_{CA}). The ASes hierarchical division and the utilization of source routing limit each router view to its own hierarchical level, reducing the number of routing entries in the DFZ.

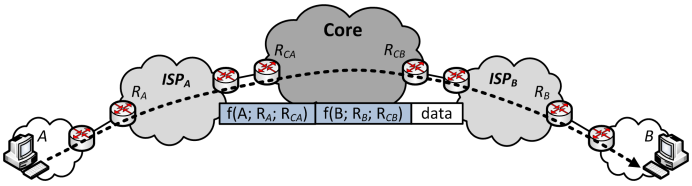


Fig. 34. HAIR operation.

BGP announces reachability messages using network IP prefixes for identification. Nevertheless, the path to a given network is expressed as a sequence of autonomous system numbers (ASNs). This combination introduces a scalability problem because the number of prefixes is much larger then the number of ASes. As a consequence, works to improve Internet scalability are also investigating the possibility of using ASNs in the interdomain routing tables instead of IP prefixes. **Shue and Gupta** propose the utilization of ASNs for routing announcements and for packet forwarding [115]. Besides reducing the DFZ routing tables, it also permits addresses with fixed sizes, which speed up the lookup procedure. Shue and Gupta, in addition, use names to identify end-hosts instead of IP addresses. Hence, a user willing to communicate to a given node resolves the destination name and receives back the corresponding destination ASN. Upon arriving at the destination ASN, the name is used for forwarding purposes. It is worth mentioning that locators are not used, and therefore, Shue and Gupta work could also be considered an alternative Loc/ID split proposal.

The **Less-Is-More Architecture (LIMA)** [116] architecture also uses ASNs for routing. In this case, however, the IP address of individual stations is dismembered into the provider AS number, the provider-local AS stub, and the stub-local intradomain address. Using the provider ASN, LIMA avoids provider-independent addresses and multihomed stubs, reducing the number of interdomain routing entries. LIMA also does not inject reachability information from stubs into the DFZ so as to further reduce the routing table sizes. LIMA considers security issues by using dynamic assignment of intradomain addresses without end-host correlation. Mobility is also taken into account as in mobile IP. To this end, LIMA assumes that only a small fraction of nodes leaves the home network and, therefore, can be handled by mobile-IP-like proposals.

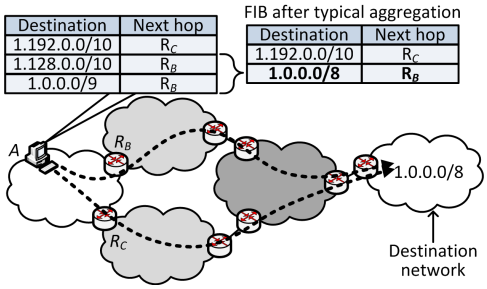
The **Recursive InterNetwork Architecture (RINA)** [117], [118], [119] builds upon the principle that applications communicate through inter-process communication (IPC) facilities. For an application to communicate through the distributed IPC facility, it only needs to know the name of the destination application and to use the IPC interface to request communication. The scope and functions provided by the different IPC facilities may vary given the different type of network and performance goals. Moreover, an IPC layer may recursively request services from other IPC layers. The idea of recursively use multiple inter-process communication services creates a multilayer structure repeated until an IPC facility can fit well the physical medium, e.g., wired or wireless medium. As a consequence, from a top-down approach, the recursive layers become more adapted for a finer-grained task in a more limited scope or region.

The premise that applications are named and that only application names are needed for them to communicate provide interesting properties to RINA. First, as addresses are not needed, they can be locally assigned by the IPC facility. Therefore, by default, private addresses are used inside an IPC facility, by default. Therefore, scalability problems related to global routing tables do not exist. This means also that global announcements are reduced, avoiding control overhead, and can reduce the address space to accommodate a fewer number of identifiers. In addition, applications need to authenticate to access a distributed IPC facility in RINA, naturally leading to a secure architecture. In RINA, from the application viewpoint, mobility is a special case of multihoming: an application name does not change when it moves, instead, links fail and reconnect. In summary, different IPC facilities can be used to tackle many future Internet issues such as scalability, multihoming, mobility, and security. Although RINA is fundamentally a clean-slate approach, it permits a soft transition. Among other possibilities, RINA can coexist with the current TCP/IP in a more limited scope, e.g., in a subnetwork, such as a layer 2.5 protocol.

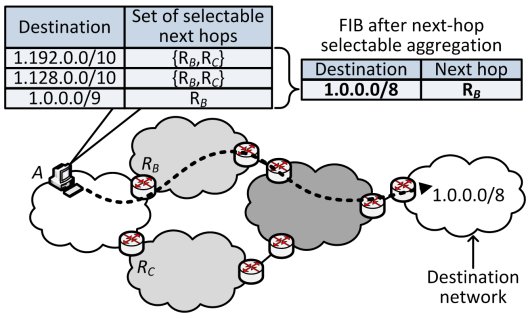
*2) **Routing table organization**:* The **Virtual Aggregation (ViAggre)** [120], [121] distributes the complete intradomain routing table among ISP routers. ViAggre divides the address space into a set of virtual prefixes assuming that each one is larger than the real aggregated prefixes maintained by each router. For instance, the entire address space known by an ISP can be divided into 128 virtual /7 prefixes (0.0.0.0/7 to

254.0.0.0/7), where each virtual prefix corresponds to a set of real ones. These virtual prefixes do not necessarily correspond to the network topology, but must include all real possible prefixes to guarantee a completely connected network. The virtual networks generated from the virtual prefixes compose a topology with aggregated addresses and, therefore, is more scalable. To create a virtual network, each ISP chooses some routers to be part of it. These routers maintain routes to all prefixes inside the virtual network and are then called aggregation points. In addition, they can aggregate more than one virtual prefix and must store routes only to the prefixes within its virtual network.

