Computer Networks 108 (2016) 233–259

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Review article

A survey on methods to provide interdomain multipath transmissions

Robert Wójcik[a,*], Jerzy Domżał[a], Zbigniew Duliński[b], Grzegorz Rzym[a], Andrzej Kamisiński[a], Piotr Gawłowicz[a], Piotr Jurkiewicz[a], Jacek Rząsa[a], Rafał Stankiewicz[a], Krzysztof Wajda[a]

[a] Department of Telecommunications, AGH University of Science and Technology, Al. Mickiewicza 30, Kraków 30-059, Poland
[b] Faculty of Physics, Astronomy and Applied Computer Science, Jagiellonian University, Reymonta 4,Kraków 30-059 Poland

ARTICLE INFO

Article history:
Received 31 December 2015
Revised 5 August 2016
Accepted 31 August 2016
Available online 7 September 2016

Keywords:
Routing
Multipath
Load balancing
Interdomain multipath

ABSTRACT

Interdomain routing relies on BGP, which does not allow multipath transmissions. Since there is usually more than one path between any pair of nodes on the Internet, it would be beneficial to have the possibility of using them at the same time. Over the years, many solutions have appeared.

In this survey, we show how 17 different approaches suggest solutions for providing interdomain multipath transmission. We divide presented mechanisms based on their relevance, starting from the most significant (assessed subjectively based on publications) and already available (implemented). Firstly, all the mechanisms are presented at a glance. Afterwards, each mechanism is described in more details. After a coherent presentation of each approach, they are compared, contrasted, and subjectively assessed. The comparison criteria include proposal visibility, additional signalling, mechanism complexity, time scale of operation, provided routing type, and path choice entities or path setup procedure. The goal of the survey is to show that there are numerous approaches to providing interdomain multipath transmissions in current IP-based networks.

1. Introduction

The Internet is a grid of interconnected Autonomous Systems (ASes). Multiple links between ASes increase resilience. These links are used for performing policy-based routing, which improves performance and lowers transit costs. The routing information on the Internet is provided by a Border Gateway Protocol (BGP). Each operator needs to establish business relations with directly connected neighbors. These relations are applied in a form of policy routing which selects routes that may be placed in routing tables in their domain. Routing policies also provide procedures for announcing selected routes to neighbouring ASes. After this first stage of selection, the BGP algorithm chooses one active route to each destination and places it in routing tables.

The standard BGP does not have ability to use several paths to a destination. It operates in such a way that it advertises one path per prefix. Multiple path announcements for a particular prefix can be announced, but only the most recent is taken into account. In the scope of a single BGP session, the previous BGP updates related to the same prefix are overridden by more recent ones. As a result, there can be several paths to a particular destination; however, only one is used for packet transfer. Only one active path between a pair of ASes exists on the level of the Internet.

The BGP was first presented in 1989 (RFC 1163). Since then, networks have evolved, and now, we have many more challenges and possibilities. One of the challenges is to overcome the restriction of BGP and provide multipath transmissions on the Internet. One of the problems is the fact that BGP is a global protocol. For that matter, each inter-domain transmission is controlled by more than one operator, usually by many of them. This means that an operator cannot force multipath transmission on its own. There needs to be a global agreement/understanding of some sort. This is difficult, although the potential benefits are obvious.

Firstly, when failures occur, the traffic can be quickly redirected to alternative path(s) and the network maintains full connectivity. Secondly, the operator has the opportunity to simultaneously use two or more paths between given endpoints in the network, thereby increasing throughput between those points and avoiding network congestions. Further, as there are different performance requirements for particular services on the Internet (e.g., low-delay communication for Voice over IP (VoIP), high throughput available for delay-tolerant applications), traffic can be forwarded via different paths leading to the same AS based on the estimated quality parameters, improving the overall user experience.

* Corresponding author.
E-mail address: robert.wojcik@kt.agh.edu.pl (R. Wójcik).

http://dx.doi.org/10.1016/j.comnet.2016.08.028
1389-1286/

Another problem related to inter-domain multipath routing, is in fact the same one that haunted the source routing mechanism. Operators are reluctant to disclose information about their networks over those that are necessary for assuring basic connectivity. In other words, any mechanism that requires knowledge of neighbouring ASes structure is likely to be rejected.

In [1], we have presented, compared and contrasted the most recognised solutions to providing multipath transmissions inside a domain. These solutions could be implemented by a single operator independently. This means that many of them could be used at once on the Internet. In this survey, we focus on interdomain multipath providing solutions. They differ in the fact that many operators need to implement the same solution for it to provide benefits.

The mentioned potential benefits attract attention to multipath solutions. In the absence of a global multipath solution, a Multipath TCP (MPTCP) [2] protocol was developed to provide similar functionality. MPTCP is a transport layer protocol and is, therefore, almost independent on the network routing which restricts its functionality. Much more can be achieved when network traffic control cooperates with the user. We presented and compared MPTCP in [1], therefore, in this survey we did not include it.

The concept of multipath is not new. Over the years, it was investigated heavily for wireless networks, where multipath transmissions take also another meaning associated with diverse physical signal transmission paths. Many surveys exist on that matter, for example [3], or [4]. For fixed networks, an interesting read is [5]. There the authors took a top-down approach and reviewed various multipath protocols, from various layers and operating at different parts of the Internet. They also described mathematical foundations of the multipath operation. The solutions presented therein are mostly intra-domain; however inter-domain are also there.

An interesting survey is [6] in which multipath solutions are presented from the perspective of congestion control mechanisms. The survey shows congestion control solutions for multipath transport protocols and discusses the multipath congestion control design in order to address the need for some desirable properties including TCP-friendliness, load-balancing, stability and Pareto-optimality. In this survey, we focus on inter-domain multipath solutions and their functionality.

The survey is divided into five parts. Part I introduces the reader to all of the surveyed mechanisms and is designed to cover the most important aspects of each mechanism, including its key ideas, timeframe of development, importance and market availability. Parts II, III and IV cover the technical aspects of each mechanism, starting from the most significant (available and implemented in current devices) (Part II), going through mechanisms which attracted some attention from researchers yet did not go into the implementation phase, (Part III), and finishing with mechanisms that are definitively interesting, but their development stopped after one or two research papers (Part IV). After coherent presentation of each approach, in Part V they are compared, contrasted and subjectively assessed. The comparison criteria include: signalling overhead, mechanism complexity, time scale of operation, provided routing type, path choice entities or path setup procedure. Part V also presents our view of the future of interdomain multipath routing development, including the drive to find an optimal mechanism.

The goal of the survey is that after initial mechanism description in Part I, readers can directly jump to those approaches they find interesting, skipping the ones that are of marginal importance, before reaching comparisons and conclusions. This way, although the survey is long, readers can quickly find the information they seek.

Part I First look

2. Timeline

Fig. 1 shows all the architectures in one timescale. The graphs marks years in which a respective mechanism was developed, which is measured by publications. Readers can form their opinion on the maturity of the mechanism judging by the amount of research conducted upon a solution. The graph also shows market availability, which means that the solution is implemented in available equipment and ready to be used.

Out of all presented approaches, four are available commercially. They are: Generalized Multiprotocol Label Switching (GMPLS), BGP Add-Paths, Locator/ID Separation Protocol (LISP) and Segment Routing.

GMPLS is a complex and multipurpose control plane solution, based on the previous MPLS proposal, extended beyond packet-based to Time Division Multiplexing, wavelength- and fiber-based cases. Since its first inception in 2001, the proposal then stabilized in 2004 [7], then was officially updated in [8]. GMPLS framework is systematically developed and improved. Equipment vendors offer GMPLS-based equipment assuming it to be the indication of technological advancement (e.g., Cisco ASR 9000, Huawei OSN6800, OSN98000, Alcatel-Lucent 7950 Extensible Routing System, XRS, etc.) and also software novelty, e.g., GMPLS software offered by MARBEN company [9].

The BPG Add-Paths extensions has been a subject of modest research interest. However, it is already implemented in many operating systems and it is a subject of many RFCs, created with significant industrial backing. The first draft appeared in May 2002 [10]. In December 2008, the draft is taken over by IETF. Until today, it is frequently updated. In 2010, BGP Add-Paths was implemented by Cisco in their devices.

LISP has been specified by the still-active IETF working group *Lisp-wg*. The first RFC [11] was released in 2013. Since then, an additional eight RFCs have been published, and there are a few active internet-drafts. LISP is currently a mature solution. Research on LISP multipath capabilities is still of interest. A few implementations of LISP currently exist. There is an open source implementation called OpenLISP [12–14]. Another open source implementation is LISPmob, recently re-branded as Open Overlay Router (OOR) [15]. This is a LISP and LISP Mobile Node implementation for Linux, Android and OpenWrt [16]. This implementation is partially supported by Cisco which, in turn, offers a LISP implementation in some of their products offered commercially [17].

Segment Routing (SR) seems to have enormous potential to be an important asset of future networks. SR allows the use of a centralized controller to select a path for a given flow of packets (similarly to Software-Defined Networks, SDN). The open source SR prototype network using SDN OpenFlow and bare-metal hardware was founded by [18]. SR is backed by important vendors, e.g., Cisco Systems, Juniper and Alcatel-Lucent, among others. The standardization process of the SR is quite vivid; however, still the only published documents are drafts. It seems that there is strong interest in development of SR.

Out of all other presented approaches, some attracted more attention and were evaluated more than others.

A New Internet Routing Architecture (NIRA) was proposed in 2003 [19] and evaluated by its inventor in 2007 [20]. NIRA and some of its parts have drawn the attention of researchers, and as a result two publications describing its architecture were cited more than 300 times altogether. Unfortunately, NIRA has not attracted much interest from vendors. Platypus did not develop with commercial implementation. Its research interest is currently marginal, although several papers appeared.

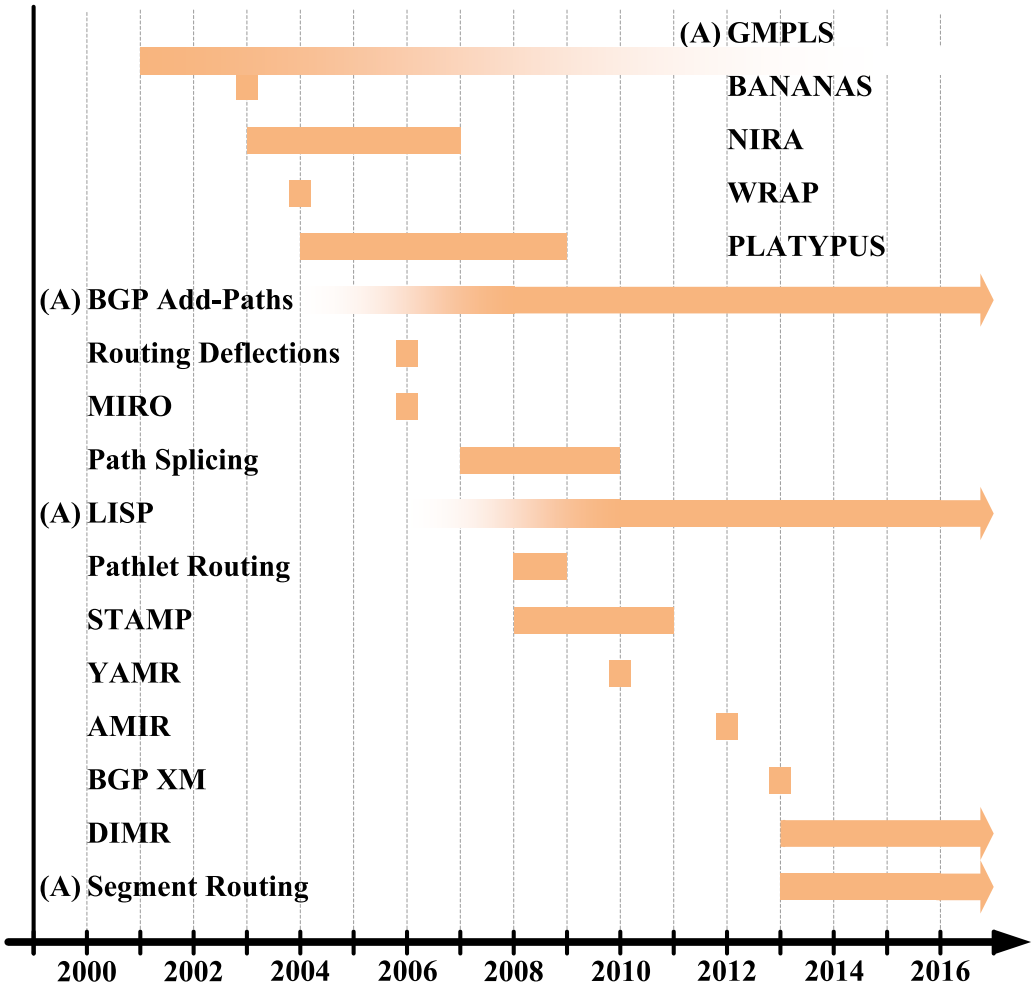


Fig. 1. Inter-domain multipath routing solutions: development history. Bars indicate times of publications on the subject. (A) indicates market availability.

SelecTive Announcement Multi-Process (STAMP), first proposed in 2008, drew attraction from other researchers. However, after some interest in the first years after publishing the first text on this mechanism, it is currently regarded as a non-attractive mechanism, which is why it has not been implemented or developed. Disjoint Interdomain Multipath Routing (DIMR) is a recently proposed solution which enhances a widely known solution - Path Diversity Aware interdomain Routing. Being still a relatively new proposal, DIMR is a promising algorithm, and we believe it may be implemented by network device manufacturers in future.

Other presented approaches, although technically viable and interesting, have not developed over one or two research papers. They did not attract enough attention to be considered for real life implementation. They are not commercially available, and interest faded after their publication.

3. Overview and key features

Table 1 presents the concepts of all the surveyed architectures. In three or four sentences, the idea of the proposal is highlighted. This allows to get an idea on how the respective approach achieves multipath transmissions.

Table 2 shows what is required for each of the presented mechanisms to work. It can be the BGP protocol adaptation, packet encapsulation, packet marking/tagging or any other modifications, and finally, external signalling. Some solutions require more than one from the list. This table points out the direction a respective mechanism ventures into.

GMPLS is a flexible, open, multi-domain and multi-technological control plane concept which supports all needs, current and future. There is a promise, expressed in [7] to consider extensions for inter-domain routing using BGP but no further information is provided, even in [21], dedicated to multi-domain and multi-layer networks. GMPLS, being a direct successor to MPLS, uses packet labelling (form of encapsulation) and external signalling protocols, such as e.g., RSVP-TE.

BGP Add-Paths is a BGP protocol extension. It introduces new type of message exchanged between BGP peers. Traffic packets are not modified or encapsulated. BGP Add-Paths does not require additional signalling, however, both peers must support it in order to be able to exchange additional paths.

