The multiple roles that IPv6 addresses can play in today's Internet

Maxime Piraux
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
maxime.piraux@uclouvain.be

Tom Barbette
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
tom.barbette@uclouvain.be

Nicolas Rybowski
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
nicolas.rybowski@uclouvain.be

Louis Navarre
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
louis.navarre@uclouvain.be

Thomas Alfroy
ICube, University of Strasbourg
Strasbourg, France
talfroy@unistra.fr

Cristel Pelsser
ICube, University of Strasbourg
Strasbourg, France
pelsser@unistra.fr

François Michel
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
francois.michel@uclouvain.be

Olivier Bonaventure
ICTEAM, UCLouvain
Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.be

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

The Internet use IP addresses to identify and locate network interfaces of connected devices. IPv4 was introduced more than 40 years ago and specifies 32-bit addresses. As the Internet grew, available IPv4 addresses eventually became exhausted more than ten years ago. The IETF designed IPv6 with a much larger addressing space consisting of 128-bit addresses, pushing back the exhaustion problem much further in the future.

In this paper, we argue that this large addressing space allows reconsidering how IP addresses are used and enables improving, simplifying and scaling the Internet. By revisiting the IPv6 addressing paradigm, we demonstrate that it opens up several research opportunities that can be investigated today. Hosts can benefit from several IPv6 addresses to improve their privacy, defeat network scanning, improve the use of several mobile access network and their mobility as well as to increase the performance of multicore servers. Network operators can solve the multihoming problem more efficiently and without putting a burden on the BGP RIB, implement Function Chaining with Segment Routing, differentiate routing inside and outside a domain given particular network metrics and offer more fine-grained multicast services.

CCS CONCEPTS

• **Networks** → **Network design principles**; *Network management*;

KEYWORDS

IP address, IPv6, multipath, multihoming, network service

1 INTRODUCTION

While the first design choices of computer networks were made in the 1960s and 1970s [4, 18], many of them still influence the networking industry and the Internet today. During the last decade, the Internet infrastructure has evolved in different ways. We focus on two of these changes.

The ongoing deployment of IPv6 is a first important change [47]. When IPv4 was designed, 32-bit addresses seemed to be an almost unlimited addressing space for the research network they were designed for. However, during the 1990s, network operators and researchers became aware that it would eventually be exhausted. They explored short term solutions such as Network Address Translation (NAT) [25], which were more successful than expected, and IPv6 as a long term solution [12]. When the first IPv6 specifications were published, most observers thought that this new protocol would be quickly embraced by network operators and users. In reality, IPv6 took more than two decades to be widely implemented. The transition to IPv6 continues to progress [47, 62] with more IPv6 capable networks deployed each year.

A second change is the decoupling of transport protocols from the network layer. In the initial design of the host-to-host protocol for the ARPANet [15], the network and the transport layer were strongly coupled. It gave birth to TCP [67] and IPv4 [66] which kept some interdependence as the TCP checksum is computed using a pseudo-header including the source and destination IP addresses. This coupling still exists between TCP and IPv6 [22]. For the past two decades, several transport protocols have been designed and extended to become network-independent with the ability of using several network paths simultaneously. SCTP [77] considered multihomed endpoints as part of its initial design, while TCP was extended to support multiple network paths [31]. More recently, QUIC [45] provided support for multihomed endpoints with connection migration and multiple network paths thanks to Multipath extensions [20, 54, 81].

Nowadays, most applications leverage encryption and authentication to secure their communications. They rarely exchange data in plain text. They use the Transport Layer Security (TLS) [40, 50] protocol either atop TCP or integrated in a transport protocol such

as the QUIC protocol [45, 52]. This change has two consequences. First, the applications that previously relied on IP addresses for authentication are now further decoupled from the network layer. Second, the use of end-to-end encryption strengthens the end-to-end paradigm structuring many protocols of the Internet. This change took more time than expected [9] but works well with the new multipath transport protocols as TLS can be combined with SCTP [79], can be used atop MPTCP, and is integrated in QUIC.

In this paper, we explore how, when combined, these recent advances enable us to reconsider the semantics of the IPv6 addresses. For most network operators and users, IP addresses are used to identify a particular network interface of a device. As such, the main change between IPv6 and IPv4 is the increase to 128-bit addresses from the more constrained 32-bit addresses, enabling a much greater number of interfaces to communicate over the Internet.

We argue that the IPv6 addressing space opens many other interesting directions for future research that go beyond using an IPv6 address to simply identify a network interface. We first analyze in Section 2 how multipath transport protocols and IPv6 addresses can provide new services to end hosts. Then, we demonstrate in Section 3 how networks can effectively solve the multihoming problem using multipath transport protocols and the IPv6 addressing space. Finally, we discuss in Section 4 how routers and middleboxes can benefit from the IPv6 addressing space.

2 MULTIPLE IPV6 ADDRESSES ON HOSTS

In this section, we revisit the IPv6 addressing on hosts to explore how combining multipath transport protocols and IPv6 addresses can solve privacy issues for clients and servers, help cellular networks to scale and improve server performance.

2.1 Privacy-compliant IPv6 addresses

The IPv6 addressing architecture [39] proposed a hierarchical allocation of IPv6 prefixes to Internet Service Providers who then delegate smaller prefixes to their customers. In the initial architecture, the low-order 64 bits of the IPv6 address contained a MAC address [43] to enable autoconfiguration at a time when DHCP was not ubiquitous. Unfortunately, this brought privacy issues since hosts could be easily identified over the Internet using the lower order bits of their IPv6 addresses [46].

Current IPv6 deployments leverage privacy-compliant IPv6 addresses [35, 60, 61] to derive complete addresses from prefixes advertised by routers. In a nutshell, these IPv6 addresses have random low-order 64 bits and limited lifetimes. When one address expires, it is replaced by another one, and both addresses can overlap for some time. This overlap is not a problem since the IPv6 stacks are designed to handle multiple addresses per interface. Yet, this limited lifetime impacts the transport layer. When a temporary address expires and a new one is allocated, all new transport flows automatically use the new address. However, established TCP connections are bound to the address used at connection establishment time and these addresses must persist for the entire connection lifetime. For long-lasting connections such as `ssh`, this implies that temporary addresses need to remain alive for long periods of time. **A multipath transport protocol could easily switch temporary IPv6 addresses without affecting the existing transport connections.**

2.2 Defeating network scanning

Most of the Internet hosts are clients that initiate connections towards servers and usually do not expect to receive unsolicited packets. The servers continuously listen for packets, and their addresses are usually advertised using the DNS. Many enterprise networks restrict the packets that clients can receive to sessions initiated by their clients for security reasons. In the IPv4 Internet, hosts are continuously scanned by network researchers trying to understand how the network is used, worms seeking hosts to infect or attackers looking for vulnerable targets [2]. With the advent of fast network scanners [24], the entire IPv4 addressing space can be scanned within less than a day. IPv6 is harder to scan given its larger space and the sparsity of used addresses within a given prefix. However, this is not sufficient, as researchers have proposed solutions to create target lists of IPv6 addresses [34, 58], and there are now active IPv6 scanners [10, 33].