Li *et al.* extrapolate single-path forwarding with multiple selectable next hops to reduce FIBs' sizes [122]. Therefore, instead of searching for entries with numerically aggregatable prefixes and with identical next hop as a condition for aggregation, they propose searching for numerically aggregatable prefixes with a set of selectable next hops with non-empty intersection. As seen in Figure 35(a), according to standard single-path forwarding, station A can only aggregate its FIB entry (1.0.0.0/9, R_B) with (1.128.0.0/10, R_B), but not with (1.192.0.0/10, R_C) because the last entry does not have the same next hop. Nevertheless, it is possible to reach the same destination using both R_B and R_C as the next hop, if the multiple egress routers are considered. Thus, building FIB prefixes with a set of selectable next hops would lead to more aggregatable entries as soon as the intersection between these sets is non empty. For example, in Figure 35(b), the entries (1.0.0.0/9, R_B), (1.128.0.0/10, $\{R_B, R_C\}$), and (1.192.0.0/10, $\{R_B, R_C\}$) can now be aggregated because R_B is an intersection among the selectable next hop sets.



(a) Typical FIB aggregation.



(b) Next-hop Selectable FIB aggregation.

Fig. 35. Different FIB aggregation approaches.

*3) **Hardware management**:* **Kim *et al.*** [59] cache a routing table only with often used routes. Other routes are stored in a slower memory and looked up only if there is a cache miss. If a packet is received by the router and the route is not available in cache, the router will immediately forward the packet through a default route and update its cache based on the information within its slower memory. The router used as the next hop of the default route is especially designed to know paths to all possible destinations in the network. Besides using cache, Kim *et al.* also propose the use of uniform-length prefixes. By analyzing real data, authors conclude that /24 prefixes are the highest granularity not filtered by providers for the sake of security. A consequence of the best prefix match strategy, using too specific prefixes can lead to prefix hijacking because of misconfiguration or malicious actions. On the other hand, combining different prefix sizes with cached routes can lead to cache-hiding problems. For example, prefixes with shorter lengths, e.g. /16, can be further divided into entries with longer prefixes using different output interfaces. The entry 10.1.0.0/16 can have `eth0` as its output interface, whereas the entry 10.1.1.0/24 can have `eth1`. If a packet destined to 10.1.2.1 is received, the entry with the longest prefix for this address, i.e., 10.1.0.0/16, would be in the cache and the packet would be forwarded through `eth0`. If right after, another packet destined to 10.1.1.1 is received, this packet would also be forwarded through `eth0`, instead of through `eth1`, which would be the best option. Kim *et al.* propose the division of the address space into /24 networks since this prefix predominates in their experimental results. Hence, using only /24 networks can reduce FIB sizes at the same time it can avoid introducing neither security nor cache-hiding problems.

VI. Summary

Table I summarizes main protocols surveyed in this work, highlighting their more prominent characteristics. We aim at demonstrating that, on the one hand, a given protocol typically does not handle a single requirement for the future Internet interconnection. On the other hand, up to now, there is no single proposal that handles all of them. This is an evidence of the problem complexity, which is intriguing researchers and is motivating the proposal of many design architectures for the future Internet [123], [124], [19], [125], [126]. The main observation is the contrast between the several emerging new application-layer protocols and medium access technologies against the "ossified" IP. This results in the so-called Internet hourglass shape, where the waist is mainly at the network layer, followed by the transport layer [127]. A notable example is the approximately 15 years of IPv4 to IPv6 transition, which has not been concluded yet.

It is worth mentioning that our classification for future interconnection proposals is not exhaustive. For instance, security is not directly addressed in this survey as one of the requirements for the future Internet, but it is clear from the proposals listed herein that it is indeed a great concern [128]. Note in Table I that, besides security, scalability and incremental deployment are essential. Scalability is always contemplated by at least one protocol of each proposal class and incremental

deployment is only not contemplated by flat routing protocols given its intrinsic complexity for real implementations in the current Internet. These three requirements are fundamental for the future Internet, which must be secure and, at the same time, capable to evolve.

We observe in Table I that some classes address a quite similar set of requirements. Loc/ID split, flat routing, and network mobility, for instance, typically tackle both mobility and multihoming at the same time (except VRR), even though they are conceptually different. We can also note that the more radical proposals for the future Internet, i.e., flat routing and programmable paths, cannot be incrementally deployable, as expected. In addition, content-based routing does not provide path customization, as it hides internal information from users. As a final observation, since multihoming is pointed out as one of the great villains of Internet scalability, several Internet scalability proposals handle this requirement.