Communicating end-hosts uses their IP identifiers as source and destination addresses but, at the border LISP router, packets are encapsulated and sent in the core network using related locator IP addresses as a source and destination. A legacy routing protocol, e.g., BGP is used to route the packet in the core network. LISP uses a signalling protocol to exchange identifier-to-locator mappings.

In the Segment Routing, a node selects a path and encodes the path into a packet header as an ordered list of segments. A path may be denoted in a packet as a stack of MPLS labels or by IPv6 addresses. Taking into account strong focus on the MPLS in the IETF, SR charter as well as it's support for traffic engineering, it

Table 1
Overview of inter-domain multipath solutions.

GMPLS	– provides a key control plane concept allowing for introduction of switching, reliability, traffic grooming and engineering in multiprotocol network – supports on-demand establishment of end-to-end engineered paths (connections) in multi-domain and multi-technological networks – provides universal UNI with flexibly defined requirements and goal functions
BGP Add-Paths	– allows the advertisement of multiple paths through the same peering session for the same prefix without the new paths implicitly replacing any previous paths – is a BGP extension, introduces new message type, so both BGP peers must explicitly support it – allows to achieve greater paths diversity, what can enable faster recovery, suppression of oscillations or load balancing
LISP	– decouples locator and identifier roles of IP address – enables multiple paths between remote hosts by mapping their identifiers to multiple globally routable locator addresses – to take advantage of mutlipath potential: LISP may be used with MPTCP – packets sent between end-hosts (identifiers) are encapsulated and sent in the core network using related locator addresses and legacy routing protocols (e.g. BGP). – signalling protocol is used to exchange identifier-to-locator mappings
Segment Routing	– allows to specify a forwarding path, other than the normal shortest path – source means the point at which the explicit route is set – the forwarding path may be specified by a node which does not originate data
NIRA	– allows users to choose a sequence of ISP for their traffic – groups ASes into two regions: Core and Access – end-to-end route is split into up- and down- graphs and represented using source and destination addresses, respectively – provides mechanisms for route discovery and monitoring
PLATYPUS	– authenticated source routing – uses policy compliance in order to address a problem of routing policy constraints among operators – path selection based on external mechanism (not addressed in the article)
Path Splicing	– uses several different routing trees in a network topology to increase the number of possible paths between the source and destination nodes – forwarding paths may be controlled by end systems using the additional bits in the packet header – offers significant advantages in terms of reliability and deployment cost
STAMP	– runs two BGP processes which are able to compute two complementary disjoint routes – immediately reacts to failures – ensures greater routing stability compared to BGP
DIMR	– simultaneously selected two disjoint paths – prevents packet losses in case of failures – involved ASes must use the BGP protocol to exchange information about the prefixes and AS-level paths
BANANAS	– source routing, path is hashed and encoded in a so called PathID – in Explicit-Exit Forwading mode multipath transmission is realized by choosing dedicated exit ASBR for specified traffic aggregate (per-packet, per-flow, per-prefix, etc.) – in Explicit AS-Path Forwarding mode forwarding realized on arbitrary selected and validated AS-path
WRAP	– source routing based on the Loose Source and Record Route (LSRR) approach specifying end-to-end domain-level path – each edge router computes at least two different AS-paths to each reachable AS – path computation can be based on measurements of QoS parameters
Routing deflections	– packets are tagged – tags indicate which router must deflect the packet from its original path – after deflection, the packet is forwarded normally
MIRO	– stands between source-controlled and network-controlled routing – operators can create tunnels which form alternative paths – created tunnels are advertised and available for end-users to choose from
Pathlet Routing	– vnode - represents an entire AS, part of a network or a node – pathlet - a sequence of vnodes along which the originating node is willing to route
YAMR	– a set of alternate paths are computed – limits signalling traffic after failure – computing many paths results in higher control plane messaging overheads than BGP
AMIR	– obtains the primary path to a destination from the local BGP route table – alternative paths are determined based on the information from other Autonomous Systems – additional paths may be introduced through the negotiation process – relies on different signalling methods (custom packet header, external signalling)
BGP-XM	– a few paths to the same destination can be used concurrently, they can traverse different ASes, they can have different length – the mechanism exploits standard BGP procedures but the path selection algorithm is used in a new way – the mechanism exploits mainly features of AS_PATH (BGP atribute) represented in a form of AS_SET

seems that an MPLS label is a primary candidate to denote the path. As a result, SR is included to packet encapsulation solutions.

Since NIRA represents end-to-end route using source and destination addresses that are usually present in packet, neither packet modification nor encapsulation are needed. Nevertheless, in order to make communication possible, NIRA requires additional external signalling for discovering routes between end hosts, tracking changes and monitoring their availability.

Platypus combines authenticated source routing with the concept of *network capabilities*. It uses *policy compliance* in order to address a problem of routing policy constraints among operators. In Platypus path selection is based on an external mechanism.

Path Splicing is based on an idea that several different routing trees in a network topology can be combined to increase the number of possible paths between the source and destination nodes. Traffic may switch trees at any node on the way to the destination

Table 2
Key features comparison.

	Mechanism	BGP adaptation	Packet encapsulation	Packet modification	External signalling
Available mechanisms					
1	GMPLS		✓		✓
2	BGP Add-Paths	✓			
3	LISP		✓		✓
4	Segment Routing	✓	✓		
Moderate research interest					
5	NIRA				✓
6	Platypus			✓	
7	Path Splicing			✓	
8	STAMP	✓		✓	
9	DIMR	✓		✓	
Marginal research interest					
10	BANANAS	✓		✓	
11	WRAP			✓	
12	Routing deflections			✓	
13	MIRO	✓			✓
14	Pathlet Routing	✓			✓
15	YAMR	✓			✓
16	AMIR		✓		✓
17	BGP-XM	✓			

node, and this process may be controlled by end systems using the additional bits in the packet header. The authors have shown that the proposed routing primitive provides a near-optimal reliability (compared to that of the underlying physical network graph), both in intradomain and interdomain scenarios. Furthermore, the solution can be deployed in the existing network in such a way that avoids modifications of the original BGP message format, and which does not rely on additional routing messages.

In STAMP and DIMR, the BGP processes are modified. It is necessary to send some more date than in original messages. Additional signalling is not necessary.

BANANAS is a framework based on source routing in which the path is hashed and encoded in a so called *PathID*. Therefore, packet modification is required. Similarly to STAMP and DIMR, the adaptation of BGP is also required for the BANANAS framework to operate.

WRAP is based on the Loose Source and Record Route (LSRR) approach specifying end-to-end domain-level path (another mechanism implementing source routing). In WRAP each edge router computes at least two different AS-paths to each reachable AS. As in standard source routing approach, the router is stored within the packets, therefore, packet modification is necessary.

In Routing Deflections, the end user specifies which router along the path should perform the deflection. The indication is signalled by inserting or modifying existing information carried in the packet header.

MIRO and Pathlet Routing operate based on external signalling coupled with the adaptation of the BGP protocol. Changes in BGP are necessary to force finding alternative paths, whereas additional signalling is used to inform the network which paths are to be used for each transmission.

YAMR uses BGP messages to select paths. Although the whole process is new, the exchanged messages are not modified. Instead, new signalling protocol is necessary for implementation.

AMIR represents an alternative way of routing which leverages an extended cooperation between adjacent ASes on the Internet to collect more information about the AS-level network topology and construct multiple paths between the source and destination AS. Signalling is based on a custom packet header and external control channels. Alternative paths are determined based on information from AS which are located on the main path for a given relation, while the primary path is obtained from the local BGP route table. The capability of AMIR to introduce new paths through the negotiation process is considered as an important feature which is not present in other solutions like YAMR and Path Splicing.

The BGP-XM does not need any change in a format of exchanged BGP information. Only routers working with BGP-XM require new interpretation of the standard BGP information. There are defined new criteria for selection of routes which are placed in a routing table. A few BGP routes to the same destination can be used in the same time.

Part II Commercially available mechanisms

4. Generalized multiprotocol label switching (GMPLS)

GMPLS [7] is a complex and universal switching concept based on flexible usage of labels in different networking environments. GMPLS is a natural extension of MPLS [22] towards multiple networking solutions: TDM-, packet-, wavelength- and fiber- aware interfaces (environments).

GMPLS proposes a unified control plane for heterogeneous networking, offering topology discovery, resource provisioning, and connection establishment and release, supplemented by management functions. GMPLS is a unique solution; since it integrates simultaneously/offers multi-domain and multi-layer integration of switching capabilities, it scales horizontally and vertically [21]. As a connection-oriented solution with a plethora of functions, GMPLS is promoted by telecom operators and serves to control multi-domain and multi-service networks.

Three main protocol-components of GMPLS are:

- RSVP-TE (Resource Reservation Protocol with Traffic Engineering) signalling protocol [23],
- CR-LDP (Constraint-based Routing Label Distribution Protocol) [24,25].
- LMP (Link Management Protocol) [26].

and additional components are OSPF-TE (Open Shortest Path First with Traffic Engineering) routing protocol [27], and IS-IS (Intermediate System to Intermediate System) [28]. From the above list of main components of GMPLS the complexity and flexibility of this framework can be seen.

Using signalling in GMPLS it is possible to transfer and require [7]:

- Parameters for established label-switched path (LSP), such as required bandwidth, type of signal,
- And special features or functions, such as the desired protection and/or restoration method, as well as specific settings, e.g, the reserved position in a particular multiplex.

Generally, GMPLS is a very complex and also intentionally open framework. GMPLS was proposed, and its further development is stimulated, by telco operators, demanding a flexible, optimized and universal control plane for connection-oriented, multi-technology and multi-service solutions. Simultaneously, being open for recent advances, GMPLS incorporates major advancements done for IP networks, such as RSVP-TE, OSPF, TE, BGP, etc.

4.1. Path setup in GMPLS

Since the path in GMPLS is set up in response to a request sent by the user via user-network interface (UNI), it is the responsibility of the user to establish more than one path simultaneously. The traffic sent to paths can be differentiated. Another option is to establish a new path when a situation such as path overload or failure occurs.

The format of the path setup request depends on the type of LSP: for Time Division Multiplexing (TDM), Lambda-Switch Capable (LSC) or Fiber-Switch Capable (FSC) types, it is done by sending a PATH/Label Request message directed to the destination, with Generalized Label Request defining the path type (relevant technology layer), payload type and an Explicit Route Object (ERO), if available.

Thanks to extended traffic engineering mechanisms, and also protection and restoration features, GMPLS enables recovery from congestion and failure situations. What is more, GMPLS is a natural enhancement of the MPLS concept - the latter already solves multipath transmission for improving resiliency and performance issues.

GMPLS inherits two important features from MPLS: LSP modification (changing some LSP parameters without changing the route) and LSP re-routing (setting up a new path and then disconnecting the previous one, without interruption of traffic transmission). These features help to modify LSP's parameters and to enhance transmission performance even in unexpected situations.

From the above list of signalling options, it may be expected that GMPLS covers different requirements for path setup. Path setup is initiated by Forwarding Equivalence Class (FEC) assignment, thus the outcome of traffic classification is responsible for precise definition of route demand. There are different options for choosing a route for the path:

- Strict (all nodes for the route are specified) or loose (not all nodes are specified the route between subsequent hops are chosen using available routing protocol with available routing tables),
- Explicit (whole route is specified) or hop-by-hop (the route towards the destination is decided using updated information) routing,

Another useful dimension is the usage of the available switching hierarchy in GMPLS and, as a consequence, the hierarchy of interfaces, from Packet Switch Capable, through Layer-2-, then Time Division Multiplex-, then Lambda- and finally Fiber- Switch Capable, giving a high level of scalability and flexibility in traffic routing and grooming.

4.2. Possible extensions

GMPLS is designed as an open solution for the control plane, regardless of the network transport technology being used, and it can be flexibly used for new purposes and technologies. E.g., in the case of the necessity to implement a flexible control plane for a novel concept, such as OBS (Optical Burst Switching) the solution of choice can be precisely GMPLS.

5. Advertisement of multiple paths in BGP (add-Paths extension)

The base BGP standard allows selection and advertisement of only one (the best) path for any prefix in a single BGP session. Advertisement of an additional route for the same prefix results in replacement of the existing route entry with the new one. This mechanism is known as *implicit withdraw*.

This constraint renders multipath interdomain routing hard to achieve. The BGP standard, however, defines a mechanism of extensions, which can extend or modify the base protocol. Extensions used in each BGP session are subject of negotiation between peers during session establishment. A BGP extension *Add-Paths* [29] has been proposed in order to facilitate multipath interdomain routing. It allows advertisement of multiple paths for the same address prefix, without the new paths implicitly replacing any previous ones. Although this extension have not gone through the complete standardization process yet, it is already implemented in many networking operating systems and some open source routing daemons. These include operating systems of Cisco, Juniper and Alcatel-Lucent, as well as in some open source routing daemons, such as BIRD or Quagga. Analysis and comparison of these implementations was a subject of an IETF report draft [30].

The essence of the extension is that each path is identified by a path identifier in addition to the address prefix. Path identifier is a 32-bit long opaque value, which is advertised along with route. Combination of path identifier and address prefix is used to identify routes unambiguously. Path identifier is assigned locally by each peer. Its only purpose is to uniquely identify a path advertised to a neighbor. It should not carry any additional semantics.

The Add-Paths extension defines how additional paths should be advertised to peers. It does not, however, specify how these advertised paths should be selected from the set of available paths – this is left to the implementations. Several possible selection modes have been proposed and analyzed [31]. An IETF Internet Draft [32] provides recommendations to implementers how Add-Paths capability should be implemented and which of these modes should be available, depending on the target application.

There are many envisioned applications of the Add-Paths extension. First, it should allow fast connectivity restoration [33]. If a router has a backup path, it can directly select that path as best upon failure of the primary path, without the need of waiting for BGP protocol to re-converge. The next application is load balancing. When multiple paths are available, traffic can be directed on all these paths simultaneously. The Add-Paths extension can be also a valuable tool in helping to churn reduction and suppression of route oscillation [34].

The first drawback of Add-Paths is that, in order to make use of it, both peers must support it and have it turned on. This distinguish it from BGP-XM, which can be used also when peer does not support it. The second drawback of the Add-Paths extension is that all additional routes needs to be stored in the routing table. As the routing table size is already a problem for many operators, this drawback can significantly reduce deployment of this extension.