Some servers use port-knocking to defeat scanners [8]. With port-knocking, the client sends one or more packets towards specific ports or using specific information. The server firewalls validate these packets and finally accept the client's request. The large IPv6 addressing space can be used to improve the privacy of servers. ChhoyHopper [72] proposed to change the IPv6 address of servers every minute. Using a pre-shared key, the client is able to derive the current suffix of the server. The current ChhoyHopper prototype uses NAT to preserve established connections when the server address changes. **A multipath transport protocol would allow clients to migrate to the new server address to improve its privacy.** The Host Identity Protocol (HIP) [57] Rendezvous Extension [51] provides another mechanism to enable the resolution of a host that is often moving. A Rendezvous server could also be used for ephemeral server addresses.

2.3 Multi-addressed cellular networks

Cellular networks typically assume that the IP address of a mobile user must remain stable even when it moves. Otherwise, transport connections identified by a 5-tuple are impacted every time an address changes. For this reason, these networks rely several data-link and network layers solution, e.g., IP tunnels, to hide the user's mobility to their devices. Mobile networks operators assume that the mobile devices are unintelligent, and place intelligence inside the network with a lot of components handling the user's mobility [56]. This contrasts with the Internet's end-to-end principle [74].

As shown by Croitoru et al. [19], by leveraging multipath transport protocols it is possible for a mobile device to seamlessly use IP addresses that correspond to its geographical location and transparently switch from one to another as it moves. This approach could simplify the architecture of mobile networks. Base stations and access points would simply need to verify the credentials of the mobile device, allocate IP addresses from their local address block and forward packets. **Multipath transport protocols allow mobile devices to seamlessly move from one access point (resp. base station) to another without impacting the existing**

transport connections. Such an architecture is also discussed by the CellBricks proposal [56].

Cellular networks could even go one step further when mobile devices can simultaneously use different radio frequencies. Modern cellular networks such as 5G can operate at frequencies in different bands, from roughly 700 MHz to 26 GHz. The lower frequencies provide wide coverage at a relatively low bandwidth while the higher frequencies provide very high throughput within a short radius around the base station. Consider for example the 700 MHz and the 26 GHz bands. Instead of being allocated one frequency band by the cellular network, the mobile device could use both simultaneously by receiving an IP address for each band. The 700-MHz IP address could then be used to initiate outbound and accept inbound connections since this address would always be active. It could also be used for long-lasting flows that exchange few data. On the other hand, the 26-GHz IP address could be used for short and high bandwidth flows. **Multipath transport protocols allow a mobile device to simultaneously use the two bands given the application requirements and dynamically switch transport connections from one frequency band to another when moving.**

2.4 Multicore dataplane

Since the design of IPv4, computers saw the arrival of multicore CPUs and servers can now reach up to hundreds of CPU cores. However, the network layer was designed with one address identifying a given network interface. A multicore server is likely to use a given IP address on several CPU cores. To spread the load between cores, techniques such as RSS [44] hash the 4-tuple to select a core for every connection. However, this can lead to a high load imbalance [6, 65]. The large IPv6 addressing space enables assigning one IPv6 address to each CPU core without using a hash.

This can be efficiently implemented in commodity NICs as they can handle IP-to-queue dispatching in hardware. Every packets can be directed to the destination CPU core to enable a more efficient sharding approach [48]. This approach does not require dispatching cores, i.e., cores responsible for selecting the destination CPU, for which Metron [49] has shown they are an impediment to hundred-gigabit speeds. Metron uses an agent on every host, and a controller to tag packets into switches according to their traffic classes. Then, the NIC dispatch packets to the right core using the tag.

When assigning IPv6 addresses to cores, it enables load-balancing directly at the network layer, for instance by using DNS or standard network load-balancers without any specific controllers. Figure 1 reports the distribution of the total throughput obtained by 128 clients making repeated 2 MB requests in one-time QUIC connections [5] towards a `picoquic` server using an increasing number of cores to serve those requests. We modified `picoquic` to use DPDK for I/O [7], enabling faster speeds and a finer control on the CPU packet processing pipeline and connection state. In the "Single IP" case, the clients reach the server with a unique destination IPv6 address. The server uses RSS to dispatch packets to cores using the NIC's internal hashing algorithm [44]. In the "One IP per core" approach, the clients use the DNS to randomly select one of the server IPv6 addresses. This can be done easily by announcing all the server addresses under a unique domain name. We added ~ 30 lines of code to program the NIC to deliver packets of each IPv6

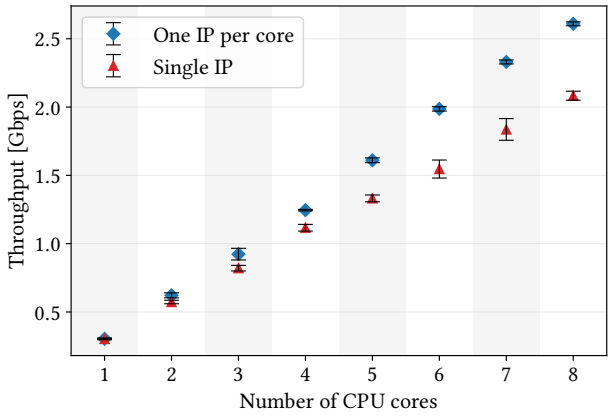


Figure 1: Having one IP per core enables a better load-balancing than hashing packet identifiers.

address to their corresponding core. This better distribution enables up to a 25 % performance increase when using 8 cores, as clients hit less often overloaded cores. **Allocating one IPv6 address per core improves the load balancing and performance of transport connections.** Further improvements could be obtained for instance by adding intelligence in the DNS server to direct requests on lightly loaded cores and overcome long-term imbalance. Cores could also receive multiple IP addresses and exchange them with less loaded cores when overloaded to overcome short-term overloads as proposed with hash buckets by RSS++ [6], or use work stealing as proposed by Affinity-Accept [65].

Our idea could be expanded to processes of an operating system. **IPv6 addresses could be used to identify processes running on a computer**. The operating system would provide each running process with one or more IPv6 addresses to communicate on the network. This could improve the security isolation between processes as the NIC could ensure that packets of each IP address are assigned to a separate queue residing in a different memory zone, which could be a secure enclave in the future.