VII. Conclusion

This survey has presented evolutionary as well as disruptive proposals for Future Internet network interconnection. Proposals have shown more concern with incremental deployment although they still imply network changes. This is because clean-slate proposals have to face the huge obstacle of drastically changing the Internet with minimal or no service disruption. This limitation always brings up concerns about soft transition periods, which can also be seen as an evolution. The lesson learned is that the problem is complex and maybe it will not be possible to embrace all emerging requirements. Requirements such as mobility may still be handled apart even in the upcoming Future Internet. Therefore, developing a new architecture economically feasible and, at the same time, with the same characteristics responsible for the Internet to become one of the greatest successes of the 20^{th} century is a great challenge. Moreover, improving Internet capacity to evolve in a scalable fashion so as to handle all the possible upcoming requirements is a hot research topic for the next years and will still lead to great interest from the networking community.

Acknowledgment

The authors would like to thank CNPq, CAPES, FAPERJ, FINEP/FUNTTEL for their financial support.

References

[1] D. Clark, R. Braden, K. Sollins, J. Wroclawski, D. Katabi, J. Kulik, X. Yang, T. Faber, A. Falk, V. Pingali, M. Handley, and N. Chiappa, "New Arch: Future generation Internet architecture," USC Information Sciences Institute Computer Networks Division, MIT Laboratory for Computer Science and International Computer Science Institute (ICSI), Tech. Rep., Aug. 2004.

[2] R. Hinden and A. Sheltzer, "The DARPA Internet gateway," IETF Network Working Group RFC 823, Sep. 1982.

[3] C. History Museum, "Internet history," Available at http://www.computerhistory.org, Mar. 2010.

[4] D. L. Mills, "Exterior Gateway Protocol formal specification," IETF Network Working Group RFC 904, Apr. 1984.

[5] K. Lougheed and Y. Rekhter, "A Border Gateway Protocol (BGP)," IETF Network Working Group RFC 1105, Jun. 1989.

[6] Caida, "Visualizing IPv4 and IPv6 Internet topology at a macroscopic scale in 2010," Available at http://www.caida.org/research/topology/as_core_network/, Aug. 2011.

[7] Internet World Stats, "World Internet users and population stats," Available at http://www.internetworldstats.com/stats.htm, Jan. 2013.

[8] CIDR, "Active BGP entries (FIB) - data sets," Available at http://bgp.potaroo.net/as2.0/bgp-active.html, Jan. 2013.

[9] I. T. U. (ITU), "Global ICT trends," Available at http://www.itu.int/ITU-D/ict/statistics/, Apr. 2013.

[10] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," IETF Network Working Group RFC 1771, Mar. 1995.

[11] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF Network Working Group RFC 4271, Jan. 2006.

[12] S. Deering and R. Hinden, "Internet Protocol, version 6 (IPv6) specification," IETF Network Working Group RFC 2460, Dec. 1998.

[13] C. Perkins, "IP mobility support for IPv4," IETF Network Working Group RFC 3344, Aug. 2002.

[14] S. E. Deering, "Multicast routing in internetworks and extended LANs," in *ACM SIGCOMM*, Aug. 1988, pp. 55–64.

[15] L. H. M. K. Costa, S. Fdida, and O. C. M. B. Duarte, "Incremental service deployment using the Hop By Hop Multicast Routing Protocol," *IEEE/ACM Transactions on Networking*, vol. 14, no. 3, pp. 543–556, Jun. 2006.

[16] J. Crowcroft, "Toward a network architecture that does everything," *Communications of ACM*, vol. 51, no. 1, pp. 74–77, Jan. 2008.

[17] S. Ratnasamy, S. Shenker, and S. McCanne, "Towards an evolvable Internet architecture," *ACM SIGCOMM Computer Communication Review*, vol. 35, no. 4, pp. 313–324, Apr. 2005.

[18] A. Ghodsi, T. Koponen, B. Raghavan, S. Shenker, A. Singla, and J. Wilcox, "Intelligent design enables architectural evolution," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Nov. 2011, pp. 1–6.

[19] T. Koponen, S. Shenker, H. Balakrishnan, N. Feamster, I. Ganichev, A. Ghodsi, P. B. Godfrey, N. McKeown, G. Parulkar, B. Raghavan, J. Rexford, S. Arianfar, and D. Kuptsov, "Architecting for innovation," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 3, pp. 24–36, Jul. 2011.

[20] J. Pan, S. Paul, and R. Jain, "A survey of research on future internet architectures," *IEEE Communications Magazine*, vol. 49, no. 7, pp. 26–36, Jul. 2011.

[21] G. S. Malkin, "RIP Version 2," IETF Network Working Group RFC 2453, Nov. 1998.

[22] J. Moy, "OSPF Version 2," IETF Network Working Group RFC 2328, Apr. 1998.

[23] X. Dimitropoulos, D. Krioukov, G. Riley, and K. Claffy, "Revealing the autonomous system taxonomy: The machine learning approach," in *Passive and Active Network Measurement Workshop (PAM)*, Mar. 2006, pp. 1–10.

[24] Z. Ge, D. Figueiredo, S. Jaiwal, and L. Gao, "On the hierarchical structure of the logical internet graph," in *SPIE ITCOM*, Aug. 2001, pp. 1–15.

[25] R. Govindan and A. Reddy, "An analysis of Internet inter-domain topology and route stability," in *IEEE Conference on Computer Communications (INFOCOM)*, Apr. 1997, pp. 850–857.