6. Locator/ID separation protocol (LISP)

LISP [11] represents a group of solutions enabling a separation of two typically combined functions of IP address: localization - namely the topological information of host location - and the identification of the host. The main motivations of LISP encompassed

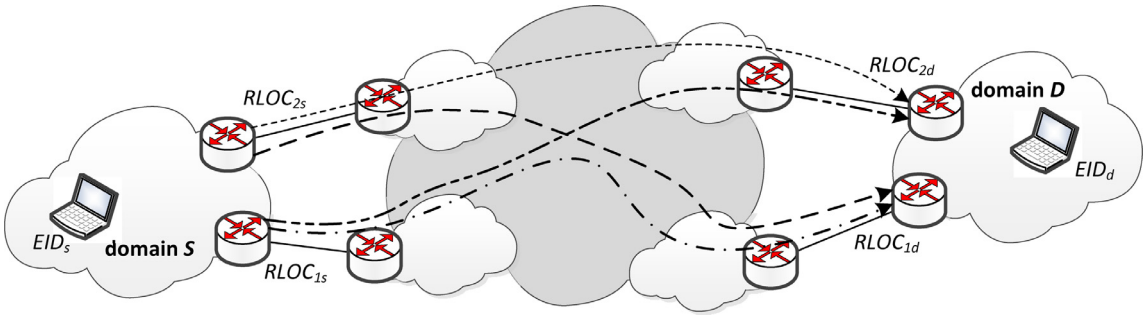


Fig. 2. Possible paths between stub ASes using LISP.

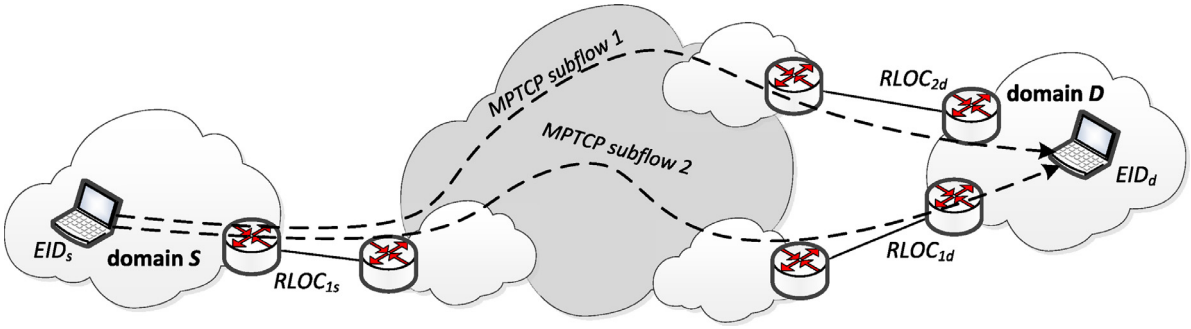


Fig. 3. Example of concurrent multipath transferring using MPTCP and LISP.

addressing scalability problems in Internet routing architecture, decreasing the number of prefixes announced in the Internet, decreasing size of BGP routing tables in the core, and, minimizing the number of routing updates. Although a support for multipath was not among the design goals of LISP its features enables realization of multipath transmissions.

LISP disassociates the locator and identity functions of an IP address as follows. Hosts located in a stub network obtain Endpoint Identifiers (EIDs). EID is a normal IP address. The host EID can be obtained by a normal DNS query. EIDs are has a local meaning and are used to route packets within stub network (e.g. using interior routing protocol). EIDs are not used for routing packets through core networks. Instead, Routing Locators (RLOCs), assigned to stub network border routers are used. The EID address is mapped to one or more RLOCs. Remote end hosts use EIDs as source and destination addresses for packets. When a packet reaches a LISP router the mapping service lookups for a destination RLOC for a given destination EID. Then the packet is encapsulated and routed through the network using IP addresses representing source and destination RLOCs, i.e., addresses of border routers of domains where two communicating end hosts are located. Packets are routed in the core network using a path determined by the underlying routing protocol, i.e., BGP. In fact, the path found is the shortest path found using some BGP metrics, and LISP had no influence on the path selection. Then, a packet is decapsulated at the border router of a destination stub network, and afterwards sent to the destination host using its EID.

The LISP mapping system uses a dedicated signalling protocol and a global distributed database that contains EID-Prefix-to-RLOC mappings. In the case of a multi-homed network, an EID prefix might be mapped to more than one RLOC. Each RLOC is assigned a *priority* and *weight*. Typically, the *priority* value indicates which RLOC is most prefered while the *weight* parameter indicates how to load-balance the traffic between RLOCs of equal *priority*. Assuming that a domain in which flows originate is aware of multiple locators of a destination EID, it may select different locators for different source-destination EID pairs. Traffic may also be balanced by sending flows destined at the same EID by using different locators for different flows.

Using LISP, multiple paths between two remote domains are possible if at least one of them is multi-homed. Fig. 2 presents a set of interdomain paths that may potentially be used between two remote domains: domain *S* serving as a traffic source and domain *D* in which host receiving the traffic is located. EIDs in *D* might be mapped to a single or two locators simultaneously, $RLOC_{1d}$ and $RLOC_{2d}$. The Internet Service Provider (ISP) may elastically divide a pool of identifiers into groups mapped to different locators. This mapping might be changed if needed. Such an approach is much more flexible than defining strict assignment of address prefixes to border router interfaces. Assuming that a given identifier (e.g., EID_d) is mapped to both RLOCs, and this mapping is announced via a LISP mapping system, the sender side may see and use two paths to the destination simultaneously. Domain *S* is also assumed to be multi-homed. Using internal routing policy, the ISP operating domain *S* may decide through which border router the traffic from EID_s and targeted at EID_d is sent. Therefore, considering a given source-destination pair (EID_s, EID_d) four different interdomain paths may be used.

Multiple paths between two communicating stub LISP enabled autonomous systems may exist on the condition that at least one of them is multi-homed. The path through the core network for a given source-destination RLOCs pair is selected by BGP independently from LISP. Therefore, it is not possible to predict if those paths are fully disjoint. In practice, only the first and the last hop of the path can be chosen.

One example of practical exploitation of the potential of LISP to create multiple paths between two end-host is a combination with MPTCP protocol [35–37]. The idea is presented in Fig. 3. Using MPTCP the flow from EID_s to EID_d is split into two subflows (either hosts should support MPTCP or an MPTCP proxy [38,39] must be used). There are two possible approaches to realize such communication (here we assume that both hosts supports MPTCP):

1. The destination end host has two identifiers assigned, e.g., EID_d^1 and EID_d^2, and both are known to the source host. Those EIDs are mapped to different RLOCs, i.e., to $RLOC_{1d}$ and $RLOC_{2d}$. The source host opens two subflows with the following source-destination address pairs: (EID_s, EID_d^1) and (EID_s, EID_d^2). The border router of AS_s obtains mappings to both destination addresses and encapsulates packets of those subflows into two different tunnels: ($RLOC_{1s}$, $RLOC_{1d}$, ($RLOC_{1s}$, $RLOC_{2d}$), respectively.
2. The destination end host has a single identifier, EID_d. It is mapped to two RLOCs with equal *priority* and *weight*. Therefore, $RLOC_{1s}$ may split the traffic targeted at EID_d among two LISP tunnels. However, the source is not aware of such a possibility; it knows only a single address of the destination. Thus, to make it possible to split MPTCP subflows among two inter-domain paths, a communication between MPTCP layer and LISP is need. This approach was taken in [35–37].

Other solutions based on separation of identity and locator are HIP [40], Shim6 [41] and ILNP [42]. Their common feature is that they are all host-based solutions (in contrast to LISP which is network based). The separation of two identity and locator functions is done at the end-host. The goal of those solutions was to solve problems with maintaining the connectivity of applications running on a mobile device when the device roams between IP domains. The host receives a constant identifier that is presented to upper layer protocols and used by the application. The identifier is mapped to a locator that is used for routing and may change depending on which IP domain the host is currently visiting.

Those solutions can be potentially used for multipath transmissions. Let's consider mobile device, such as a smartphone, has two types of network access active at the same time: wifi and LTE. Let us assume that both connections are provided by different ISPs. The host identifier may be mapped to two different locators: one in the wifi network, the other in the LTE network. Then the host may communicate with the other remote host using two different paths. For instance, an MPTCP enabled mobile device may coomunicate with remote host (also MPTCP enabled or via remote MPTCP proxy). One of the practical proposals towards the exploitation of the multipath potential of host- based identity/locator separation solutions is mHIP (multipath HIP [43]). For shim6 based multipath solution see [44]. Finally, such solutions as HAIR [45] and GLI-Split [46] might be considered hybrid or multi-level solutions.

7. Segment routing (SR)

The Segment Routing (SR) is a relatively new concept in networking proposed by Source Packet Routing in the Networking (SPRING) IETF working group. The main purpose of the SR is to enable selection and usage of a non-shortest path for a packet. In a network with the SR a source routing paradigm is reused, but with the assumption that the term 'source' means a node where the source routing is used. As a result, a source routing may be used by a node different than the node where a packet was originated. Contrary to the original source routing concept, the SR does not have to be implemented in all nodes on a path from given source to destination. A network operator may use the SR independently to other operators. This feature ease the SR implementation. The aim of source routing usage is to alleviate network virtualization including multi-topology routing as well as to enhance implementation of load balancing and traffic engineering. Both strict and loose source routes can be utilized in the SR network. The term 'segment' is defined as an instruction which is executed in a node on an incoming packet. An 'instruction' in this case means packet forwarding through a selected interface, or routing of the packet according to the shortest path, or to deliver the packet to a given application or service instance. Therefore, the segment may be identified with an instruction – for example: convey a packet to node X or process a packet by module Y. Each segment has an identifier (Segment Identifier – SID) and these identifiers are stored in a segment list. The Segment Routing can be directly applied to Multiprotocol Label Switching (MPLS) architecture. In such a network a segment is encoded as a label, whereas a list of segments may be seen as a the stack of labels. The SR may be also applied for the IPv6 network. Here, a segment is encoded as an IPv6 extension header. A stack of labels or an extension IPv6 header defines the path through which a packet should be transported. The SR reuses the MPLS dataplane without any changes. Moreover, there is no need for an extra dedicated signalling protocol. However, the IGP based segments require some extensions to the utilised link-state routing protocols. The enhancements for these protocols are, so far, available in draft versions, e.g.[47,48]. For the interdomain routing protocols some Internet drafts are also published [49]. The SR allows the carrying of data without the usage of cumbersome label signalling protocols like Label Distribution Protocol (LDP) or Resource Reservation Protocol (RSVP).

So far, the main focus is the implementation of the SR in the MPLS network. The following types of segments for Segment Routing MPLS network are defined:

- An IGP segment,
- An LDP LSP segment,
- An RSVP-TE LSP segment,
- A BGP LSP segment.
- A BGP Peering segment,

The IGP segment may denote an IGP-prefix, an IGP-Node, an IGP-Anycast or an IGP-Adjacency segments. The segments identify a path, a node, a set of routers or an interface, respectively. The LDP LSP, RSVP-TE and BGP LSP are segments which allow to use a path which is different than the shortest one and, in part, selected and denoted by LDP, RSVP-TE or BGP.

From the interdomain multipoint transmission point of view the most interesting is the BGP Peering segment. If a flow of packets has to be transported from one autonomous system to another, then a list of segments may be specified. The list is used to forward a flow of data within the AS towards an egress node. In such a case, the BGP Peering Engineering policy may be used to attach two segments: the Node Segment ID of the egress node, and the BGP Peering Segment for the selected egress node or peering interface. The following BGP peering segments in [50] are specified:

- PeerNode SID,
- PeerAdj SID,
- PeerSet SID.

The first segment identifies the node, the second the interface of a node, and the last one the group of nodes. From a load balancing point of view, the most important segment is called PeerSet SID and should allow load balancing among a set of connected peers. For example, nodes G and H may be selected as a PeerSet (see Fig. 4). Selection of a group and the exact definition of load balancing, i.e., routing policy, is specified by a network operator. The defined PeerSet ID and BGP peering may enhance utilization of available resources between autonomous systems, since more than one path may be concurrently used for data transmission. Moreover, if there is a failure on a connection within a given PeerSet, then the other connection from the set may be used instantly. As a result, exchange of information between autonomous systems can be smoothly restored. It is worth noting that the SR policy employed in the AS 1 shown in Fig. 4 may steer some packets through a path other than the best selected by the BGP protocol. For example, a packet may be sent from user U1 via egress point A rather than B. Alternatively, the AS 3 may be preferred via node H rather

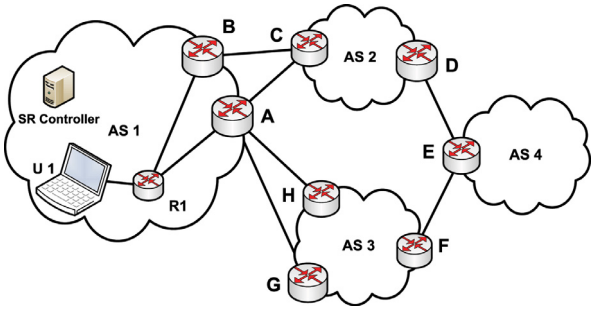


Fig. 4. An example of the network with the SR implemented.

than through node G, or the highest priority may be set for the path through any node of the PeerSet. The path selection process in the SR network may be performed in a distributed or centralized controller. For example, in Fig. 4, a centralized controller is responsible for path selection. The SR is in the early stages of the standardization process. There are no valid standards, merely internet drafts published by the IETF working group. However, necessary enhancements in draft versions to the OSPF, IS-IS and MPLS protocols are proposed. Similarly, drafts for BGP are published as well [49,51]. It is worth to note that the standardization process seems to be quite vivid, hence, som standards may be available in the near future. Some research papers are also available [52–54].

Part III Mechanisms of moderate research interest

8. A New Internet routing architecture (NIRA)

The main goal of NIRA is to provide end users with the possibility of choosing the sequence of Internet service providers (domain-level routes) a packet traverses. The authors believe that their solution will foster competition between ISPs, and users will gain from the improvement of end-to-end performance and reliability as well as new, enhanced services that will be introduced. However, they are also aware that it can lead to route oscillation or suboptimal route selection. To make the design more tractable, NIRA supports user choice only at domain-level rather than router-level.

The NIRA architecture was first introduced in [19]. Yang presented it as a viable technical solution covering a broad range of issues: (i) deployment feasibility, (ii) efficient route representation, (iii) route discovery, (iv) failure handling, (v) provider compensation. In [20], Yang et al. evaluated the design of NIRA using a combination of simulation, network measurements and analysis.

8.1. Design overview

In the NIRA design AS domains are divided into two regions: *Core* and *Access*. The *Core* consists of Tier-1 providers which do not purchase transit service from other ISPs. The *Access* region is a chain of providers between users and *Core* (called the user's up-graph). The customer-provider business relationship is most common in this region, but peer-to-peer can also be present. The domain-level route is constructed with two up-graphs of sender and receiver and it is said to be *valley-free*. An example route from U1 to U2 is presented in Fig. 5 with bold arrows.