2.5 Per-connection IPv6 addresses

QUIC identifies connections with a `Connection ID` to support load-balancers [45]. QUIC uses variable length connection IDs to support different types of load-balancers. When each host disposes of a local IPv6 subnet derived from the global network IPv6 prefix, the `Connection ID` could be embedded in the IPv6 address suffix. As QUIC allows the server to migrate with the `Preferred Address` mechanism, the handshake could be performed on one IPv6 address and the connection could then be migrated to addresses embedding the `Connection ID`. This allows a QUIC connection to directly reach a server behind a load balancer after the QUIC handshake. Multipath TCP also provides a similar mechanism [23, 32] by enabling the server to specify the address that the client needs to use for subsequent subflows. **Using IPv6 addresses to embed the QUIC `Connection ID` simplifies the protocol parsing and routing, and reclaims some bytes in the packet for application data.** Similarly, we question the need of transport ports as a

necessary commodity. As a destination address directly identifies a connection, further bytes could be reclaimed in the transport header and reused for application data by removing ports. A separate IPv6 address could be allocated for each service to accept incoming connections, which would then migrate to their own IPv6 address identifying the connection. From a deployment perspective, this also releases pressure on the port number usage, with several processes being able to run several programs using the same port number on the same operating system while assigned to different IPv6 addresses.

3 HOST-BASED IPV6 MULTIHOMING

Scalability is a key concern for network operators. Within five decades, the Internet grew from a research network to a critical infrastructure connecting more than 70 k networks called Autonomous Systems (AS) [42]. Each AS uses the Border Gateway Protocol (BGP) [70] to advertise routes towards its IP prefixes and exchange routing information with its peers. BGP is used by millions of Internet routers and the scalability of the global Internet routing depends on various factors including: (*i)* the number of ASes, (*ii)* the number of prefixes advertised by each AS and (*iii)* the stability of the nodes and links.

The Internet is mostly composed of two types of ASes: transit and stub. A transit AS is typically an Internet Service Provider (ISP) that is connected to larger provider ASes and forwards packets on behalf of its customers and . A stub AS is a network that serves only its hosts and does not forward packets on behalf of its peers. Most enterprise networks are stub ASes.

As of early 2022, there are about 10.8 k transit ASes and 62 k stub ASes [42]. In average, an AS advertises more than 2 IPv6 and 13 IPv4 prefixes while the mean advertised prefix length is 24 bits for IPv4 and 47 for IPv6. In theory, each AS should advertise a single prefix per IP version. However, this does not happen in practice for several reasons. First, the IPv4 addressing space is heavily fragmented [16, 76]. As almost all IPv4 prefixes have been assigned, an AS that needs new IPv4 addresses buys small blocks of addresses from other ASes. Second, some ASes announce short IPv4 prefixes to minimize the impact of potential hijacking attacks. Third, network operators often split their IP prefixes to advertise them differently to different peers for traffic engineering purposes [41].

With the ongoing deployment of IPv6, the first factor will become less important. The deployment of secure BGP extensions [13, 53] could alleviate the second problem. However, the traffic engineering problem is likely to remain as an AS cannot advertise a large IP prefix with BGP and control its incoming traffic (e.g., to ensure that it is balanced among its different peers and links) [27, 78].

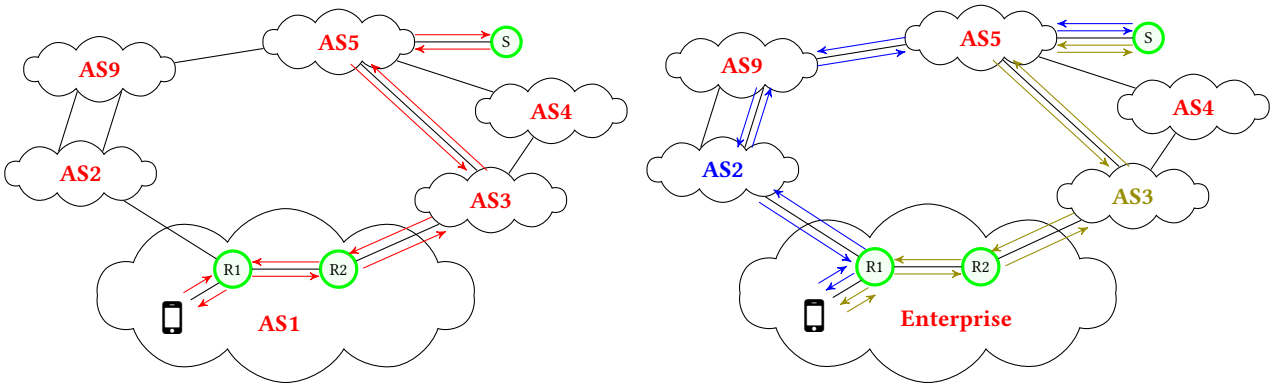
For enterprise networks, site multihoming is a very important traffic engineering problem. While most enterprise networks are usually first connected to the Internet using one link and one provider, many enterprise networks use several providers to cope with a provider failure. These are called multihomed networks. In addition, multihomed networks have a benefit in terms of performance [1, 59] as different providers expose different paths with differing performances often differ.

Traditionally, such a multihomed network is identified by one AS number and uses BGP to advertise its prefix to the global Internet. Unfortunately, each multihomed enterprise network contributes to the growth of the BGP routing tables. However, the multihoming problem can also be solved with a host-based solution that does not pollutes the BGP routing tables with enterprise prefixes. To achieve it, an enterprise network should receive one provider-aggregatable (PA) network prefix from each of its providers as part of their larger IP prefix. This enables each provider to advertise one large prefix that includes all its customers. These different prefixes received by an enterprise network are then distributed so that each of its hosts receives one address from each provider. This is the host-based solution to the site multihoming problem [3]. It has the advantage of requiring fewer BGP advertisements.

The difference between BGP-based multihoming, used with IPv4, and host-based multihoming is illustrated in Figure 2. Figure 2a shows a dual-homed enterprise network named `AS1`. `AS1` advertises its prefix and learns the server prefix using BGP. Both `AS1` and `AS5` select one interdomain path to reach the server and the client prefixes (shown in red in the figure). When an interdomain link fails, e.g., `AS1–AS3`, BGP will take seconds or more to converge on other interdomain paths. Figure 2b shows the interdomain paths used with host-based IPv6 multihoming. In this case, the client has two IP addresses, one from `AS2` and one from `AS3`. The entreprise network does not need to have an AS number and advertise BGP routes since its prefixes are covered by the prefixes advertised by its providers. The main benefit is that the client and the server can use different interdomain paths to communicate [21]. Multipath transport protocols could simultaneously use both paths or automatically select the best performing one matching the application requirements. When the link between the enterprise network and `AS3` fails, the client and the server can immediately continue to exchange data using the path via `AS2`.