[26] A. A. Hanbali, E. Altman, and P. Nain, "Survey of TCP over ad hoc networks," *IEEE Communications Surveys & Tutorials*, vol. 7, no. 3, pp. 22–36, Third Quarter 2005.

[27] Y. Wang, E. Keller, B. Biskeborn, J. van der Merwe, and J. Rexford, "Virtual routers on the move: live router migration as a network-management primitive," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 231–242, Aug. 2008.

[28] P. S. Pisa, N. C. Fernandes, H. E. T. Carvalho, M. D. D. Moreira, M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "OpenFlow and Xen-based virtual network migration," in *The World Computer Congress 2010 - Network of the Future Conference*, Sep. 2010, pp. 170–181.

[29] D. Johnson, C. Perkins, and J. Arkko, "Mobility support in IPv6," IETF Network Working Group RFC 3775, Jun. 2004.

[30] B. McCarthy, C. Edwards, and M. Dunmore, "Using NEMO to support the global reachability of MANET nodes," in *IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2009, pp. 2097–2105.

[31] D. Meyer, "The Locator Identifier Separation Protocol (LISP)," *The Internet Protocol Journal*, vol. 11, no. 1, pp. 23–36, Mar. 2008.

[32] E. Nordmark and M. Bagnulo, "Shim6: Level 3 multihoming Shim protocol for IPv6," IETF Network Working Group RFC 5533, Jun. 2009.

[33] CIDR, "Active BGP entries (FIB)," Available at http://www.cidr-report.org/as2.0/, Apr. 2013.

[34] J. He and J. Rexford, "Towards Internet-wide multipath routing," *IEEE Network*, vol. 22, no. 2, pp. 16–21, Mar-Apr 2008.

[35] F. Valera, I. van Beijnum, A. García-Martínez, and M. Bagnulo, *Multipath BGP: motivations and solutions*, ser. Next-Generation Internet: Architectures and Protocols. Cambridge University Press, Jan. 2009, ch. 12, pp. 238–256.

[36] M. Goyal, M. Soperi, E. Baccelli, G. Choudhury, A. Shaikh, H. Hosseini, and K. Trivedi, "Improving convergence speed and scalability in OSPF: A survey," *IEEE Communications Surveys & Tutorials*, vol. 14, no. 2, pp. 443–463, Second Quarter 2012.

[37] E. Elena, J.-L. Rougier, and S. Secci, "Characterisation of AS-level path deviations and multipath in internet routing," in *EURO-NF Conference on Next Generation Internet (NGI)*, Jun. 2010, pp. 1–7.

[38] C. Raiciu, C. Pluntke, S. Barre, A. Greenhalgh, D. Wischik, and M. Handley, "Data center networking with multipath TCP," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Oct. 2010, pp. 10:1–10:6.

[39] C. Raiciu, S. Barre, C. Pluntke, A. Greenhalgh, D. Wischik, and M. Handley, "Improving datacenter performance and robustness with multipath TCP," in *ACM SIGCOMM*, Aug. 2011, pp. 266–277.

[40] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: Enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, Apr. 2008.

[41] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," IETF Network Working Group RFC 4984, Sep. 2007.

[42] CIDR, "Aggregation summary," Available at http://www.cidr-report.org/as2.0/, Apr. 2013.

[43] D. Massey, L. Wang, B. Zhang, and L. Zhang, "A scalable routing system design for future Internet," in *ACM SIGCOMM IPv6 and the Future of the Internet Workshop*, Aug. 2007, pp. 1–6.

[44] D. Jen, M. Meisel, H. Yan, D. Massey, L. Wang, B. Zhang, and L. Zhang, "Towards a new Internet routing architecture: Arguments for separating edges from transit core," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Oct. 2008, pp. 1–6.

[45] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, I. Stoica, and S. Shenker, "ROFL: Routing On Flat Labels," in *ACM SIGCOMM*, Aug. 2006, pp. 363–374.

[46] I. Stoica, R. Morris, D. Liben-Nowell, D. Karger, M. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 17–32, Feb. 2003.

[47] P. Ganesan, K. Gummadi, and H. Garcia-Molina, "Canon in G major: Designing DHTs with hierarchical structure," in *International Conference on Distributed Computing Systems (ICDCS)*, Mar. 2004, pp. 263–272.

[48] V. Jacobson, D. Smetters, J. Thornton, M. Plass, N. Briggs, and R. Braynard, "Networking named content," in *Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, Dec. 2009, pp. 1–14.

[49] J. L. Martins and S. Duarte, "Routing algorithms for content-based publish/subscribe systems," *IEEE Communications Surveys & Tutorials*, vol. 12, no. 1, pp. 39–58, First Quarter 2010.

[50] A. Passarella, "A survey on content-centric technologies for the current Internet: CDN and P2P solutions," *Computer Communications*, vol. 35, no. 1, pp. 1–32, Jan. 2012.

[51] A. Carzaniga, M. J. Rutherford, and A. L. Wolf, "A routing scheme for Content-Based Networking," in *IEEE Conference on Computer Communications (INFOCOM)*, Mar. 2004, pp. 918–928.

[52] T. Koponen, S. Shenker, I. Stoica, M. Chawla, B. Chun, A. Ermolinsky, and K. Kim, "A Data-Oriented (and beyond) Network Architecture," in *ACM SIGCOMM*, Aug. 2007, pp. 181–192.