8.2. Route representation scheme

NIRA splits an end-to-end route into two parts: (i) a sender part, and (ii) a receiver part. Both parts are represented using addresses; this means that to send and forward packets, routers have

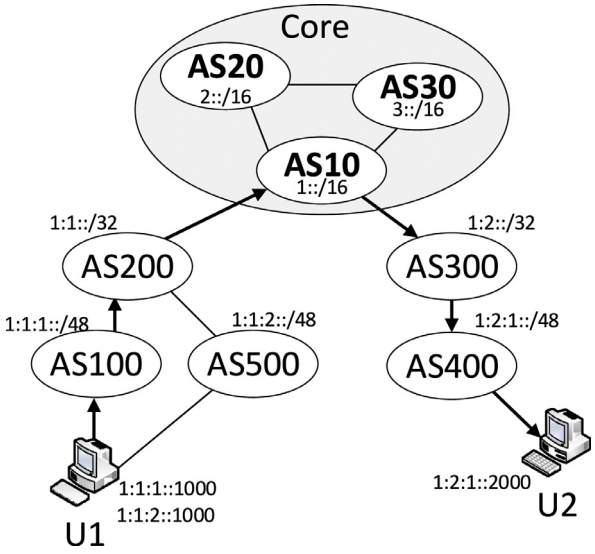


Fig. 5. Example of the provider-rooted hierarchical addressing in NIRA.

to check not only the destination address, but the source address as well. It is worth noting that, contrary to source routing, NIRA supports user choice without expanding packet headers.

The authors decided to use a provider-rooted addressing scheme based on IPv6 to encode a route that connects the user to the *Core*. In Fig. 5, an example of address assignment is shown. AS10 in *Core* has a globally unique address prefix — 1::/16 – and allocates prefixes 1:1::/32 and 1:2::/32 to its customers AS200 and AS300 respectively. ASes continue to assign prefixes to their customers until end users are reached. As a result, route AS100-AS200-AS10-AS300-AS400 between U1 and U2 is uniquely represented with two addresses: 1:1:1::1000 and 1:2:1::2000. This is a basic example, i.e., without peer-to-peer relationships in the *Access* region, but NIRA architecture also covers such cases. Interested readers can find more details in [19] and [20].

8.3. Route discovery

In order to bootstrap communication, the sender needs to discover his and the destination's up-graphs. Then the user can use a source and destination address to encode an end-to-end route and change routes by changing addresses.

NIRA provides two mechanisms for route discovery: (i) The Topology Information Propagation Protocol (TIPP) and (ii) The Name-to-Route Resolution Service (NRRS) to discover sender and receiver up-graphs respectively. With the help of TIPP, providers propagate to a user his addresses and the routes associated with these addresses. Moreover, TIPP informs users if domain-level topology changes occur. NRRS maps the name of a destination to the route segment the destination is using. When a user wants to be reached by others, he has to register his route segments corresponding to addresses obtained from TIPP in NRRS. The user is also responsible for updating the entries in NSSR upon reception of topology change information from TIPP.

After solving the bootstrap problem and successful transmission of the first packet, two users can exchange all possible routes they have and agree to use one they both like.

8.4. Failure handling

To use discovered routes successfully, a user has to know whether they are failure free. The mechanism that provides failure discovery consists of a combination of proactive and reactive feed-

back. A user is immediately notified about any changes in his up-graph. Thus, during the communication initiation, the user knows which of his routes are available.

In order to reduce the number of TIPP messages users receive, they are not propagated globally (i.e. the user receives only messages related to his providers' domains). As a consequence of this rule, a user does not know the availability of routes on a destination's up-graph. Thus, to discover route failure, a sender node has to rely on reactive mechanisms such as: (i) a router feedback — a router in a network has to notify the sender when it notices that the route specified in the packet header is unavailable; (ii) a time-out — in cases when the router is overloaded or the route between router and sender is broken, a sender uses a time-out mechanism to detect route failure. The former solution provides fast route fail-over and allows the user to switch to a new route in a period on the order of a round trip time. In the latter solution, switching time depends on the time-out value.

As the reactive notifications can increase the time of connection initialization, it is advised that users should cache states of recently used routes and use only ones that are available. In addition, users or ISPs can employ any mechanism to discover route availability, such as monitoring routes by sending a probe.

8.5. Provider compensation

No technical design will be implemented if there is no practical payment scheme for service, and if providers cannot benefit from giving the user the possibility to choose routes, they would not allow for it. Being aware of that fact, the authors also addressed this issue.

It is not feasible to sign a contract with every ISP in the world, so to make a payment scheme practical, NIRA constrains users to choose a route only from a set of providers they agree to pay for by signing bilateral contracts.

Yang proposed two compensation schemes: (i) Direct Business Relationships — directly connected ISPs sign the agreement, and monitor and charge the customer differently based on the routes he/she uses. Some mechanism of policy checking is required to prevent usage of illegitimate route fragments; (ii) Indirect Business Relationships — users are able to sign a contract with non-directly connected providers. However, in this case, packets coming from one adjacent domain may come under various transit policies, and preventing route misuse becomes more complicated.

9. Platypus

Platypus [55,56], is a relatively new approach to enable source routing. The proposed mechanism combines authenticated source routing with the concept of network capabilities. Such a solution can be deployed at both the edge and the core of wide-area networks.

9.1. Design goal

Platypus, like other source-based routing proposals, is able to select among multiple paths in order to ensure efficient, flexible and reliable packet forwarding. However, it addresses a problem of widely used routing policy constraints among operators which causes part of the network to be hidden from others. The key challenge in the use of source routing is the motivation of transit ISPs for sharing their resources and allowing them to transmit arbitrary traffic through filtered links. That is why the authors of Platypus propose a *policy compliance*. Clear business relations with appropriate billing mechanisms are fundamental incentives for the use of source routing. Due to policy compliance, an intermediary ISP can authorize a transfer of traffic belonging to others and bill the appropriate party.

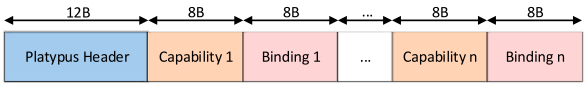


Fig. 6. Platypus header with series of capabilities and its bindings.

9.2. Mechanism operations

The authors assume that many paths exist between the end points and that the best path (different from the default) is selected based on an external optimization mechanism (not addressed in the article). Such a path is a set of hops (*waypoints*) that are used during traffic forwarding. The specification of each router on the path is not required. Waypoints have to have routable IP addresses. The first Platypus router on the path replaces the destination IP address to the first waypoint address and the source address to its own IP address. The router then inserts an additional header immediately after the IP header with, among others, original source and destination IP addresses. After this, the Platypus header list of so called *capabilities* is inserted (see Fig. 6). This list includes the set of waypoints. When the Platypus packet arrives at the second Platypus router (second waypoint), it replaces the destination IP address with the next from the capability list, and replaces the source address with its own IP address. Additionally, the pointer to the next waypoint inside the Platypus header is changed. This process is continued until the packet reaches the last router on the path. The final Platypus-enabled router replaces both IP addresses, source and destination with the original in accordance with the Platypus header, and removes the additional header. As can be observed, it is not required that all routers on the path support the discussed mechanism. It is worth mentioning that replacing the source IP address allows for the preservation of local routing policies.

As well as the original source and final destination addresses, the Platypus header also comprises the version and flags field, capability list length, capability list pointer, and encapsulated protocol field (taken from original IP header and replaced with a Platypus specified number, used to facilitate de-encapsulation). The protocol field from the IP header is used to recognize Platypus packets. Also, the particular capability is more complicated. As well as the waypoint, it contains a *resource principal* — an entity willing to be charged – and *binding* — a stamp used for authorization. Since it is easy to eavesdrop Platypus packets and forward attacker traffic through a given waypoint, meaning that the indicated resource principal will be billed, bindings are used. When a Platypus packet arrives at a given waypoint, it validates it using capability calculated as a function of a packet content known only to the capability owner.

9.3. Discussion

The proposed mechanism has some disadvantages. Firstly, it limits available payload size in a packet due to inserting an additional header immediately after the IP header. After the Platypus header, a list of capabilities and its bindings is inserted. This list strongly depends on the number of hops between end users. Moreover, the authors do not propose any mechanism for wide area route discovery. It can be hard to determine all existing paths between given parties due to, e.g., prefix filtering and policy constraints. Additionally, charging of other ISPs is not an easy process; clear definition of business relationships is required. Finally, selection of the best path from the path set is not addressed.

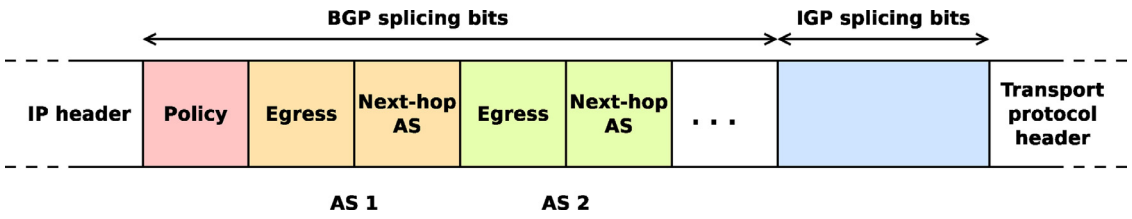


Fig. 7. Splicing bits embedded in the packet header.

10. Path splicing

Path Splicing is a routing technique introduced in [57]. It is based on an idea that several different routing trees (called *splices*) in a network topology can be combined to increase the number of possible paths between the source and destination nodes. Traffic may switch trees at any node on the way to the destination node, and this process may be controlled by end systems using the additional bits in the packet header. The authors have shown that the proposed routing primitive provides a near-optimal reliability (compared to that of the underlying physical network graph), both in intradomain and interdomain scenarios. In this section, we briefly discuss the applicability of the proposed technique to the interdomain routing case.

10.1. Interdomain path splicing

The authors propose a way to deploy path splicing in the existing network that avoids modifications of the original BGP message format and which does not rely on additional routing messages. Instead, BGP routers are reprogrammed to exchange the best k routes corresponding to each destination, considering the fact that backbone routers usually maintain several sessions with neighbours, and consequently learn multiple alternative routes. Then, at each hop on a path, one of k possible routes is selected based on the *splicing bits* which are embedded in the header of each packet forwarded through the network. This approach contrasts with MIRO (Section 16), as it does not rely on the control plane with regard to the discovery of alternative routes. Fig. 7 shows the general splicing bits embedding scheme, as proposed in [57].

The additional segment in the packet header contains the following two main parts:

- **BGP splicing bits** — for each Autonomous System (AS), the ingress and egress nodes are considered separately, and both nodes make their routing decisions individually based on the corresponding sections with splicing bits (for instance, in Fig. 7, the *Next-hop AS* field for AS 2 is analysed by the ingress node of this AS, whereas the *Egress* field — by the egress node of the same AS); in addition, there is an additional *Policy* field which determines whether the packet should be forwarded through a peer or customer AS;
- **IGP splicing bits** — this field may be reused while traversing different Autonomous Systems along a path, and its content determines the internal route in the related AS.

The authors have identified the following issues which might arise in the network after the deployment of the proposed routing primitive:

- **interdomain forwarding loops** — as traffic flows may be switched between different network trees, a mechanism is needed that will prevent forwarding loops on an interdomain level;
- **AS-level forwarding consistency** — the actual forwarding path between Autonomous Systems may differ from the advertised AS path.

At the same time, the authors have shown that both problems can be solved without having to introduce substantial modifications to network routers. Furthermore, they have demonstrated that the proposed technique offers significant advantages in terms of reliability and deployment cost. An additional evaluation covering the selected performance- and security-related aspects of path splicing in the case of two real network topologies is presented in [58].

11. Selective announcement multi-Process routing protocol (STAMP)

STAMP was first proposed in [59]. The main assumption of this proposal is to run two BGP processes which are able to compute two complementary routes. Each process is a slightly modified BGP process, in which the path is selected as in the standard BGP. Only two new path attributes are added to the BGP messages. One of them is responsible for coordination of the process in an AS, and the second one determines which path should be used. In case of any failure or other instability in the network, working paths to destination nodes remain available. After an unpredicted event in the network, it may take as much as 30 minutes until BGP converges. During this time period, transient loops may occur.

The STAMP protocol improves reliability of interdomain routing. When one path fails, another one (computed based on another process of BGP) can be used immediately. The complementary paths are computed in such a way that they are not affected by the same set of routing events in the AS.

In STAMP, one of the parallel BGP processes is designated *red* and the other *blue*. Disjointness between red and blue paths means that the same AS nodes are not in both paths (except source and destination). This ensures that paths are not affected by the same set of events. The functionality of the STAMP can be realized by using distinct TCP ports or by handling two routing instances to differentiate processes. To compute red and blue paths, it is assumed that selective routing announcements are propagated to a constrained number of ASes. This means that blue announcements are propagated along one set of providers and red ones are propagated along a disjoint set of providers. The method for selecting providers for blue and red processes may be chosen in many ways. The authors of [59] propose the use of priorities. For example, when we have multi-homed AS (AS with many providers), such an AS may choose one provider (blue) to which it announces all prefixes through its blue process only. Other providers are announced through the red process. This ensures that each AS can be reached through both paths (red or blue) associated with different last hop providers. The authors of [59] also propose the use of a BGP path attribute *Lock* to sign blue announcements. When a provider receives an announcement with *Lock* set to 1, it is obliged to propagate it further to the next provider with the same attribute and to propagate other announcements (red) to other providers with *Lock* set to 0. This ensures that paths of both colours to all prefixes are always established.

Each STAMP routing process is safe as long as BGP is safe. The experimental evaluation described in [59] confirms that STAMP

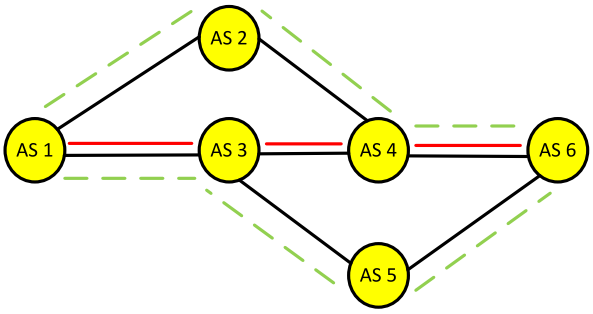


Fig. 8. Example of paths between two ASes.