There is an additional benefit to host-based multihoming when considering the number of BGP messages that are exchanged. Indeed, according to RIPE RIS [71], more than 2 millions BGP messages are exchanged each hour in 2022. This amount of messages can overload routers. Moreover, reducing the number of BGP messages could help to mitigate the appearance of BGP zombies. Zombies are unused prefixes that could induce some routing loops [64]. Indeed, since processing all the BGP messages could be too time-consuming for a router during a spike of BGP messages, some messages could be dropped and thus lead to the appearance of BGP zombies. Figure 3 shows the percentage of BGP update messages corresponding to stub ASes over the past 10 years. We randomly sampled one thousand periods of 5 minutes for each year, and computed the proportion of BGP updates originating from stub ASes. A stub AS is defined as having a customer cone of size 1, according to the CAIDA asrank dataset [14]. We can observe that in average, as represented by the dashed curves, 43 % of IPv4 BGP updates and 46 % of IPv6 BGP updates are coming from stub ASes. A solution where stubs use addresses from their providers could reduce the BGP churn by a similar percentage. We further note that while the Internet has grown over time, the results in Figure 3 show that this percentage is rather stable over time.

Another system used by ASes when announcing their prefixes is the Resource Public Key Infrastructure (RPKI). RPKI allows network



(a) With BGP-based IPv4 multihoming, the client and the server use one interdomain path.

(b) With Host-based IPv6 multihoming, the client and the server can use one path per provider of `Enterprise`. This provides more diversity and resilience for the interdomain paths.

Figure 2: Comparison of BGP-based IPv4 multihoming and host-based IPv6 multihoming.

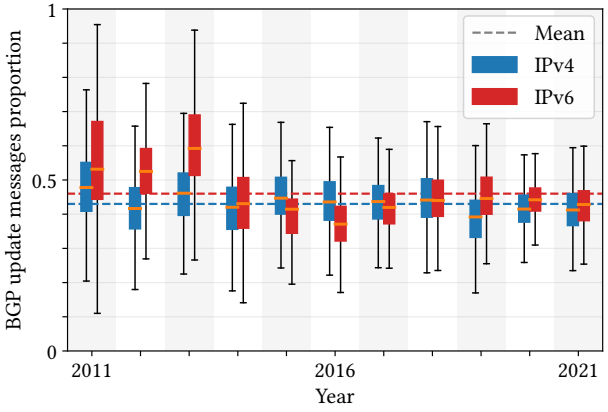


Figure 3: BGP update messages relative to stub prefixes. In average, 43 % of the messages concern stub prefixes in IPv4, while in IPv6 the mean corresponds to 46 % of the messages.

operators to protect their prefixes against inadvertent advertisement by some other AS. However, registering Route Origin Advertisements (ROA) in the RPKI is not globally adopted. Today, the adoption of RPKI is close to 25 % of all ASes, meaning that each of these ASes has at least one RPKI record.[1]. Further, not all ASes validate the BGP advertisement against this information. Misconfigurations resulting in origin hijacks still occur almost daily on the Internet. Since multihoming dispense stubs AS from announcing their own prefixes, they also do not need to register them in the RPKI. Not counting the stubs ASes, the RPKI adoption rate increases to 54 %, leading to a potentially safer Internet.

In conclusion, combining provider-aggregatable IPv6 prefixes with multipath transport protocols solves the multihoming problem more efficiently and without overloading BGP. Hosts can react much faster to the failure of a provider and can balance their load across several network paths.

[1]We observe this rate of adoption from the data available at https://rpki.gin.ntt.net/api/export.json.

4 MULTIPLE IPV6 PREFIXES INSIDE NETWORKS

Today, the IPv6 address space remains vastly unused, as only 0.0034 % of its space is advertised [42]. In this section, we reconsider the use of IPv6 prefixes inside networks. We explore various use-cases enabled by associating IPv6 prefixes to network-level services.

4.1 Segment Routing and Function Chaining

Segment Routing (SR) [30] is a modern variant of Source Routing. With SR, a packet embeds an ordered list of instructions called *segments*. These instructions can be executed on intermediate routers in the network for several service and topological purposes. Segment Routing does not require per-flow state on routers as the instruction is embedded in the packet.

The two predominant variants of SR are MPLS and IPv6 [30]. With IPv6 Segment Routing (SRv6) [29], a segment is encoded as an IPv6 address. This large space enables flexible instruction encoding to support the Network Programming paradigm [29]. An IPv6 address is broken down into `Locator:Function:Arguments`. The `Locator` uniquely identifies the router in the network (typically with a `/48` or `/64` prefix). The two remaining fields encode the function that the router executes on the incoming packet. Researchers and network operators have used network programming for several purposes [30, 80, 85].

Steering traffic through middleboxes in enterprise networks can be a daunting task, especially when it has to follow a specific order [69]. Service Function Chaining (SFC) [36] proposes an architecture allowing easier traffic steering through middleboxes, also called Service Functions (SF).

SRv6 over Network Programming is flexible enough to provide a generic answer to the Service Chaining issue. Clad et al. [17] propose to associate Segment IDs (SIDs) to SFs. This way, using SR to steer the traffic among the services is quite straightforward. We push this idea one step further by proposing to gather related

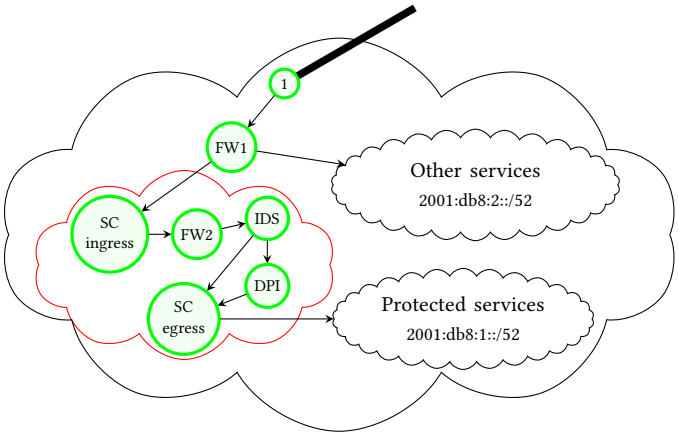
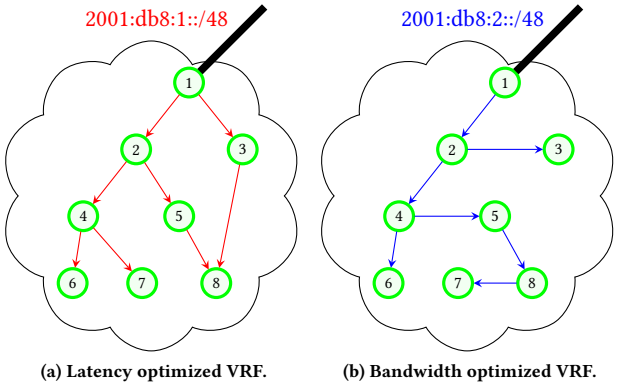


Figure 4: Illustration of a security-oriented Service Function Chain (SFC).