[53] A. Ghodsi, T. Koponen, J. Rajahalme, P. Sarolahti, and S. Shenker, "Naming in Content-Oriented Architectures," in *ACM SIGCOMM Workshop on Information-Centric Networking*, Aug. 2011, pp. 1–6.

[54] J. Choi, J. Han, E. Cho, T. Kwon, and Y. Choi, "A survey on content-oriented networking for efficient content delivery," *IEEE Communications Magazine*, vol. 49, no. 3, pp. 121–127, Mar. 2011.

[55] B. Ahlgren, M. D'Ambrosio, M. Marchisio, I. Marsh, C. Dannewitz, B. Ohlman, K. Pentikousis, O. Strandberg, R. Rembarz, and V. Vercellone, "Design considerations for a network of information," in *ACM Workshop on Re-Architecting the Internet (ReArch)*, Dec. 2008, pp. 66:1–66:6.

[56] K. Visala, D. Lagutin, and S. Tarkoma, "LANES: An inter-domain data-oriented routing architecture," in *ACM Workshop on Re-Architecting the Internet (ReArch)*, Dec. 2009, pp. 55–60.

[57] D. D. Clark, C. Partridge, J. C. Ramming, and J. T. Wroclawski, "A knowledge plane for the Internet," in *ACM SIGCOMM*, Aug. 2003, pp. 3–10.

[58] R. Sherwood, M. Chan, A. Covington, G. Gibb, M. Flajslik, N. Handigol, T.-Y. Huang, P. Kazemian, M. Kobayashi, J. Naous, S. Seetharaman, D. Underhill, T. Yabe, K.-K. Yap, Y. Yiakoumis, H. Zeng, G. Appenzeller, R. Johari, N. McKeown, and G. Parulkar, "Carving research slices out of your production networks with OpenFlow," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 1, pp. 129–130, 2010.

[59] C. Kim, M. Caesar, A. Gerber, and J. Rexford, "Revisiting route caching: The world should be flat," in *International Conference on Passive and Active Network Measurement (PAM)*, Apr. 2009, pp. 3–12.

[60] I. Stoica, D. Adkins, S. Zhuang, S. Shenker, and S. Surana, "Internet indirection infrastructure," *IEEE/ACM Transactions on Networking*, vol. 12, no. 2, pp. 205–218, Apr. 2004.

[61] K. Lakshminarayanan, D. Adkins, A. Perrig, and I. Stoica, "Brief announcement: towards a secure indirection infrastructure," in *ACM PODC*, Jul. 2004, pp. 383–383.

[62] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "The Locator/ID separation protocol (LISP)," IETF Network Working Group RFC 6830, Jan. 2013.

[63] D. Saucez, L. Iannone, and O. Bonaventure, "OpenLISP: An open source implementation of the locator/ID separation protocol," in *ACM SIGCOMM Demos Session*, 2009, pp. 1–2.

[64] D. Saucez, B. Donnet, L. Iannone, and O. Bonaventure, "Interdomain traffic engineering in a locator/identifier separation context," in *Internet Network Management Workshop (INM)*, Oct. 2008, pp. 1–6.

[65] S. Secci, L. Kunpeng, G. K. Rao, and B. Jabbari, "Resilient traffic engineering in a transit-edge separated internet routing," in *IEEE International Conference on Communications (ICC)*, Jun. 2011, pp. 1–6.

[66] R. Atkinson, S. Bhatti, and S. Hailes, "Evolving the Internet architecture through naming," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1319–1325, Oct. 2010.

[67] R. Moskowitz, P. Nikander, P. Jokela, and T. Henderson, "Host Identity Protocol," IETF Network Working Group RFC 5201, Apr. 2008.

[68] OpenHIP, "Host Identity Protocol (HIP)," Available at http://www.openhip.org/, Dec. 2009.

[69] P. Nikander, T. Henderson, C. Vogt, and J. Arkko, "End-host mobility and multihoming with the Host Identity Protocol," IETF Network Working Group RFC 5206, Apr. 2008.

[70] A. R. Natal, L. Jakab, M. Portolés, V. Ermagan, P. Natarajan, F. Maino, D. Meyer, and A. C. Aparicio, "LISP-MN: Mobile networking through LISP," *Wireless Personal Communications*, vol. 70, no. 1, pp. 253–266, May 2013.

[71] M. Menth, M. Hartmann, and K. Dominik, "Global locator, local locator, and identifier split (GLI-Split)," *Future Internet*, vol. 5, no. 1, pp. 67–94, 2013.

[72] M. Hoefling, M. Menth, and M. Hartmann, "A survey of mapping systems for locator/identifier split internet routing," *IEEE Communications Surveys & Tutorials (to appear)*, 2013.

[73] L. Iannone and O. Bonaventure, "On the cost of caching locator/ID mappings," in *ACM CoNEXT*, Dec. 2007, pp. 7:1–7:12.

[74] H. Luo, Y. Qin, and H. Zhang, "A DHT-based identifier-to-locator mapping approach for a scalable Internet," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1790–1802, Dec. 2009.

[75] E. Lear, "NERD: A Not-so-novel ID (EID) to Routing Locator (RLOC) Database," IETF Network Working Group RFC 6837, Jan. 2013.

[76] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS hierarchy to support the LISP mapping system," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1332–1343, Oct. 2010.