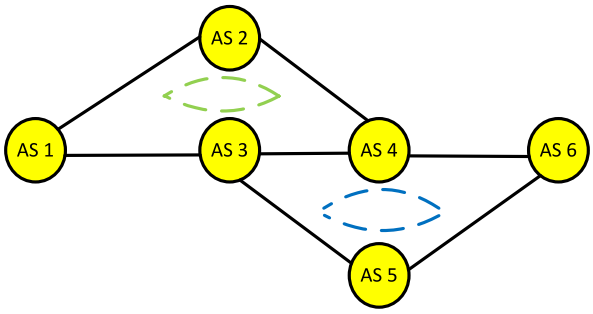


Fig. 9. Example of DIMR operation.

ensures greater routing stability compared to BGP. STAMP is less complex than some other modifications of BGP, e.g., when route cause information is used in routing updates, and guarantees similar improvements for failure scenarios.

12. Disjoint interdomain multipath routing (DIMR)

The DIMR algorithm for multipath interdomain routing was proposed in [60]. The aim of this proposal is to find two disjoint paths between two ASes. The main assumption is that both paths are selected simultaneously. For example, if we look at Fig. 8, we can see that if the primary path is selected first (red in the figure) it is impossible to find the second disjoint path between nodes AS1 and AS6. However, if both paths are selected at the same time it is possible to choose disjoint paths (green in the figure).

DIMR can be considered as an advanced version of the Path Diversity Aware interdomain Routing (PDAR) algorithm. PDAR is a mechanism which allows for multipath inter-domain routing which prevents packet losses in case of failures. The authors of [61] propose to use two routing protocols to implement PDAR:

- Path Diversity-aware interdomain Routing (D-BGP)
- Bloom Filter-based D-BGP (B-BGP)

D-BGP allows a router to advertise the most disjoint alternative path and the best path. B-BGP, by implementing Bloom Filters, makes it possible to obtain a more scalable and practical solution. Bloom Filters compress the path lists originally used in D-BGP. As a result, the best paths and the alternative paths are compressed in an array of m bits which allows to limit memory usage.

The main assumption of PDAR is the possibility of advertising the most disjoint path and the best path to the same destination. This improves network reliability in case of failure. Moreover, information about failure is propagated along with PDAR messages and informs about failure location. When a link fails, the routers connected to that link send routing update information to all routers along each impacted path. This mechanism is called Root Cause Notification (RCN) and ensures that all invalid paths are removed quickly and traffic is rerouted to the alternative paths.

The simulation results presented in [61] confirm that the convergence delay of the D-BGP and B-BGP is reduced by 60% in comparison to BGP. Moreover, while more routing updates are generated due to the alternative paths, in case of failure the signalling is reduced.

In DIMR, paths are selected at the same time and, unlike in PDAR, have to be completely disjoint. The operation of the algorithm is based on defining circles composed of ASes in the network. Other Ases are informed about existing circles, and as a result can select disjoint paths to other ASes. An example of DIMR operation, which assumes to find disjoint paths between nodes AS1 and AS6, is presented in Fig. 9.

First, AS1 announces (1) to ASes AS2 and AS3. ASes 2 and 3 know paths to AS1 but cannot find two disjoint paths to this AS. Therefore they announce known paths to AS4 (from AS2 and AS3) and to AS5 (from AS3). Now, AS4 is able to set up two disjoint paths to AS1 and is aware of the existence of the green circle. This AS is responsible for informing ASes 1, 2 and 3 about the green circle. Next, AS4 informs AS6 about paths (4, 2, 1) and (4, 3, 1), and AS5 informs AS6 about path (5, 3, 1). As a result, AS6 becomes aware of the blue circle and informs ASes 3, 4, and 5 about it. Finally, AS6 is able to set up two disjoint paths to AS1: (6, 4, 2, 1) and (6, 5, 3, 1). Moreover, all ASes in the network are able to set up two separate paths to any other AS in the network.

12.1. Implementation requirements

To deploy DIMR in a network, the involved ASes must use the BGP protocol to exchange information about the prefixes and AS-level paths. Further, DIMR introduces the following requirements:

- All related forwarding tables must contain two additional fields: *Index* and *Next Index*; these fields are used to distinguish alternative paths from each other and to indicate the preferred AS path at the Next Hop AS;
- The packet header must contain the *Index* field, which is used by the Next Hop AS to select the proper path;
- The BGP update message must contain the *Index* field to distinguish alternative paths from each other;
- The involved routers must support the DIMR protocol.

Policy in DIMR is implemented with the aid of export filters.

The simulation results presented in [60] show that DIMR outperforms PDAR when considering convergence delay. Moreover, the average length of paths is lower in DIMR and far more disjoint paths were set up. The difference between both algorithms is small when we compare the number of control messages sent when routing changes. This means that the overhead of DIMR is comparable to PDAR. The proposed mechanism remains relatively simple, but at the same time it introduces nonstandard modifications to the operation of BGP (together with the forwarding subsystem) and requires that the firmware of the existing routers must be modified to support DIMR.

Part IV Mechanisms of marginal research interest

13. BANANAS

BANANAS is a source routing framework applying a loose path that is encoded, hashed and stored in so called *PathID* [62]. The PathID value is computed as a short hash of a sequence of globally known identifiers that can be used to define an end-to-end path, e.g., router IDs, link interface IDs, AS numbers. More information on how such hashes are computed can be found in [62].

BANANAS does not introduce any new scheme for path computation. As the authors propose in [62] new paths (different from

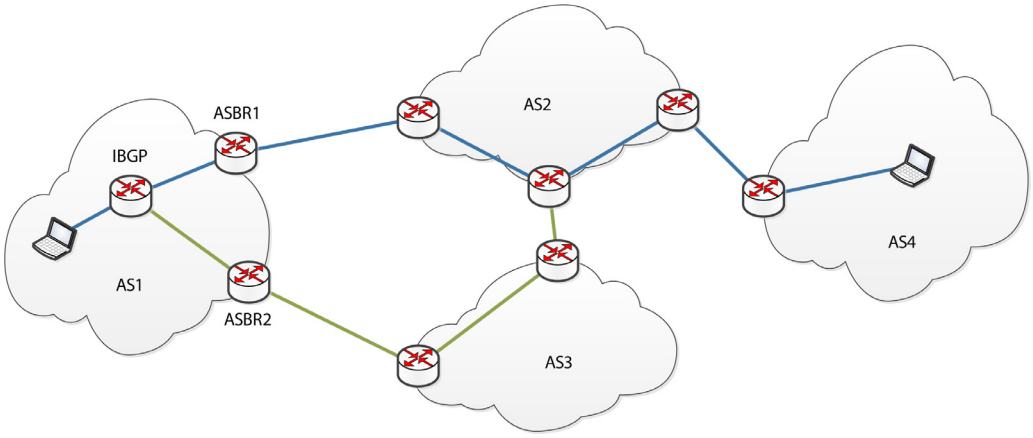


Fig. 10. Explicit-Exit Forwarding example.

those calculated by commonly used routing protocols) can be calculated using well known multipath computation algorithms, e.g., k-shortest paths, all k-hop paths, k-disjoint paths, etc. Such an approach is, rather, a new forwarding scheme that can be applied to an existing router's forwarding engine.

13.1. Packet forwarding

In comparison to a regular packet forwarding table in a router where a two-tuple [*destination prefix, outgoing interface*] forwarding table is used, BANANAS extends it to a four-tuple registry in form [*destination prefix, incoming PathID, outgoing interface, outgoing PathID*]. Two different hashes — *incoming PathID* and *outgoing PathID* are calculated as a hash of the explicit path from current interface to destination prefix and as a hash of the path from the next *upgraded* (BANANAS-enabled) router, respectively.

In BANANAS an *upgraded router* firstly matches the destination IP address following the longest prefix match, as in a regular router. Next, it matches PathID for that destination and, if found, the incoming PathID is replaced with the outgoing PathID and the packet is sent to the outgoing interface. If a match is not found, the hash is set to zero and the packet is sent in accordance with the routing protocol being used. This situation also occurs when the currently forwarding router is the final router on the path. Routers which are not aware of the BANANAS framework use regular procedures in order to forward packets.

BANANAS, in its basic functionality, is dedicated to be used as a mechanism enabling a multipath approach in an intradomain forwarding scheme, but it can easily be extended to an interdomain traffic forwarding scheme. Only a fine-grained approach can be applied to extend existing interdomain routing.

13.2. Explicit-Exit forwarding

The explicit-exit forwarding idea can be applied in cases where a given autonomous system has at least two different interdomain links (is multi-homed). The objective of such a solution is to use different interdomain links for a specified traffic aggregate (e.g., per-packet, per-flow, per-prefix) which in consequence may lead to reaching the destination with a different (selected by BGP) path.

The explicit-exit routing uses modified Internal BGP (IBGP) and External BGP (EBGP) routers. It works as follows: the IBGP router selects an arbitrary ASBR (Autonomous System Border Router) for a given traffic aggregate based on applied multipath computation algorithm, e.g. k-shortest paths, all k-hop paths, k-disjoint paths. Then it replaces the destination IP address in packets with a selected ASBR address, saving the original destination IP in additional 32-bit field called *address stack* (*options* field of IP header) and recalculating the packet's checksum. When such a packet reaches the exit ASBR, the exit ASBR replaces the destination IP with the original and recalculates the IP header checksum. Then the packet is forwarded in the regular way. In the case of the network presented in Fig. 10, the default BGP path from AS1 to AS4 is coloured with a blue line [AS1, AS2, AS4] and the default ASBR to AS4 is ASBR1, but when the IBGP router selects ASBR2 as the exit router for packets to AS4 the AS path will be changed to [AS1, AS3, AS2, AS4].

The proposed solution requires that the IBGP router has two routing tables: [Dest-Prefix Exit-ASBR Next-Hop-to-Exit-ASBR] and [Dest-Prefix Default-Next-Hop]. The latter is a regular IBGP table, but the former is added for the proposed mechanism. The explicit-forwarding mechanism requires that only some of the routers belonging to a given AS have to be upgraded: deciding IBGP routers and ASBRs.

13.3. Explicit AS-Path forwarding

The explicit AS-path forwarding proposal uses a distributed mechanism to send packets along an arbitrary selected and validated AS-path. It extends basic BANANAS functionality with a hash of external-Path ID (e-PathID) being a sequence of ASes.

To illustrate how the mechanism works, Fig. 11 will be used. Let us assume that traffic is sent from AS1 to AS5. Available paths are [AS1, AS2, AS5], [AS1, AS2, AS4, AS3, AS5], [AS1, AS2, AS3, AS5]. Let's assume that the arbitrary selected path by router in AS1 is [AS1, AS2,AS3, AS5], so the suffix AS-path placed in the e-PathID field is [AS2,AS3, AS5]. In such a case the next-hop is the ASBR21, since the packet outgoing AS1 exactly matches the prefix and e-PathID of AS2. Next, ASBR21 forwards the packet to ASBR22, and ASBR22 swaps e-PathID filed to [AS3, AS5]. A similar process takes place in AS3 — only exit ASBR (ASBR32) swaps e-PathID field. But in consequence of forwarding the outgoing e-PathID from AS3 will be set to 0 because AS5 is the destination AS for forwarded packet.

14. Wide-Area relay addressing protocol (WRAP)

WRAP is an approach based on source routing [63]. WRAP is based on the Loose Source and Record Route (LSRR) [64] but it uses a proprietary header instead of the implementation of the variable length IP options field.

WRAP allows a source node to specify the end-to-end domain-level path. It assumes that each system (edge domain) computes at least two different domain-level paths to each reachable domain in

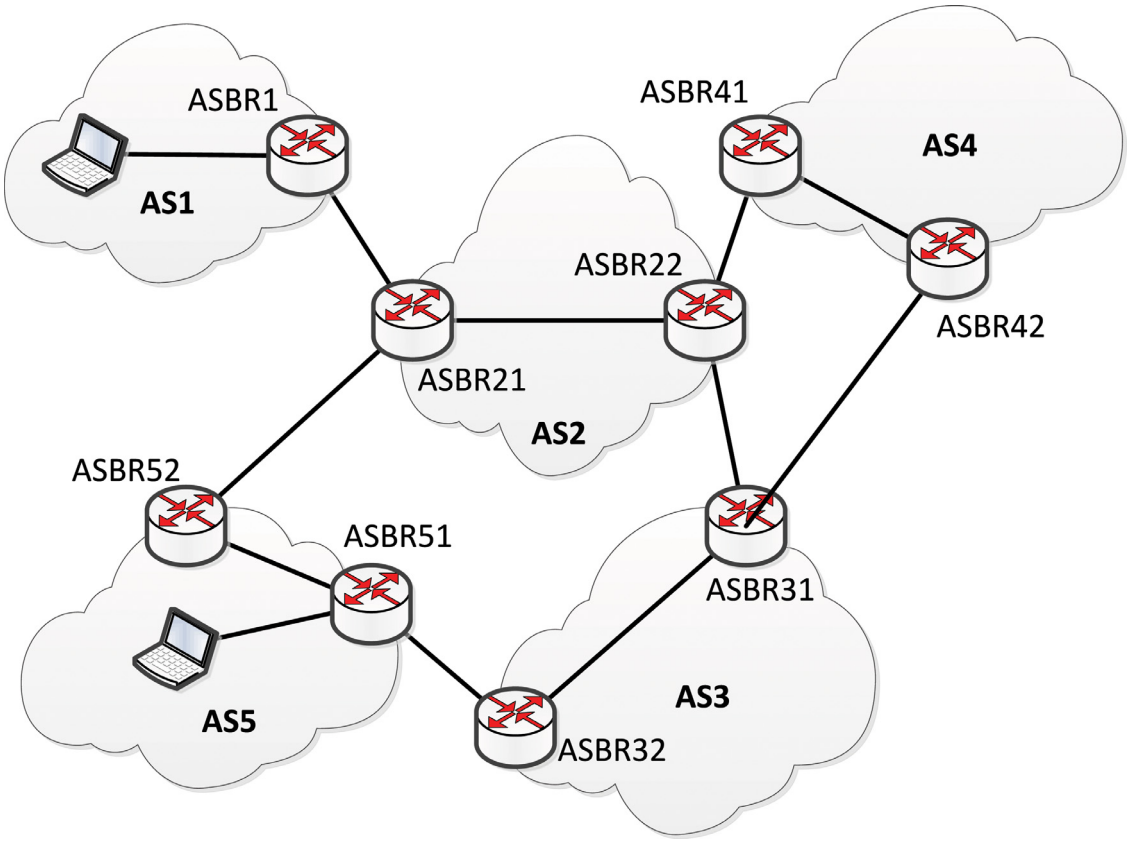


Fig. 11. Explicit AS-Path Forwarding example.

the network, and allows an edge router to choose between them. WRAP also enables an alternative approach — each end-user specifies the domain-level path that meets its QoS requirements and moves the responsibility of path computation from the edge router to the end-node. Alternatively, the end-user only marks packets with proper, previously agreed Type of Service (ToS) value, and the edge router selects a path which fulfills the end-user's QoS requirements.