(a) Latency optimized VRF. **(b) Bandwidth optimized VRF.**

Figure 5: Routes from border router (Node 1) to internal nodes.

services in a same SRv6 domain. According to the SRv6 specification [29], such domain could be assigned a given IPv6 prefix (e.g., a `/48` prefix) and the nodes (i.e., the services) belonging to this domain could receive unique subnets (e.g., `/64`) derived from this prefix. This can be easily achieved thanks to the flexibility of IPv6. Figure 4 illustrates an example of such a Service Chain used to secure incoming traffic according to its destination in the network. In our example, an enterprise uses one prefix (`2001:db8:1::/52`) for protected services and another one (`2001:db8:2::/52`) for other services. The `/64` prefixes are further distributed among the specific services inside these two subdomains. The SR domain delimiting the Service Chain is represented as a red cloud and receives the `2001:db8::/48` prefix. When the first firewall (FW1 node in Figure 4) detects traffic destined to the protected services, it forwards it to the Service Chain ingress node. This node (i.e., the SR domain head-end) encodes the SR Policy representing the following security Service Chain. The traffic is forced through a second firewall (FW2 in Figure 4). When the traffic is not dropped, it is forwarded to the IDS. Depending on the IDS decision, the packets are sent to the DPI or directly to the SC egress. On the DPI node, the packets are either dropped or forwarded to the SC egress. Finally, the SC egress removes the SR Policy encapsulation. The traffic successfully passing through the security chain is forwarded to the protected services. **Distributing several IPv6 prefixes inside networks allows defining fine-grained packet steering policies in the Source Routing paradigm, enabling complex network-level service chains.**

4.2 Differentiated routing

Large ASes typically use link-state routing protocols such as IS-IS or OSPFv3. These protocols rely on two main components: *(i)* a link-state database summarizing the local view of each node, and *(ii)* a path computation algorithm which computes the bests paths to forward IP packets. This algorithm relies on dimensionless metrics associated to each link in the network. One drawback of using a single metric is the lack of flexibility. OSPFv2 was originally designed with the ability of computing separate routes according to the IP Type of Service. This feature has been abandoned in subsequent versions of OSPFv2 and OSPFv3. However, a current effort works on the so-called Flex-Algorithm extension [68], allowing IGPs to compute their routes based on given constraints.

Currently, some operators use multiple routing instances and Virtual Routing and Forwarding tables (VRFs), each optimizing a different metric. While this solves the flexibility issue, it introduces a configuration burden.

With IPv6, operators could make these choices more explicit by associating an IPv6 prefix to each VRF. Each virtual interface belonging to a given VRF should then receive an IPv6 address derived from the prefix associated to the VRF. A network could advertise such prefixes as a new service, ensuring to its users different type of service for each of them. From an operational point of view, this also simplifies the manner VRFs are configured and maintained in routers. Figure 5 illustrates such configuration. The network uses prefix `2001:db8:1::/48` for a latency optimized VRF (Figure 5a) and prefix `2001:db8:2::/48` for a bandwidth optimized VRF (Figure 5b). Clients can then select a specific source address to obtain a given forwarding behavior. Such prefix differentiation could also have a positive impact in the global routing. **Associating IPv6 prefixes to virtual topologies could allow more flexible packet forwarding depending on various operators-defined optimization policies.**

4.3 Multicast

Multicast can also benefit from IPv6 addresses to simplify the operation of routers. The IPv6 multicast addresses [38] include flags and an explicit scope. This contrasts with IPv4 multicast where the scope is implicitly encoded in the TTL. The initial IPv4 multicast architecture defined a flat-space for group identifiers. Besides a few well-known groups, addresses were allocated using dynamic mechanisms such as SDR [37]. This caused operational problems and the IETF reserved a subset of the IPv4 addressing space (i.e., `233/8`) for so-called IPv4 GLOP addresses that contain an 16-bit AS number and a group identifier managed by the AS. The IPv6

multicast addressing architecture uses a similar approach by embedding the /64 prefix of the organization that allocated the group identifier.

Another difference between IP versions is the support of the Any Source Multicast service model. To support this model, IP networks usually deploy the PIM protocol [28]. PIM uses a Rendez-vous Point (RP) router. In IPv4, the address of the RP needs to be configured on all routers or specific protocols must be deployed to distribute the groups to RP mappings [11, 26]. With IPv6, the address of the RP can simply be embedded inside the IPv6 multicast address [75].

Finally, the large addressing space allows combining multiple multicast groups for a given application. Currently, users register to a single multicast group address for a specific application. The multicast flow offers an identical service to all users registered to the group. Users with different bandwidth capabilities are treated as equal, and this could lead to decreased performance in congestion-controlled communication [73, 82]. **With IPv6, a multicast application could spread its services among several multicast group addresses to offer a more fine-grained user experience**. Each user may register to one or more services depending on its capabilities. For example, a streaming application could send the data to one address, and Forward Error Correction (FEC) packets [55, 63] to another group. A bandwidth-limited user does not need to register to these FEC packets, but other users may benefit from the recovering capabilities to ensure a better communication.

The Bit Indexed Explicit Replication (BIER) goes one step further by reconsidering scalable multicast forwarding [83]. This multicast routing architecture does not require per-flow state on routers. It embeds a *BitString* in each packet and assigns one bit position to each network router. When a router forwards a packet, it considers the bits set in the *BitString* and forward the packets towards the corresponding routers using forwarding state from the underlying routing protocol. This requires some cooperation with the routing protocol and a *BitString* that has as many bits as the number of routers inside the network. This *BitString* must be embedded in a dedicated BIER header [84]. **We could directly embed the *BitString* inside the IPv6 destination address of the multicast packets to reduce the overhead**. This would require sufficient space in the IPv6 address suffix, but could save additional payload bytes by removing the need for the BIER header.

5 DISCUSSION

The design and deployment of IPv6 were primarily motivated by the exhaustion of the IPv4 addressing space. To this end, the IETF succeeded to provide an evolution path to the Internet. The majority of networks support IPv6 and yet, only 0.0034 % of the addressing space is advertised using BGP, leaving several orders of magnitude of growth available. We argue that the sole use of IPv6 to identify network interfaces on devices must be reconsidered and that the large amount of available addresses enables new opportunities.