[77] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "LISP ALternative Topology (LISP+ALT)," IETF Network Working Group RFC 6836, Jan. 2013.

[78] M. Menth, M. Hartmann, and M. Hofling, "FIRMS: A mapping system for future internet routing," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 8, pp. 1326–1331, 2010.

[79] M. Caesar, M. Castro, E. B. Nightingale, G. O'Shea, and A. Rowstron, "Virtual Ring Routing: network routing inspired by DHTs," in *ACM SIGCOMM*, Aug. 2006, pp. 351–362.

[80] M. E. M. Campista, L. H. M. K. Costa, and O. C. M. B. Duarte, "A routing protocol suitable for backhaul access in wireless mesh

networks," *Elsevier Computer Networks*, vol. 56, no. 2, pp. 703–718, Feb. 2012.

[81] R. Laufer, H. Dubois-Ferrière, and L. Kleinrock, "Polynomial-time algorithms for multirate anypath routing in wireless multihop networks," *IEEE/ACM Transactions on Networking*, vol. 20, no. 3, pp. 742–755, Jun. 2012.

[82] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker, "Accountable Internet Protocol (AIP)," in *ACM SIGCOMM*, Aug. 2008, pp. 339–350.

[83] A. Singla, P. B. Godfrey, K. Fall, G. Iannaccone, and S. Ratnasamy, "Scalable routing on flat names," in *ACM CoNEXT*, Dec. 2010, pp. 20:1–20:12.

[84] E. Perera, V. Sivaraman, and A. Seneviratne, "Survey on network mobility support," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 2, pp. 7–19, Apr. 2004.

[85] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert, "Network Mobility (NEMO) basic support protocol," IETF Network Working Group RFC 3963, Jan. 2005.

[86] B. McCarthy, M. Jakeman, C. Edwards, and P. Thubert, "Protocols to efficiently support nested NEMO (NEMO+)," in *International workshop on Mobility in the evolving Internet Architecture (MobiArch)*, Aug. 2008, pp. 43–48.

[87] M. Sabeur, B. Jouaber, and D. Zeghlache, "Light-NEMO+: Route optimization for light-NEMO solution," in *IEEE International Conference on Networks (ICON)*, Sep. 2006, pp. 1–6.

[88] B. McCarthy, C. Edwards, and M. Dunmore, "Using NEMO to extend the functionality of MANETs," in *IEEE International Conference on Communications (ICC)*, May 2008, pp. 455–460.

[89] T. Clausen, P. Jacquet, A. Laouiti, P. Muhlethaler, A. Qayyum, and L. Viennot, "Optimized link state routing protocol," in *IEEE International Multi Topic Conference (INMIC)*, Dec. 2001, pp. 62–68.

[90] N. Toledo, M. Higuero, E. Jacob, and J. Matias, "Analytical evaluation of a HIP registration enhancement for NEMO scenarios," *IEEE Communications Letters*, vol. 15, no. 5, pp. 587–589, May 2011.

[91] T.-C. Chen, J.-C. Chen, and Z.-H. Liu, "Secure network mobility (SeNEMO) for real-time applications," *IEEE Transactions on Mobile Computing*, vol. 10, no. 8, pp. 1113–1130, Aug. 2011.

[92] H. T. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, and A. Gandhi, "BANANAS: an evolutionary framework for explicit and multipath routing in the Internet," in *ACM SIGCOMM workshop on Future Directions in Network Architecture (FDNA)*, Aug. 2003, pp. 277–288.

[93] M. Motiwala, M. Elmore, N. Feamster, and S. Vempala, "Path splicing," in *ACM SIGCOMM*, Aug. 2008, pp. 27–38.

[94] G. T. Nguyen, R. Agarwal, J. Liu, M. Caesar, P. B. Godfrey, and S. Shenker, "Slick packets," in *ACM SIGMETRICS*, Jun. 2011, pp. 245–256.

[95] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient Overlay Networks," *ACM SIGOPS Operating Systems Review*, vol. 35, no. 5, pp. 131–145, Dec. 2001.

[96] W. Xu and J. Rexford, "MIRO: Multi-path Interdomain ROuting," in *ACM SIGCOMM*, Aug. 2006, pp. 171–182.

[97] R. Mahajan, D. Wetherall, and T. Anderson, "Towards coordinated interdomain traffic engineering," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Nov. 2004, pp. 1–6.

[98] S. Secci, J.-L. Rougier, A. Pattavina, F. Patrone, and G. Maier, "Peering equilibrium multipath routing: A game theory framework for internet peering settlements," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 419–432, Apr. 2011.

[99] X. Y. D. Wetherall, "Source selectable path diversity via routing deflections," in *ACM SIGCOMM*, Aug. 2006, pp. 159–170.

[100] G. M. Brito, P. B. Velloso, and I. M. Moraes, *Information-Centric Networks: A New Paradigm for the Internet*, 1st ed., ser. FOCUS - Networks and Telecommunications Series. Wiley-ISTE, 2013.

[101] M. D'Ambrosio, C. Dannewitz, H. Karl, and V. Vercellone, "MDHT: A hierarquical name resolution service for Information-Centric Networks," in *ACM SIGCOMM Workshop on Information-Centric Networking*, Aug. 2011.