Path computation proceeded by edge router can be based on measurements of QoS parameters taken by this node, i.e., the edge router processes received routing advertisements and computes two alternative paths which are as independent as possible. Additionally, each edge router monitors selected paths in order to ensure that they meet preconfigured parameters (packet loss, throughput, delay, etc.). Due to constant monitoring of the network, the edge router can always choose the better path (in term of QoS parameters) and avoid paths that are overloaded or congested. The authors show that such a computation is feasible and scalable thanks to the fact that only edge systems are responsible for path computation, while core routers only forward packets.

Proposed mechanism uses a variable-length header placed between the IP and the transport layer header. Such a header, in addition to fields such as *protocol, length, offset, reserved*, consists of two variable length fields: *reverse path* and *forward path*. Each WRAP-enabled packet carries information about the endpoint addresses (source and destination IP) and the addresses of relays on the path between source and destination.

The example presented in Fig. 12 shows how the proposed mechanism works. Suppose source node *S* sends a packet to destination point *D*. Such a packet can pass along three different domain-level paths: AS1-AS2-AS4, AS1-AS3-AS4, AS1-AS3-AS2-AS4. The end-node (S) computes two as paths which are as independent as possible, i.e., AS1-AS2-AS4 and AS1-AS3-AS4. Let us assume that at a given moment the link between routers RD and RF is congested and path AS1-AS3-AS4 was selected for a new packet. When the packet comes to the next node, its source address in the IP header is changed to the address of this node, and its destination address is changed to the next IP from a *forward path* list of the WRAP header. At the same time, the IP address of the previous hop is placed in the *reverse path* field. Thanks to this, packet length remains constant during the forwarding process and allows the destination node to use the same path in the return direction.

15. Routing deflections

Source routing is an established solution. It is, however, not popular, as it does not fit into the Internet model in which ISPs set and govern all routing policies. In 2006, the authors of [65] proposed a method to allow source-induced path diversity by routing deflections. Routing deflection is a mechanism which allows routers to force packets out of their shortest-path by forwarding them to neighbours other than those shortest-path would indicate.

Fig. 13 shows how deflections work. A standard routing protocol sets a path between A and F through nodes B and C (bold arrows). Normally, all packets in this relation follow this path. However, router B can deflect certain packets to router E (based on packet tagging), thereby disobeying the protocol's established path. Router E forwards the deflected packets normally – in this case, directly to F. Similarly, router A can deflect packets to router D. In this way, routing deflections can provide multipath transmissions. In this example, untagged packets follow the original path. Packets tagged with X are deflected by router B, and packets tagged with Y are deflected by router A. This presented scheme is very simple, yet there are two issues. Firstly, which deflections are permissible and

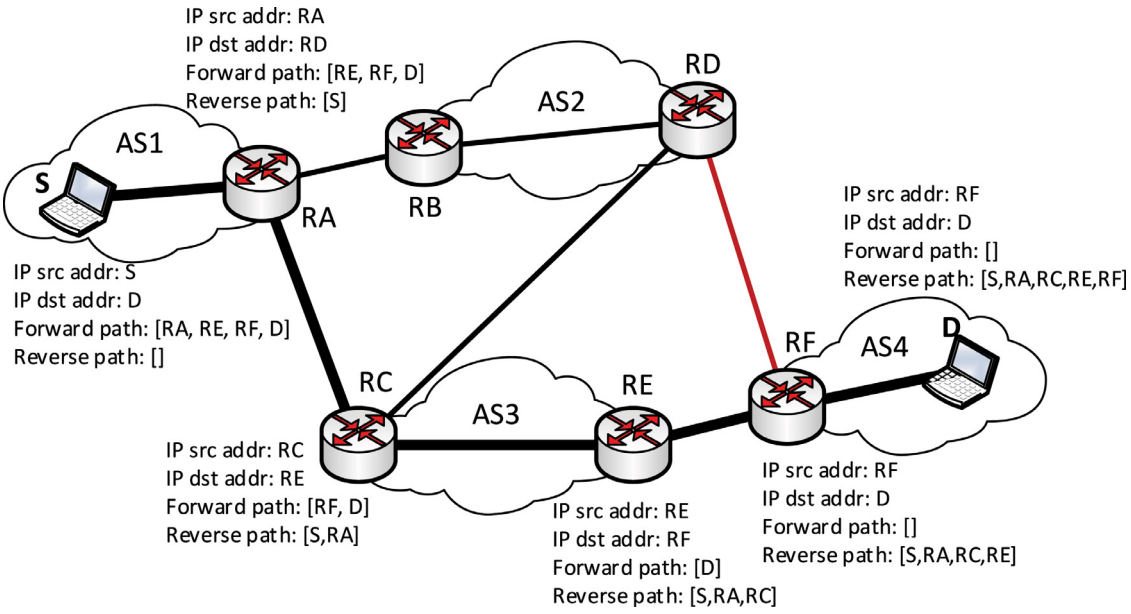


Fig. 12. WRAP relaying example.

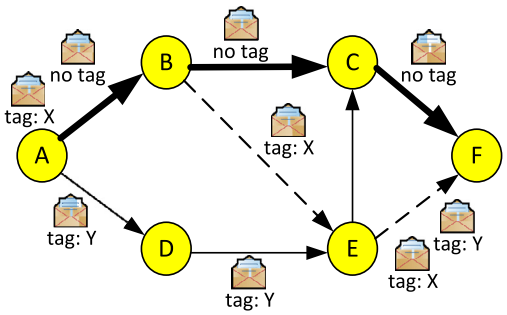


Fig. 13. Path ABCF is the shortest-path selected by a routing protocol; router B can deflect packets from the path ABCF to router E which forwards them to F.

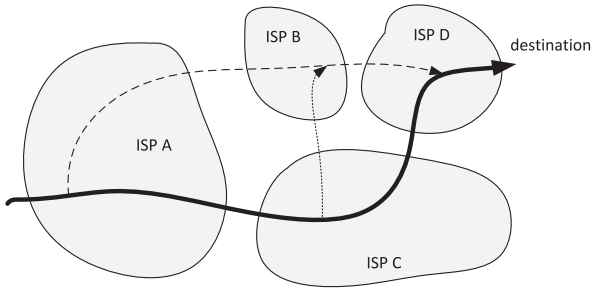


Fig. 14. Depending on the point of deflection, the resulting path changes.

which are forbidden in order to avoid loops. Secondly, how can the source trigger these deflections.

In [65], the authors propose rules which must be obeyed by routers while deflecting packets. The simplest rule is that a router can deflect a packet to its neighbour, if that neighbour has a lower path cost to reach the destination. These costs are either already signalled in a distance-vector routing protocol, or can easily be computed in a link-state protocol. As the neighbour has a lower cost path to the destination, it will definitively not return the packet to the deflecting router, and hence a loop will never occur. This rule is safe; however, it is very harsh, resulting in the omission of possible loop-free deflections. Therefore, the authors proposed two additional rules, which are more complicated, but also safe. It is proven in the paper that all three rules provide paths that are loop-free and reach their destinations even when there are standard routers along the way.

To trigger packet deflections, tagging of the packets is envisioned. The authors propose two approaches: to insert a 32-bit tag right after the IP header, or to employ certain bits from the IP header itself. The first option gives more flexibility, at the cost of carrying additional bits. The second assures complete backward compatibility, as there are no new fields carried by the packets. Regardless of the method used, the additional information carried in each packet can trigger certain routers to use deflections instead of shortest-path routes.

Fig. 14 shows how a user can select a path. Depending on the inserted tag, different routers will trigger the packet deflection procedure. Without the deflections, the path goes through ISPs A, B and D. A source can put a tag which forces early hops to deflect packets. In this case, the deflection procedure will occur inside ISP A and will go through ISP B and D. When the tag indicates late hops to deflect packets, the path is altered inside ISP C. Although users cannot explicitly define paths for their packets, by choosing tags they can influence the path change until they are satisfied with the performance the path provides.

Routing deflection is a scalable solution. It remains decentralized and can be introduced incrementally. It provides end-users with the possibility to exchange the original path proposed by the ISPs. Even though users cannot explicitly define paths as in source routing, they obtain a powerful tool through which they can blindly influence path changes until they are satisfied with the result.

16. Multipath interdomain routing (MIRO)

MIRO [66] is an approach to enhance the BGP-based interdomain routing. The authors realize that there are, for practical purposes, two approaches currently used to realize routing: BGP and source routing. In BGP, although it allows ASes to apply a plethora of routing policies, the protocol requires each router to use a single route for each destination prefix. This leaves many ASes with little control over the paths their traffic takes once it leaves the

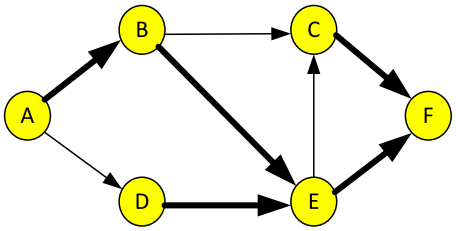


Fig. 15. BGP-based single path routing to AS F; bold arrows show routes chosen by BGP.

domain. In particular, multipath transmissions are not possible. Source routing, on the other hand, leaves transit domains with very little control and introduces scalability and security problems. The lack of control for ISPs is a significant hindrance to the adoption of source routing. This approach, however, provides multipath transmission possibilities and increases traffic flow diversity.

MIRO stands in the middle of these two approaches. It offers some flexibility in choosing the best path, while giving transit domains control over the traffic through their infrastructure. In MIRO, BGP is the main source of the route calculation process, i.e., routers learn their default routes through standard BGP operations. Additional paths can be installed as a result of the agreement between pairs of domains. Two ASes can negotiate the conditional use of additional paths. In this way, MIRO provides backward compatibility with standard BGP, which allows for incremental deployment. Also, for most traffic, the simplicity of BGP is retained.

One of the motivations of MIRO is the following example. Consider the network presented in Fig. 15, in which there are 6 ASes, A to F. Each AS chooses its route to F. Now, let us assume that A is not satisfied with the performance provided by E and does not want E to carry its traffic to F. However, A has no choice since both ASes B and D forward their traffic to F through E. Simply asking B to permanently switch to an alternative path is not a viable option because B may be satisfied (quality-wise and/or financially) with the status quo. MIRO provides the possibility for A and B to negotiate and establish a path ABCF only for certain traffic that A cares about. Such an agreement can be subject to additional charges between operators.

MIRO provides the following features:

- **AS-level path selection** — flow's path is chosen by the AS instead of the end-user: this is simpler and more scalable,
- **Negotiation for alternative routes** — BGP learns the default route and negotiates to learn additional routes if needed,
- **Policy-driven export of alternative routes** — advertised alternative routes are subject to each AS policies which allows them to control what they offer,
- **Tunnels to direct traffic on alternate paths** — after the negotiation is successful, a tunnel is established and may be used as needed.

16.1. Route retrieval

To retain scalability, ASes do not advertise alternative routes voluntarily. Instead, they may be polled by other ASes when such demand occurs. For example, if A is dissatisfied with default route ABEF, it may ask B to advertise alternatives. AS B answers based on its policy and desires. If B is not MIRO-compliant, the poll is simply rejected and nothing changes.

16.2. Bilateral negotiation between ASes

Negotiations begin when one AS asks another to advertise alternative routes to a destination. Negotiations should only be triggered if none of the current routes satisfy the desired requirement, such as bandwidth or delay. The ASes which originate negotiations do not need to be adjacent. In the example, A can negotiate the BCF route with B and create a tunnel to B, or can establish a tunnel to C and use default routing from that point, which also results in the BCF path.

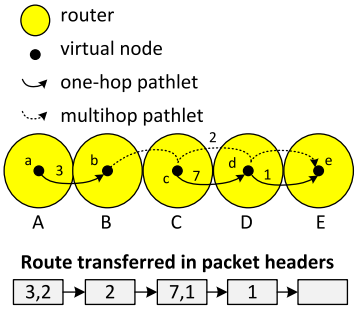


Fig. 16. A pathlet routing example [67].

16.3. Tunnels

Upon negotiations, two ASes establish an IP- in-IP tunnel for carrying packets to the tunnel's egress node. In the egress node, the packets are decapsulated and forwarded according to standard routing procedures. The tunnel's egress node, therefore, is a node from which the default remaining path to the destination leads across desired ASes. In the example, if A wants to establish a path ABCF, the tunnel needs to be constructed to one of the nodes in B from which the default path to F goes through C.

17. Pathlet routing

The aim of Pathlet Routing [67] is to provide the possibility of using multipath data transport between and across autonomous systems. It allows network operators to concurrently use various routing policies. In Pathlet Routing, each autonomous system advertises fragments of paths, called pathlets, to its peers. A pathlet is a sequence of virtual nodes (vnodes), and the connections between them. A vnode is created by the AS operator. A vnode may be created in each router within the AS, or on a subset of them. By not creating a vnode on a physical router, an administrator excludes such a router from path selection procedured. The node is established by setting a proper routing table for the vnode and assigning an identifier. The AS number and vnode identifiers construct a globally unique identifier of each vnode. Vnodes create a sort of virtual topology, and Pathlet Routing can be perceived as source routing over this topology.

It is assumed that any pair of nodes may disseminate information about pathlets. A node may request several pathlets from a router or pathlets may be sent to, at least, its AS neighbours. As soon as information about pathlets is distributed among nodes, a router may use a pathlet or sequence of pathlets to route a packet to a given destination. It is assumed that using a routing policy in a given AS only a limited number of known pathlets is distributed to other nodes. A node announces pathlets which form the shortest paths to all reachable destinations and some additional pathlets beyond the previous set. It is assumed that the higher the degree of an AS, the more additional pathlets may be advertised.

This fairly complicated architecture can be presented by a simple example. Fig. 16 illustrates one scenario of packet processing in pathlet routing. In the figure, there are five physical routers (*A, B, C, D* and *E*), each having its virtual node (*a, b, c, d, e*). In this example, the mapping of physical to virtual nodes is one-to-one (a vnode is created on each physical node), although this is not mandatory. Initially, routers learn the vnodes of their neighbours. After that,

they can construct one-hop pathlets to their direct neighbours: *A* constructs a pathlet to *B* and assigns a forwarding identifier (FID) of 3, *C* constructs a pathlet to *D* with FID of 7, and *D* constructs a pathlet to *E* with FID of 1. FID is an identifier which points to the respective pathlet in the routing table. For example, entry 7 in the routing table of vnode *C* instructs the router *C* to forward the packet to *D*. After one-hop pathlets are constructed, they can be used to create multihop pathlets. In the example, *B* builds a pathlet $b \rightarrow c \rightarrow d \rightarrow e$.