First, together with multipath transport protocols, hosts can benefit from several IPv6 addresses to improve server privacy, to defeat network scanning, to better utilize mobile access networks and frequency bands, to improve the performance of multicore servers and to reclaim bytes in the protocol headers. Second, the site multihoming problem can be solved using IPv6 provider-aggregatable addresses and multipath transport protocols without polluting the BGP routing tables. It enables these networks to improve their resilience and the Internet to scale. Third, IPv6 prefixes can be used to define Service Function Chains with Segment Routing, to perform differentiated routing optimizing different network metrics and to both simplify and scale the use of multicast.

We encourage researchers to explore the new usages of IPv6 in research works designing, prototyping and evaluating these use-cases. However, more importantly, we argue that the paradigm of addressing network interfaces must be reconsidered by network researchers and hope to see more works exploring the vast possibilities of IPv6.

ACKNOWLEDGEMENTS

The work of Maxime Piraux was partially supported by the NGI-POINTER programme with funding from the European Union's Horizon 2020 research and innovation programme under the Grant Agreement no 825354. Tom Barbette is funded by an FSR Fellowship from UCLouvain. Thomas Alfroy is funded by ArtIC project "Artificial Intelligence for Care" (grant ANR-20-THIA-0006-01) and co funded by Région Grand Est, Inria Nancy - Grand Est, IHU of Strasbourg, University of Strasbourg and University of Haute-Alsace. François Michel is an F.R.S.-FNRS Research Fellow. We thank Quentin De Coninck and Thomas Wirtgen for their comments and suggestions to improve this paper.

REFERENCES

[1] Aditya Akella, Bruce Maggs, Srinivasan Seshan, Anees Shaikh, and Ramesh Sitaraman. 2003. A measurement-based analysis of multihoming. In *SIGCOMM'03*. 353–364.
[2] Mark Allman, Vern Paxson, and Jeff Terrell. 2007. A brief history of scanning. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*. 77–82.
[3] F. Baker, C. Bowers, and J. Linkova. 2019. Enterprise Multihoming using Provider-Assigned IPv6 Addresses without Network Prefix Translation: Requirements and Solutions. RFC 8678 (Informational). https://doi.org/10.17487/RFC8678
[4] Paul Baran. 2002. The beginnings of packet switching: some underlying concepts. *IEEE Communications Magazine* 40, 7 (2002), 42–48.
[5] Tom Barbette. 2022. *WRK-MultiProtocol*. Retrieved June 1, 2022 from https://github.com/tbarbette/wrk-quic
[6] Tom Barbette, Georgios P Katsikas, Gerald Q Maguire Jr, and Dejan Kostić. 2019. RSS++ load and state-aware receive side scaling. In *Proceedings of the 15th international conference on emerging networking experiments and technologies*. 318–333.
[7] Tom Barbette and Nikita Tyunyayev. 2022. *picoquic-dpdk*. Retrieved June 1, 2022 from https://github.com/IPNetworkingLab/picoquic-dpdk
[8] Paul Barham, Steven Hand, Rebecca Isaacs, Paul Jardetzky, Richard Mortier, and Timothy Roscoe. 2002. Techniques for lightweight concealment and authentication in IP networks. *Intel Research Berkeley. July* (2002).
[9] Steven M Bellovin. 1989. Security problems in the TCP/IP protocol suite. *ACM SIGCOMM Computer Communication Review* 19, 2 (1989), 32–48.
[10] Robert Beverly, Ramakrishnan Durairajan, David Plonka, and Justin P Rohrer. 2018. In the IP of the beholder: Strategies for active IPv6 topology discovery. In *Proceedings of the Internet Measurement Conference 2018*. 308–321.
[11] N. Bhaskar, A. Gall, J. Lingard, and S. Venaas. 2008. Bootstrap Router (BSR) Mechanism for Protocol Independent Multicast (PIM). RFC 5059 (Proposed Standard). https://doi.org/10.17487/RFC5059
[12] S. Bradner and A. Mankin. 1993. IP: Next Generation (IPng) White Paper Solicitation. RFC 1550 (Informational). https://doi.org/10.17487/RFC1550
[13] R. Bush. 2017. BGPsec Operational Considerations. RFC 8207 (Best Current Practice). https://doi.org/10.17487/RFC8207
[14] CAIDA AS Rank 2022. *CAIDA AS Rank*. Retrieved May 31, 2022 from https://as-rank.caida.org/
[15] Vinton Cerf and Robert Kahn. 1974. A protocol for packet network intercommunication. *IEEE Transactions on communications* 22, 5 (1974), 637–648.
[16] Luca Cittadini, Wolfgang Mühlbauer, Steve Uhlig, Randy Bush, Pierre Francois, and Olaf Maennel. 2010. Evolution of Internet address space deaggregation:

myths and reality. *IEEE Journal on Selected Areas in Communications* 28, 8 (2010), 1238–1249.

[17] Francois Clad, Xiaohu Xu, Clarence Filsfils, Daniel Bernier, Cheng Li, Bruno Decraene, Shaowen Ma, Chaitanya Yadlapalli, Wim Henderickx, and Stefano Salsano. 2021. *Service Programming with Segment Routing*. Internet-Draft draft-ietf-spring-sr-service-programming-05. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-spring-sr-service-programming-05 Work in Progress.

[18] Stephen D Crocker. 2019. The ARPAnet and its impact on the state of networking. *Computer* 52, 10 (2019), 14–23.

[19] Andrei Croitoru, Dragos Niculescu, and Costin Raiciu. 2015. Towards wifi mobility without fast handover. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 219–234.

[20] Quentin De Coninck and Olivier Bonaventure. 2017. Multipath QUIC: Design and evaluation. In *Proceedings of the 13th international conference on emerging networking experiments and technologies*. 160–166.

[21] Cedric De Launois, Bruno Quoitin, and Olivier Bonaventure. 2006. Leveraging network performance with IPv6 multihoming and multiple provider-dependent aggregatable prefixes. *Computer Networks* 50, 8 (2006), 1145–1157.

[22] S. Deering and R. Hinden. 1998. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard). https://doi.org/10.17487/RFC2460

[23] Fabien Duchene and Olivier Bonaventure. 2017. Making Multipath TCP friendlier to load balancers and anycast. In *2017 IEEE 25th International Conference on Network Protocols (ICNP)*. IEEE, 1–10.

[24] Zakir Durumeric, Eric Wustrow, and J Alex Halderman. 2013. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *22nd USENIX Security Symposium (USENIX Security 13)*. 605–620.