[102] Y.-J. Kim, J. Lee, G. Atkinson, H. Kim, and M. Thottan, "SeDAX: A scalable, resilient, and secure platform for smart grid communications," *IEEE Journal on Selected Areas in Communications*, vol. 30, no. 6, pp. 1119–1136, Jul. 2012.

[103] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Computer Networks*, vol. 56, pp. 2742–2771, 2012.

[104] D. Perino and M. Varvello, "A reality check for Content Centric Networking," in *ACM SIGCOMM Workshop on Information-Centric Networking*, Aug. 2011, pp. 44–49.

[105] M. Amadeo, C. Campolo, and A. Molinaro, "CRoWN: Content-centric networking in vehicular ad hoc networks," *IEEE Communications Letters*, no. 9, pp. 1380–1383, Sep. 2012.

[106] X. Yang, D. Clark, and A. W. Berger, "NIRA: a New Inter-domain Routing Architecture," *IEEE/ACM Transactions on Networking*, vol. 15, no. 4, pp. 775–788, Apr. 2007.

[107] X. Yang, "NIRA: a New Internet Routing Architecture," in *ACM SIGCOMM workshop on Future Directions in Network Architecture (FDNA)*, Aug. 2003, pp. 301–312.

[108] P. B. Godfrey, I. Ganichev, S. Shenker, and I. Stoica, "Pathlet routing," in *ACM SIGCOMM*, Aug. 2009, pp. 111–122.

[109] H. Ballani and P. Francis, "CONMan: A step towards network manageability," *ACM SIGCOMM Computer Communication Review*, vol. 37, no. 4, pp. 205–216, Oct. 2007.

[110] A. Shieh, E. G. Sirer, and F. B. Schneider, "NetQuery: A knowledge plane for reasoning about network properties," in *ACM SIGCOMM*, Aug. 2011, pp. 278–289.

[111] Y. Song, L. Gao, and F. Kenji, "Resilient routing under hierarchical automatic addressing," in *IEEE Globecom*, Dec. 2011, pp. 1–5.

[112] D. Jen, M. Meisel, D. Massey, L. Wang, B. Zhang, and L. Zhang, "APT: A Practical Tunneling architecture for routing scalability," UCLA, Tech. Rep., 2009.

[113] C. Vogt, "Six/One Router: A scalable and backwards compatible solution for provider-independent addressing," in *Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*, Aug. 2008, pp. 13–18.

[114] A. Feldmann, L. Cittadini, W. Mühlbauer, R. Bush, and O. Maennel, "HAIR: Hierarchical Architecture for Internet Routing," in *ACM Workshop on Re-Architecting the Internet (ReArch)*, Dec. 2009, pp. 43–48.

[115] C. A. Shue and M. Gupta, "An internet without the internet protocol," *Elsevier Computer Networks*, vol. 54, no. 18, pp. 3232–3245, Dec. 2010.

[116] J. Li, M. Veeraraghavan, M. Reissleiny, M. Manley, R. Williams, P. Amerz, and J. Leighton, "A less-is-more architecture (LIMA) for a future Internet," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Mar. 2012, pp. 55–60.

[117] J. Day, I. Matta, and K. Mattar, "Networking is IPC: a guiding principle to a better internet," in *ACM CoNEXT*, Dec. 2008, pp. 67:1–67:6.

[118] J. Day, E. Trouva, E. Grasax, P. Phelank, M. P. de Leon, S. Bunch, I. Matta, L. T. Chitkushev, , and L. Pouzin, "Bounding the router table size in an ISP network using RINA," in *Network of the Future Conference (NoF)*, Nov. 2011, pp. 57–61.

[119] J. Touch, I. Baldine, R. Dutta, G. G. Finn, B. Ford, S. Jordan, D. Massey, A. Matta, C. Papadopoulos, P. Reiher, and G. Rouskas, "A Dynamic Recursive Unified Internet Design (DRUID)," *Elsevier Computer Networks*, vol. 55, no. 4, pp. 919–935, Mar. 2011.

[120] H. Ballani, P. Francis, T. Cao, and J. Wang, "ViAggre: Making routers last longer!" in *ACM Workshop on Hot Topics in Networks (HotNets)*, Oct. 2008, pp. 1–6.

[121] ——, "Making routers last longer with ViAggre," in *Symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2009, pp. 453–466.

[122] Q. Li, D. Wang, M. Xu, and J. Yang, "On the scalability of router forwarding tables: Nexthop-selectable FIB aggregation," in *IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2011, pp. 321–325.

[123] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, "A data-oriented (and beyond) network architecture," in *ACM SIGCOMM*, Aug. 2007, pp. 181–192.

[124] A. Anand, F. Dogar, D. Han, B. Li, H. Lim, M. Machado, W. Wu, A. Akella, D. G. Andersen, J. W. Byers, S. Seshan, and P. Steenkiste, "XIA: An architecture for an evolvable and trustworthy Internet," in *ACM Workshop on Hot Topics in Networks (HotNets)*, Nov. 2011, pp. 1–6.

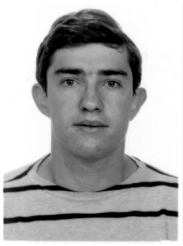
[125] M. B. Anwer and N. Feamster, "Building a fast, virtualized data plane with programmable hardware," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 75–82, Jan. 2010.

[126] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *ACM SIGCOMM Computer Communication Review*, vol. 40, pp. 26–33, Apr. 2010.