Now, when the pathlets are constructed, a packet arrives in router *A* with the following list of pathlets in its header: 3, 2. This instructs router *A* to forward the packet along pathlet 3 and strip FID 3 from its header. The packet then arrives at *B* with one pathlet: 2. A multihop pathlet 2 indicates that a packet is to be forwarded to node *C* with further indication of pathets 7 and 1. Therefore, router *B* forwards the packet to node *C* and puts the combination 71 into its header, as this was the combination used to construct the multihop pathlet number 2. Nodes *C* and *D* follow the same procedures. Finally, the packet arrives at node *E* with an empty route, which indicates that it has reached its destination.

Once pathlets are established, they can be used in the way presented in the example above. The challenge is in creating pathlets. In [67], the authors also present a pathlet dissemination algorithm. This algorithm is a path vector algorithm, which works in a similar way to how BGP notifies nodes of the existence of IP prefixes. The distinct feature of this algorithm is that routers propagate an arbitrary subset of their known pathlets. The operator is responsible for indicating which pathlets can be used and which should not. Even though Pathlet Routing resembles source routing, it is the operator's responsibility to show which pathlets can be used. Therefore, the operator remains in control of the traffic while providing multipath capabilities. The algorithm presented in the paper is only a one feasible solution, and others can be developed and used (e.g., RSVP is only one feasible signalling protocol for Integrated Services).

18. Yet another multipath routing protocol (YAMR)

YAMR is to enable multipath transmission for inter-domain traffic [68]. It constructs a set of paths between edge nodes. As a result, transmission is resilient to any single failure of interdomain links. Moreover, the signalling traffic generated by edge nodes is minimized by reducing routing updates.

YAMR is based on two components:

- A mechanism for computing paths,
- A technique for minimizing churn by localizing routing updates.

The first component is called the YAMR Path Construction mechanism (YPC). It is based on the assumption that a set of alternate paths are computed. Such paths are deviations from BGP's default path. It means that the paths with the lowest cost are selected. Moreover, alternate paths are chosen in such a way that failure of any link (node failures are not considered)from the shortest path cannot break transmission (unless all policy-compliant paths between source and destination nodes fail). The proposed mechanism has one important disadvantage – computing many paths results in higher control plane messaging overheads than BGP.

The second component allows the limiting of the signalling traffic and solves the mentioned problem. After any link failure, all nodes which use paths composed of this link have to be notified about the failure. As a result, more traffic is transmitted in a network and the protocol can become unstable. YAMR solves this problem implementing the mechanism for limiting the signalling traffic described in detail in section 18.2.

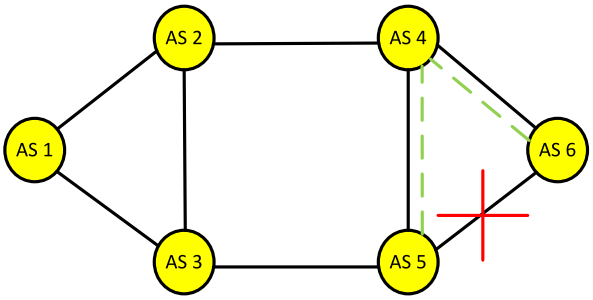


Fig. 17. The hiding mechanism in YAMR.

18.1. YAMR Path construction

YPC works similarly to BGP. The first step in YAMR is to compute the default path between two ASes, which is the same as in BGP. In the next step, for each interdomain link which belongs to the default path, another path (if possible and which does not contain this link) is calculated. The mechanism works under three assumptions:

- widest-advertisement – which means that information about paths is broadly distributed; without this assumption, some paths may become unavailable,
- No dispute wheels – which means that oscillations are minimized; without this assumption, YPC and BGP may become unstable,
- Next-hop policy – it is necessary in the context of the second YAMR mechanism (hiding technique).

As a result, it is possible to maintain default and alternative paths in edge routers and to use the best of them according to network state. Alternative paths are signed. Packets are labelled, which is necessary to select paths for them. A 32-bit field in a packet header is used to mark packets. YAMR routers need to have forwarding entries for default and alternative paths (if such entries are different). As a result, YAMR forwarding tables are $k+1$ times greater than for BGP, where k is the average interdomain path length. The mechanism presented in the next section minimizes communication overhead by reducing YAMR's churn below the BGP level.

18.2. Mechanism for limiting the signalling traffic

The signalling traffic is not broadcast to all ASes after a link failure. The ASes connected to a failed link are known as 'hiding ASes'. First, they try to redirect traffic after failure by themselves. If this is impossible, they immediately inform their neighbours, which can solve the problem or inform their neighbours recursively. This operation is illustrated in Fig. 17.

When AS5 detects a link failure, it redirects traffic through AS4 to AS6. ASes 1, 2 and 3 are not informed about the failure. If it had not been possible to redirect traffic in AS5 to AS4, AS5 would have been obliged to inform AS3 about the failure. Further, AS1 or AS2 would be informed about the failure recursively.

Such a mechanism allows the minimizing of the signalling traffic among ASes after failure in a network. However, it is not easy to ensure that loops are not created and the transmission is continued in a network. However, the authors of [68] ensure that this solution, along with the YPC algorithm, allows the limiting of loops and guarantees substantially lower churn compared with simple YPC.

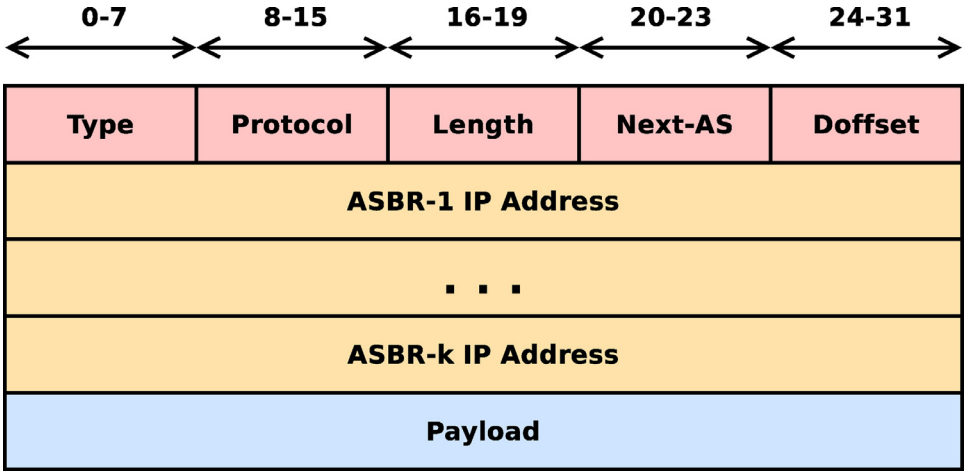


Fig. 18. The AMIR header together with an encapsulated payload.

19. Another multipath interdomain routing (AMIR)

AMIR was first introduced in [69] and represents an alternative way of routing which leverages an extended cooperation between adjacent Autonomous Systems on the Internet to collect more information about the AS-level network topology and construct multiple paths between the source and destination Autonomous Systems. It is proposed as a mechanism offering new features that are missing in BGP. In the original proposal [69], AMIR obtains the primary path to a destination from the local BGP route table[1] and then, based on this information, it determines alternative paths. The authors argue that at the cost of increased transmission-, storage-, and processing-related overheads, it is advantageous to provide multipath interdomain routing as a paid service to selected customers. To avoid major scalability issues, the authors propose a method which determines the alternative paths based on the limited number of available interconnected Autonomous Systems.

19.1. Route retrieval

In AMIR, alternative paths are determined based on the information from Autonomous Systems which are located on the main path for a given relation. The primary path is obtained from the local BGP route table[2]. The first AS on the main path queries the following ASes on the path for their adjacent ASes and aggregates the response into a set. In the next step, AMIR tries to determine several alternative paths traversing only those ASes included in the set. If some adjacent ASes cannot set up a candidate path, a negotiation procedure is employed to resolve the issue. The authors propose an example concept of a negotiation algorithm and claim that bilateral negotiation strategies in general may stimulate the deployment of multipath routing on the Internet.

19.2. The main components

AMIR consists of the following two modules:

- **Multipath Service Agent (MPSA)** — responsible for the computation of alternative AS-level paths; this task includes the collection of topology-related information from ASes adjacent to those on the main path, as well as all necessary negotiations between adjacent ASes;
- **Multipath Forwarding Agent (MPFA)** — encapsulates and forwards packets with an AMIR header (Fig. 18).

Whenever an Autonomous System Border Router (ASBR) receives the first packet belonging to an application handled according to a paid multipath service, it requests the MPSA associated with the ASBR's domain to determine the candidate paths according to the proposed Multipath Discovery algorithm. In the next step, it prepares an appropriate AMIR header for the selected path, and then inserts it at the beginning of each packet and forwards the traffic to the next node. The last ASBR on the path strips the AMIR header from all packets, while the intermediate ASBRs only modify its content.

The AMIR header shown in Fig. 18 consists of five fields and a list of 32-bit IP addresses of ASBRs forming a path. The five initial fields are as follows:

- **Type** — the packet type identifier of AMIR;
- **Protocol** — the identifier of the upper layer protocol (e.g., IP);
- **Length** — number of IP addresses on the list;
- **Next-AS** — offset of the next hop address (relative to the beginning of the list);
- **Doffset** — offset of the payload segment (relative to the beginning of the packet).

Although it is not clear whether other address families and interdomain routing protocols (used as the source of primary paths to destination ASes) may also be supported in AMIR, the authors claim that their simulation-based evaluation has confirmed that AMIR is actually feasible. Furthermore, the capability of AMIR to introduce new paths through the negotiation process is considered as an important feature which is not present in other solutions like YAMR (Section 18) and Path Splicing (Section 10).

20. BGP Extended multipath for transit domains (BGP-XM)

The BGP extended multipath for transit domains (BGP-XM) is a routing mechanism which explores many redundant paths on the Internet [70]. This mechanism enables deployment of interdomain multipath and enables (flow) load balancing between different paths connecting distant sites (with many ASes on the way) on the Internet. The mechanism has been designed in such a way that it preserves all existing BGP features, so it is backward compatible. The main concept of BGP-XM is to use already existing route aggegation capability to aggregate multiple paths into single route advertisement. Therefore, unlike the Add-Paths BGP extension, it does not require both peers to explicitly support it.

[1] It is not stated explicitly that this is a requirement, and whether any other inter-domain routing protocol could be used as a source of primary paths to destination ASes.

[2] See the previous comment in Section 19.

BPG-XM preserves business relationships, the route filtering based on policy routing. The common traffic engineering techniques offered by BGP are preserved: the COMMUNITY attribute can be used for informing about local preferences, the *AS_PATH* prepending, and the *MED* attribute represents metrics between neighbouring ASes. Also, the *LOCAL_PREF* attribute is used for representing the business preferences of a given AS owner. The paths placed in the routing table are free from routing loops since *AS_PATH* is used for this purpose. The whole mechanism is stable under non-conflicting routing polices. Each operator at the level of AS can define routing polices on their own. The BGP-XM imposes the path diversity requirement: different numbers of traversed ASes and diversity in AS traverse sequence. The mechanism is incrementally deployable, which means that deployment in one AS does not require routers equipped with this mechanism in other ASes. The mechanism also enables techniques for controlling the size of routing tables. The standard BGP may send multiple path announcements, however, the most recent is taken into account. In order to preserve multipath information, the authors of BGP-XM confined multipath information in a single update. In this way, they obtained a backward BGP compatible update announcement pattern, and the resulting update is related to many paths. This approach also enables incremental deployment of interdomain multipath. Each prefix is advertised together with a few BGP attributes. Different paths for the same prefix can possess different attributes; some of the selection and mapping procedure has been designed in order to define common values for a particular set of paths.

Routers which operate using the BGP- XM mechanism may distribute traffic for the same prefix using disjoint paths. The BGP aggregation procedure prevents the creation of routing loops in the case of multipath transmission [71]. The path assembling algorithm used by BGP-XM aggregates some BGP attributes for each prefix reached by many paths. The *NEXT_HOP* attribute indicates the address of one of the routers running the path assembling algorithm. The ORIGIN attribute is set to the maximum value of this attribute for all routes chosen for aggregation. The *LOCAL_PREF* and *MED* must be the same for all selected routes in order to be announced in aggregated form.

The BGP-XM router preserves standard BGP policy routing path selection. After receiving the BGP message, the received routes are filtered according to defined filtering rules. The BGP-XM router applies an algorithm for selecting paths, called K- Best Route Optimizer (K-BESTRO). This algorithm works in four steps (Policy Filtering Rules, Ranking Rules, Assembling Filtering Rules, K-Best Selection Rule). In the first step it discards routes which are excluded by input routing policies. The routes with highest *LOCAL_PREF* and lowest *MED* are analysed further. An operator defines the Unequal Length MultiPath (ULMP) parameter, which indicates the length diffence in *AS_PATH* of allowed routes. ULMP indicates how much longer than the shortest path (number of ASes on *AS_PATH*) a particular route can be. In addition, IBGP routes are removed if EBGP routes are present for the same prefix. In the second step, the path ranking procedure is applied to routes from the previous step. In the third step, a set of paths is constructed; these routes can be assembled together in a single BGP update. In the final step, the first K routes from the set are chosen for the path assembling algorithm. The K parameter is specified by the operator, and it limits its usage of resources by multipath procedures. The details of K-BESTRO selection procedure are described in [70].

The BGP-XM assembling algorithm establishes the *AS_PATH* attribute for multipath announcement. It works in the following sequence:

1. Pick one of the shortest paths from the multipath set. Let this path be Shortest Path (SP).
2. Create an empty *AS_SET* S.
3. For every AS number from other paths and not present in SP, add it to *AS_SET* S.
4. If SP already containes an *AS_SET*, merge it with S.
5. Else, if S is not empty, move the rightmost AS to S and append S to SP.
6. If the assembled path is advertised through an EBGP session, prepend the local AS number to the *AS_PATH*.

In order to illustrate the operation of the BGP-XM assembling algorithm, let us consider three paths to the same prefix: $P_1 = (1, 2, 3, 8)$, $P_2 = (4, 5, 6, 8)$, $P_3 = (7, 8)$. The numbers represent AS numbers. The shortest path is P_3 and it does not contain *AS_SET*. According to point 2, we create an empty *AS_SET* denoted by S and we put into it all AS numbers from other paths not present in SP: $S = \{1, 2, 3, 4, 5, 6\}$. After applying the rules from point 5 we obtain the *AS_PATH* $P = (7, \{1, 2, 3, 4, 5, 6, 8\})$. The obtained aggregated path has the same length as the SP.

Part V Comparison and conclusion

21. Comparison and contrast

In this section we compare the presented techniques for multipath interdomain transmission. We analyse a routing type in a mechanism, how paths are chosen and set up. Finally, we estimate the complexity of the mechanisms, signalling methods and time scales.