[25] K. Egevang and P. Francis. 1994. The IP Network Address Translator (NAT). RFC 1631 (Informational). https://doi.org/10.17487/RFC1631

[26] D. Farinacci and Y. Cai. 2006. Anycast-RP Using Protocol Independent Multicast (PIM). RFC 4610 (Proposed Standard). https://doi.org/10.17487/RFC4610

[27] Nick Feamster, Jay Borkenhagen, and Jennifer Rexford. 2003. Guidelines for interdomain traffic engineering. *ACM SIGCOMM Computer Communication Review* 33, 5 (2003), 19–30.

[28] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, and L. Zheng. 2016. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761 (Internet Standard). https://doi.org/10.17487/RFC7761

[29] C. Filsfils (Ed.), P. Camarillo (Ed.), J. Leddy, D. Voyer, S. Matsushima, and Z. Li. 2021. Segment Routing over IPv6 (SRv6) Network Programming. RFC 8986 (Proposed Standard). https://doi.org/10.17487/RFC8986

[30] C. Filsfils (Ed.), S. Previdi (Ed.), L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir. 2018. Segment Routing Architecture. RFC 8402 (Proposed Standard). https://doi.org/10.17487/RFC8402

[31] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure. 2013. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 6824 (Experimental). https://doi.org/10.17487/RFC6824

[32] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch. 2020. TCP Extensions for Multipath Operation with Multiple Addresses. RFC 8684 (Proposed Standard). https://doi.org/10.17487/RFC8684

[33] Kensuke Fukuda and John Heidemann. 2018. Who knocks at the IPv6 door? detecting IPv6 scanning. In *Proceedings of the Internet Measurement Conference 2018*. 231–237.

[34] Oliver Gasser, Quirin Scheitle, Pawel Foremski, Qasim Lone, Maciej Korczyński, Stephen D Strowes, Luuk Hendriks, and Georg Carle. 2018. Clusters in the expanse: Understanding and unbiasing IPv6 hitlists. In *Proceedings of the Internet Measurement Conference 2018*. 364–378.

[35] F. Gont, S. Krishnan, T. Narten, and R. Draves. 2021. Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6. RFC 8981 (Proposed Standard). https://doi.org/10.17487/RFC8981

[36] J. Halpern (Ed.) and C. Pignataro (Ed.). 2015. Service Function Chaining (SFC) Architecture. RFC 7665 (Informational). https://doi.org/10.17487/RFC7665

[37] M. Handley, C. Perkins, and E. Whelan. 2000. Session Announcement Protocol. RFC 2974 (Experimental). https://doi.org/10.17487/RFC2974

[38] R. Hinden and S. Deering. 2006. IP Version 6 Addressing Architecture. RFC 4291 (Draft Standard). https://doi.org/10.17487/RFC4291

[39] R. Hinden (Ed.) and S. Deering (Ed.). 1995. IP Version 6 Addressing Architecture. RFC 1884 (Historic). https://doi.org/10.17487/RFC1884

[40] Ralph Holz, Jens Hiller, Johanna Amann, Abbas Razaghpanah, Thomas Jost, Narseo Vallina-Rodriguez, and Oliver Hohlfeld. 2020. Tracking the deployment of TLS 1.3 on the Web: A story of experimentation and centralization. *ACM SIGCOMM Computer Communication Review* 50, 3 (2020), 3–15.

[41] Geoff Huston. 2017. BGP more specifics: routing vandalism or useful? (June 2017). https://blog.apnic.net/2017/06/26/bgp-specifics-routing-vandalism-useful/.

[42] Geoff Huston. 2022. BGP in 2021 – The BGP Table. (Jan 2022). https://www.potaroo.net/ispcol/2022-01/bgp2021.html.

[43] IEEE. [n.d.]. Guidelines for 64-bit Global Identifier (EUI-64) Registration Authority.

[44] Intel. 2016. *Receive-Side Scaling (RSS)*. Retrieved May 18, 2022 from https://www.intel.com/content/dam/support/us/en/documents/network/sb/318483001us2.pdf

[45] J. Iyengar (Ed.) and M. Thomson (Ed.). 2021. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000 (Proposed Standard). https://doi.org/10.17487/RFC9000

[46] Said Jawad Saidi, Oliver Gasser, and Georgios Smaragdakis. 2022. One Bad Apple Can Spoil Your IPv6 Privacy. *ACM SIGCOMM Computer Communication Review* 52, 2 (2022).

[47] Siyuan Jia, Matthew Luckie, Bradley Huffaker, Ahmed Elmokashfi, Emile Aben, Kimberly Claffy, and Amogh Dhamdhere. 2019. Tracking the deployment of IPv6: Topology, routing and performance. *Computer Networks* 165 (2019), 106947.

[48] Rishi Kapoor, George Porter, Malveeka Tewari, Geoffrey M Voelker, and Amin Vahdat. 2012. Chronos: Predictable low latency for data center applications. In *Proceedings of the Third ACM Symposium on Cloud Computing*. 1–14.

[49] Georgios P Katsikas, Tom Barbette, Dejan Kostic, Rebecca Steinert, and Gerald Q Maguire Jr. 2018. Metron: NFV Service Chains at the True Speed of the Underlying Hardware. In *Proc. USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 171–186.

[50] Platon Kotzias, Abbas Razaghpanah, Johanna Amann, Kenneth G Paterson, Narseo Vallina-Rodriguez, and Juan Caballero. 2018. Coming of age: A longitudinal study of tls deployment. In *Proceedings of the Internet Measurement Conference 2018*. 415–428.

[51] J. Laganier and L. Eggert. 2016. Host Identity Protocol (HIP) Rendezvous Extension. RFC 8004 (Proposed Standard). https://doi.org/10.17487/RFC8004

[52] Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, et al. 2017. The quic transport protocol: Design and internet-scale deployment. In *Proceedings of the conference of the ACM special interest group on data communication*. 183–196.

[53] M. Lepinski (Ed.) and K. Sriram (Ed.). 2017. BGPsec Protocol Specification. RFC 8205 (Proposed Standard). https://doi.org/10.17487/RFC8205

[54] Yanmei Liu, Yunfei Ma, Quentin De Coninck, Olivier Bonaventure, Christian Huitema, and Mirja Kühlewind. 2022. *Multipath Extension for QUIC*. Internet-Draft draft-ietf-quic-multipath-01. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/draft-ietf-quic-multipath-01 Work in Progress.

[55] Michael Luby, Lorenzo Vicisano, Jim Gemmell, Luigi Rizzo, M Handley, and Jon Crowcroft. 2002. *The use of forward error correction (FEC) in reliable multicast*. Technical Report. RFC 3453, December.