[127] S. Akhshabi and C. Dovrolis, "The evolution of layered protocol stacks leads to an hourglass-shaped architecture," in *ACM SIGCOMM*, Aug. 2011, pp. 206–217.

[128] G. Huston, M. Rossi, and G. Armitage, "Securing BGP - a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, pp. 199–222, Second Quarter 2011.

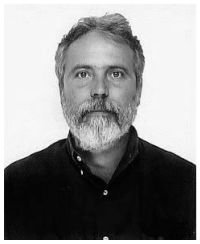
Miguel Elias M. Campista [S'05-M'10] is associate professor with Universidade Federal do Rio de Janeiro (UFRJ), Brazil, since 2010. He received his Telecommunications Engineer degree from the Fluminense Federal University (UFF), Brazil, in 2003 and his M.Sc. and D.Sc. degrees in Electrical Engineering from UFRJ, in 2005 and 2008, respectively. In 2012, Miguel has spent one year with LIP6 at Université Pierre et Marie Curie (UPMC), Sorbonne Universités, Paris, France, as invited professor. He is now heading his research group GTA (Grupo de Teleinformática e Automação) at COPPE/UFRJ. His major research interests are in communication protocols, Internet, wireless networks and complex networks.



Marcelo G. Rubinstein [S'95-M'03] received a B.Sc. degree in electronics engineering, and M.Sc. and D.Sc. degrees in electrical engineering from UFRJ in 1994, 1996, and 2001, respectively. From January to September 2000 he was at the PRiSM Laboratory, University of Versailles, France. He is now an associate professor with Universidade do Estado do Rio de Janeiro (UERJ). His major interests are in wireless networks, home networking, medium access control, and quality of service.

Igor M. Moraes is currently an Associate Professor in the Instituto de Computação (IC) at Universidade Federal Fluminense (UFF). Igor received the cum laude Electronic Engineer degree in 2003 and the M.Sc. and the D.Sc. degrees in electrical engineering from Universidade Federal do Rio de Janeiro (UFRJ) in 2006 and 2009, respectively. His major research interests are in architectures for the Future Internet, information-centric networks, peer-to-peer video streaming systems, wireless networks, and security.

Luís Henrique M. K. Costa received his electronics engineer and M.Sc. degrees in electrical engineering from Universidade Federal do Rio de Janeiro (UFRJ), Brazil, in 1997 and 1998, respectively, and the Dr. degree from Université Pierre et Marie Curie (Paris 6), Paris, France, in 2001. Since August 2004 he has been associate professor with COPPE/UFRJ. His major research interests are in the areas of routing, wireless networks, vehicular networks, and future Internet. Luís has been a member of IEEE COMSOC and from ACM since 2001.



Otto Carlos M. B. Duarte received the Electronic Engineer degree and the M.Sc. degree in electrical engineering from the Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, in 1976 and 1981, respectively, and the Dr.Ing. degree from ENST/Paris, Paris, France, in 1985. Since 1978 he has been a Professor with UFRJ. From January 1992 to June 1993, he was with MASI Laboratory, University Paris 6, Paris. In 1995, he spent three with the International Computer Science Institute (ICSI), University of California, Berkeley. In 1999 and 2001, he was an Invited Professor with the University Paris 6. His major research interests are in multicast, QoS guarantees, security, and mobile communications.

TABLE I
SUMMARY OF THE SURVEYED PROPOSALS FOR FUTURE INTERNETWORKING.

Proposals Classes	**Protocols**	**Addressed Requirements**						
		Mobility	Multi-homing	Multi-path	Path customization	Scalability	Incremental deployment	Security
Loc/ID Split	i3	√	√				√	√
	LISP[a]	√	√		√	√	√	√
	ILNP	√	√			√	√	√
	HIP	√	√				√	√
	Shim6[a]	√	√			√	√	√
	GLI-Split	√	√		√	√	√	√
Flat Routing	ROFL	√	√	√	√			√
	VRR	√						
	AIP	√	√	√				√
	Disco	√	√			√		
Network Mobility	NEMO BS	√	√				√	√
	NEMO+	√	√			√	√	√
	Light-NEMO+	√	√			√	√	√
	MANEMO[b]	√	√			√	√	√
Multiple Paths	BANANAS			√			√	
	MIRO			√		√	√	√
	Path Splicing[c]			√	√	√	√	
	Slick Packets[d]			√	√	√		
Content-based Routing	DONA	√	√	√			√	√
	CCN	√	√	√		√	√	√
	NetInf	√	√	√		√	√	√
Programmable Paths	NIRA				√	√		
	Pathlet			√	√	√		
	CONMan				√			√
	NetQuery				√	√	√	√
	OpenFlow	√			√		√	√
Internet Scalability	GRA/GDA		√			√	√	√
	APT		√		√	√	√	√
	Six/One		√			√	√	√
	HAIR	√	√	√	√	√	√	
	Shue and Gupta	√	√			√	√	
	LIMA	√	√	√		√		√
	ViAggre					√		
	Next-hop selectable					√	√	
	RINA	√	√			√	√	√

[a] It considers scalability as a requirement, but uses mapping systems.

[b] All modifications to NEMO BS aim at improving the previous version in terms of scalability.

[c] Although it speeds up failure recovery, it requires additional state per router.

[d] Although it speeds up failure recovery, it adds up to 50 bytes in packet headers.