[56] Zhihong Luo, Silvery Fu, Mark Theis, Shaddi Hasan, Sylvia Ratnasamy, and Scott Shenker. 2021. Democratizing cellular access with CellBricks. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*. 626–640.

[57] R. Moskowitz (Ed.), T. Heer, P. Jokela, and T. Henderson. 2015. Host Identity Protocol Version 2 (HIPv2). RFC 7401 (Proposed Standard). https://doi.org/10.17487/RFC7401

[58] Austin Murdock, Frank Li, Paul Bramsen, Zakir Durumeric, and Vern Paxson. 2017. Target generation for internet-wide IPv6 scanning. In *Proceedings of the 2017 Internet Measurement Conference*. 242–253.

[59] Ryo Nakamura, Kazuki Shimizu, Teppei Kamata, and Cristel Pelsser. 2022. A First Measurement with BGP Egress Peer Engineering. In *Passive and Active Measurement - 23th International Conference, PAM 2022*.

[60] T. Narten and R. Draves. 2001. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 3041 (Proposed Standard). https://doi.org/10.17487/RFC3041

[61] T. Narten, R. Draves, and S. Krishnan. 2007. Privacy Extensions for Stateless Address Autoconfiguration in IPv6. RFC 4941 (Draft Standard). https://doi.org/10.17487/RFC4941

[62] Mehdi Nikkhah and Roch Guérin. 2015. Migrating the internet to IPv6: An exploration of the when and why. *IEEE/ACM Transactions on Networking* 24, 4 (2015), 2291–2304.

[63] Jörg Nonnenmacher and Ernst W Biersack. 1996. Reliable multicast: Where to use FEC. In *International Workshop on Protocols for High Speed Networks*. Springer, 134–148.

[64] Porapat Ongkanchana, Romain Fontugne, Hiroshi Esaki, Job Snijders, and Emile Aben. 2021. Hunting BGP Zombies in the Wild. In *Proceedings of the Applied Networking Research Workshop* (Virtual Event, USA) *(ANRW '21)*. Association for Computing Machinery, New York, NY, USA, 1–7. https://doi.org/10.1145/3472305.3472315

[65] Aleksey Pesterev, Jacob Strauss, Nickolai Zeldovich, and Robert T Morris. 2012. Improving network connection locality on multicore systems. In *Proceedings of the 7th ACM european conference on Computer Systems*. 337–350.

[66] J. Postel. 1981. Internet Protocol. RFC 791 (Internet Standard). https://doi.org/10.17487/RFC0791

[67] J. Postel. 1981. Transmission Control Protocol. RFC 793 (Internet Standard). https://doi.org/10.17487/RFC0793

[68] Peter Psenak, Shraddha Hegde, Clarence Filsfils, Ketan Talaulikar, and Arkadiy Gulko. 2022. *IGP Flexible Algorithm*. Internet-Draft draft-ietf-lsr-flex-algo-20. Internet Engineering Task Force. https://datatracker.ietf.org/doc/html/

draft-ietf-lsr-flex-algo-20 Work in Progress.
[69] P. Quinn (Ed.) and T. Nadeau (Ed.). 2015. Problem Statement for Service Function Chaining. RFC 7498 (Informational). https://doi.org/10.17487/RFC7498
[70] Y. Rekhter (Ed.), T. Li (Ed.), and S. Hares (Ed.). 2006. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard). https://doi.org/10.17487/RFC4271
[71] RIPE RIS 2022. The RIPE Routing Information Services. Retrieved May 31, 2022 from http://www.ris.ripe.net.
[72] A S M Rizvi and John Heidemann. 2022. Chhoyhopper: A Moving Target Defense with IPv6. In *4th Workshop on Measurements, Attacks, and Defenses for the Web (MADWeb 2022)*.
[73] Luigi Rizzo. 2000. pgmcc: a TCP-friendly single-rate multicast congestion control scheme. *ACM SIGCOMM Computer Communication Review* 30, 4 (2000), 17–28.
[74] Jerome H Saltzer, David P Reed, and David D Clark. 1984. End-to-end arguments in system design. *ACM Transactions on Computer Systems (TOCS)* 2, 4 (1984), 277–288.
[75] P. Savola and B. Haberman. 2004. Embedding the Rendezvous Point (RP) Address in an IPv6 Multicast Address. RFC 3956 (Proposed Standard). https://doi.org/10.17487/RFC3956
[76] Joao Luis Sobrinho, Laurent Vanbever, Franck Le, and Jennifer Rexford. 2014. Distributed Route Aggregation on the Global Network. In *ACM CoNEXT 2014*. Sydney, Australia.
[77] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. 2000. Stream Control Transmission Protocol. RFC 2960 (Proposed Standard). https://doi.org/10.17487/RFC2960
[78] Florian Streibelt, Franziska Lichtblau, Robert Beverly, Anja Feldmann, Cristel Pelsser, Georgios Smaragdakis, and Randy Bush. 2018. BGP communities: Even more worms in the routing can. In *Proceedings of the Internet Measurement Conference 2018*. 279–292.
[79] M. Tuexen, R. Stewart, R. Jesup, and S. Loreto. 2017. Datagram Transport Layer Security (DTLS) Encapsulation of SCTP Packets. RFC 8261 (Proposed Standard). https://doi.org/10.17487/RFC8261
[80] Pier Luigi Ventre, Stefano Salsano, Marco Polverini, Antonio Cianfrani, Ahmed Abdelsalam, Clarence Filsfils, Pablo Camarillo, and Francois Clad. 2020. Segment Routing: a comprehensive survey of research activities, standardization efforts, and implementation results. *IEEE Communications Surveys & Tutorials* 23, 1 (2020), 182–221.
[81] Tobias Viernickel, Alexander Froemmgen, Amr Rizk, Boris Koldehofe, and Ralf Steinmetz. 2018. Multipath QUIC: A deployable multipath transport protocol. In *2018 IEEE International Conference on Communications (ICC)*. IEEE, 1–7.
[82] Jörg Widmer and Mark Handley. 2001. Extending equation-based congestion control to multicast applications. In *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*. 275–285.
[83] IJ. Wijnands (Ed.), E. Rosen (Ed.), A. Dolganow, T. Przygienda, and S. Aldrin. 2017. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279 (Proposed Standard). https://doi.org/10.17487/RFC8279
[84] IJ. Wijnands (Ed.), E. Rosen (Ed.), A. Dolganow, J. Tantsura, S. Aldrin, and I. Meilik. 2018. Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks. RFC 8296 (Proposed Standard). https://doi.org/10.17487/RFC8296
[85] Mathieu Xhonneux, Fabien Duchene, and Olivier Bonaventure. 2018. Leveraging eBPF for programmable network functions with IPv6 Segment Routing. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. 67–72.