21.1. Setup features of inter-AS multipath mechanisms

In strict routing, the signalling procedure is used, which specifies the path, node-by-node, that must be visited by packets on the way to the destination node. On the other hand, loose routing describes nodes which have to be on a packet's route, but it is not necessary for packets to visit these nodes in fixed order. We also check whether paths are selected in ingress or core nodes, and whether the routing process is centralized or distributed. All solutions described in previous sections are compared in Table 3.

BGP Add-Paths and BGP-XM have been designed in such a way that they can be implemented incrementally, and they can work in a heterogeneous environment (standard BGP routers can cooperate with extended BPG routers). Path selection is done locally by a router based on information from BGP, a few paths to the same destination can simultaneously exist in a routing table, and packets may be routed via these paths. There is no path setup procedure; each router decides the availability of a path to a destination, so this is a distributed way of path selection.

In YAMR, the routing type can be depicted as loose. Alternative paths are selected based on the BGP announcements, which can change in time, e.g., as a result of link cost changes. Decisions about paths selected for multipath transmission are taken inside the network (each node plays a significant role in process of selecting paths).

The routing type in STAMP is dependent on the BGP attributes. This is why it is loose. Two BGP processes are active in STAMP. As a result, two paths are set up in ingress nodes. As BGP processes are not centralized, the STAMP routing process is also distributed.

In DIMR, two disjoint paths between two nodes are established. While they can be chosen differently under the same conditions, the routing type should be depicted as loose. The decisions about routing are taken by several nodes in the network core and this is a distributed process.

In the SR network both strict and loose routing may be used. A source routing may be initiated at any point in the network and both centralized and distributed path setup may be used. However, so far the focus is on the centralized approach.

Table 3
Comparison of inter-AS multipath mechanisms.

Algorithm/Protocol		Routing type		Path choice		Path setup	
		Strict	Loose	Ingress	Core	Central.	Distrib.
Available mechanisms							
1	GMPLS	✓	✓	✓	✓	✓	✓
2	BGP-Add-Paths		✓	✓			✓
3	LISP		✓		✓		✓
4	Segment Routing	✓	✓	✓	✓	✓	✓
Moderate research interest							
5	NIRA		✓	✓		✓	✓
6	Platypus		✓	✓			✓
7	Path Splicing	✓		✓			✓
8	STAMP		✓	✓			✓
9	DIMR		✓		✓		✓
Marginal research interest							
10	BANANAS		✓	✓			✓
11	WRAP		✓	✓			✓
12	Routing deflections		✓	✓			✓
13	MIRO	✓			✓		✓
14	Pathlet routing	✓		✓	✓		✓
15	YAMR		✓		✓		✓
16	AMIR	✓		✓			✓
17	BGP-XM		✓	✓			✓

BANANAS proposes two options for a new path setup: *Explicit-Exit Forwarding* and *Explicit AS-Path Forwarding*. When the former is based only on selection of exit ASBR for a given traffic aggregate, the latter specifies more precisely the domain-level path. A path setup is proceeded distributively by each domain. In both cases, the path can be established loosely.

In MIRO, alternative interdomain paths are created by the operators. The functionality allows alternative paths for certain parts of traffic on top of standard BGP-established paths. Paths are established inside the network — end users cannot choose paths. The path setup process is distributed, as it is an enhancement to distributed BGP protocol.

Path splicing is based on the strict routing scheme. Paths are chosen at ingress nodes and configured in a distributed way.

Pathlet routing represents a strict routing approach. The path is chosen by the ingress node and is put into packet headers. Once that is done, routers must obey the chosen path and send packets accordingly. There is no room for choosing paths. The path choice process, however, is performed both by the ingress node and core nodes. The ingress node chooses a path from pathlets that are advertised by the core nodes. Core nodes are responsible for creating pathlets, thereby, providing possible paths for ingress nodes to choose from. The path setup is distributed, as the path creation process operates without any central entity.

In LISP, an interdomain path between border routers of LISP enabled autonomous systems is found by the BGP protocol. Thus, the routing type, path selection and path setup are the same as for BGP. The decision on the next hop for a packet is taken by each router in the core using BGP metrics. It is a distributed process; LISP itself has no influence on BGP path selection. However, if communicating LISP enabled autonomous systems are multi-homed, multiple paths between them may exist if an EID (or EID group) is mapped to at least two RLOCs with the same *priority*. The decision on how to take advantage of multiple existing paths and how to manage the traffic, as well as assignment of EIDs to RLOCs, is under the control of the ISP. The ISP can manage the path choice by dynamic changing of mapping and/or policy-based assignment of EIDs to RLOCs. Concurrent multipath transferring is also possible, e.g., if the end host is aware of the existence of multiple paths and uses the MPTCP protocol.

In routing deflections, the ingress node does not explicitly choose a path. Instead, it asks the network to deflect packets from its original path, which is unknown to the ingress node. However, by marking packets differently, the ingress node can choose which deflection (ergo which path) to use. Therefore, the path choice is performed in the ingress node.

In AMIR, alternative paths for a traffic flow are determined by the Multipath Service Agent once the first packet is handled by the related ASBR. The path discovery process is based on additional topology-related information about the ASes adjacent to those on the main path. In the original proposal, the primary path is obtained from a local BGP table.

In Platypus the source system calculates a new path (different from that calculated by commonly used routing protocols) using one of a number of well known multipath computation algorithms (not addressed in the article). The calculated path does not require that all hops will be specified strictly. Path setup is distributed — there is no central controller, each edge system computes a new path independently.

WRAP is based on the LSRR [64], so path selection is done by an ingress system. In WRAP not all intermediate hops have to be specified — loose path. In such a case, default routing among them will be used. Each edge system independently calculates new paths, so path setup is distributed.

GMPLS uses a flexible approach to path computation and set up. In nominal situation, GMPLS implements RSVP-TE for path setup. In special cases, GMPLS is equipped with advanced path computation features aiming at the increase of network resiliency. Multiple paths can be established using formal methods from graph theory, with the aim of creating edge-disjoint, vertex-disjoint or physically disjoint paths.

Since NIRA gives the user a possibility to choose domain-level routes, its routing mechanism should be perceived as loose. As mentioned in Section 8, NIRA splits a route into a up-graph (sender part) and down-graph (receiver part), and uses addresses to represent each part. Path choice is performed by a user, who chooses both source and destination addresses used for forwarding. The path setup process is divided between end user and provider. An ISP uses TIPP to provide a user with addresses and routes associated with them. If the user wants to be reached using a specific

Table 4
Features of multipath mechanisms.

	Multipath in ...	Signalling	Complexity	Time scale
Available mechanisms				
1	GMPLS	CR-LDP, RSVP-TE	high	long
2	BGP Add-Paths	routing protocols	medium (BGP level)	short
3	LISP	BGP (+ protocol for map service)	low	short
4	Segment routing	packet header	low	long
Moderate research interest				
5	NIRA	packet header, TIPP, NRRS	medium	long
6	Platypus	packet header	flexible	short
7	Path Splicing	packet header	low	short
8	STAMP	BGP attributes	low	long
9	DIMR	packet header	low	medium
Marginal research interest				
10	BANANAS	packet header	medium	short
11	WRAP	packet header	high	long
12	Routing deflections	packet header	medium	short
13	MIRO	external	medium	long
14	Pathlet routing	packet header	high	long
15	YAMR	routing protocols	low	long
16	AMIR	external + packet header	high	long
17	BGP-XM	routing protocols	medium (BGP level)	short

route, he has to register the address corresponding to this route in NRRS.

21.2. Complexity and operating features of inter-AS multipath mechanisms

In this section, we analyse complexity and operating features of inter-AS multipath mechanisms. We describe whether a mechanism uses signalling to set up paths or not, and if so, what type of signalling is used. Next, we estimate the mechanisms' complexity and timescale. By timescale we mean the estimated duration of established paths (expected duration for paths in the 'on' state) in non-failure network conditions. It does not include the time required for path computation or setup. All solutions described in previous sections are compared in Table 4.

BGP Add-Paths is an extension of BGP. It changes the format of route information exchanged by BGP protocol. Therefore, in order to make use of it, both peers must support it. Routers supporting the Add-Paths extension can coexists with routers which does not support it. Therefore, this extension can be deployed incrementally.

BGP-XM does not modify the information exchanged by standard BGP. The main change is related to the processing and interpretation of exchanged information done by BGP-XM routers. Although a BGP-XM router may have more routing table entries, persistence of entries is the same as in a typical BGP network.

YAMR uses BGP-updates. However, signalling in YAMR is limited by reducing the number of routing updates. The complexity of YAMR is relatively low. YAMR chooses alternative paths among those announced by the BGP protocol. As a result, the setup time is long and selected paths do not change frequently.

In STAMP, two BGP processes are active. The signalling is based on the BGP attributes used for selecting alternative paths. The process is not complex, and it usually takes a short time to set up multiple paths between two nodes in the network. Similarly to YAMR, paths are set up for long time.

No additional signalling protocols are used in DIMR. Decisions about routing are taken by nodes based on information from packet headers. As a result, complexity is low. However, it takes some time before all paths are properly selected in the network. Paths may change in the network, which is why we estimate the timescale as medium.

In the SR network signalling is based on information carried by a packet header (using IPv6 header or MPLS label). There is no need to use label distribution protocols (e.g., LDP or RSVP) since a routing protocol is used to carry information about labels. Additionally, global labels are introduced. As a result, the operation, maintenance and troubleshooting of MPLS networks is simplified and less time consuming. The estimated duration of the paths is determined by the BGP protocol.

BANANAS proposes two different capabilities in the context of interdomain routing: *Explicit Exit Routing* and *Explicit AS-Path Routing*. For the former, the destination IP of the packet is overwritten with the IP of the exit ASBR. This causes the packet to be forwarded to a given exit ASBR. Next, the ASBR replaces the destination IP with the original destination IP stored in the packet and forwards a packet using information stored in the regular routing table. As a consequence, a different path can be selected. For the latter, e-PathID field stored in the packet is used. This field consist of a hash of an AS-level path sequence that a given packet should traverse. In BANANAS the path setup is in short timescale - the path can be specified per each packet.

In MIRO, signalling required to establish paths is not specified. The information needs to be passed in some way – probably similar to how paths are advertised through BGP. The overall complexity of the path setup process is assessed as medium, as it requires negotiations between operators. The path setup process is long-term. Once established, paths do not change unless necessary.

Path splicing is a relatively simple and flexible technique which might be deployed without introducing major changes to the existing infrastructure. Signalling is based on additional information stored in the packet header. The proposed solution is able to respond quickly to network failures, and it does not rely on the control plane with respect to the discovery of alternative routes. The actual forwarding paths depend on the selected splicing bits stored in the packet header.

In Pathlet routing, additional information which determines the packets' path is carried with each packet. The mechanism in general is quite complex, as it involves a pathlet creation and distribution mechanism. There is also additional information which needs to be maintained on every router. The timescale for paths is long, as pathlets are fairly static.

Deployment of LISP in current networks is not complex and can be done incrementally. Border routers of stub ASes must be up-

dated to support LISP. No changes are needed to the routing protocol (BGP) in the core network. Since the multiple paths between LISP enabled ASes are set up by the BGP protocol they are subject to changes as in legacy core networks (e.g. due to failures, congestions or changes in announced network prefixes). Thus the path setup and update are done on a short timescale (as for BGP). In turn, if we consider the LISP layer itself, a signalling protocol is needed to obtain EID-RLOC mappings from the LISP mapping system.

Routing deflections provides a possibility to change the packet's original path by diverting and forwarding it to an alternative next-hop than the one it is supposed to go to. The indication which router should perform the deflection is carried inside each packet. If the packet carries no such information, the deflection is not performed. The complexity of this proposal is low - this is a very easy mechanism which does not involve path establishment. Each action is performed on-the-fly. This is also why the path creation process is short-term, as it concerns each packet separately.

Operation of AMIR depends on external (between MPSA and ASBRs) and in-packet signalling. To ensure that packets are forwarded along the intended paths, AMIR stores the necessary information in the AMIR packet header, which is modified along the path and stripped by the last ASBR in the sequence. The authors of AMIR argue that at the cost of increased transmission-, storage-, and processing-related overheads, it is advantageous to provide multipath interdomain routing as a paid service to selected customers. As we can see, AMIR is relatively complex, but the configured paths do not change very often, unless it is necessary.

Platypus uses information stored in a proprietary header that consist of a list of next hops. The mechanism requires changes in router software in order to properly interpret data stored in such a packet. The path selected by a source is valid only for a given packet — short timescale.

Since WRAP calculates and stores at least two different domain-level paths to each reachable autonomous system it requires huge computational resources. Stored paths are then monitored for QoS parameters. Selected paths are stored for a long period of time.

GMPLS offers a complete control framework for network operators to run a multi-service and multi-domain network with differentiated services and flexible granularity of paths' resources. Probably the most advocated features related to GMPLS-based path setup are advanced protection (characterized by very fast reaction for failures) and restoration (featuring intensive usage of signalling function in order to coordinate switching and to make reprovisioning of resources for paths). With a global view of resources, path setup mechanisms are efficient and relevantly fast. Applicability issues of GMPLS in Multi-Region (MRL) and Multi-Layer networks were recently addressed in the novel RFC 5212 [21].

As long as only customer-provider relationships are present in the access network, NIRA does not introduce any overhead and the path is encoded using only source and destination addresses. In the case of peer-to-peer relationships between ASes, additional addresses are needed in the packet header to encode the path. NIRA requires the running of two new entities in the network, namely TIPP and NRRS. The overall complexity of the NIRA architecture is assessed as medium, as it requires cooperation between ISP and users, i.e., the ISP is responsible for the maintenance of TIPP, and users have to keep NRRS updated. In normal operation, path setup time depends mainly on Round-Trip-Time value, and in case of failure on set time-out value. When set up, paths do not change for a long time.

22. Future of inter-AS multipath routing

Interdomain traffic is transmitted through paths selected by using BGP. In this paper, we describe and analyse several mechanisms that allow for multipath interdomain transmission. All of them are promising solutions and can be implemented. However, only a few of them have a real chance of being used in a global network. In this section, we present a short analysis on the possible future of the presented mechanisms.

BPG Add-Paths and BGP-XM seem to be a very promising mechanisms for Internet multipath transmission. They can be deployed incrementally, and standard BGP routers can communicate with routers employing them. Deployment of these mechanisms increases Internet path diversity and capacity. Currently, the main BGP improvement efforts are being concentrated on reducing the size of the routing table. BPG Add-Paths and BGP-XM provide more paths to the same destination, what results in routing table growth. But BGP-XM preserves policy based routing functioning in the pure BGP. Thus selective multipath entries for dedicated destinations may increase bandwidth. Application of policies performing route aggregation together with BPG Add-Paths or BGP-XM multipath can provide more efficient router operation in the future Internet. Not all routers on the Internet have to maintain full BGP tables; in some regions, only reduced tables are required. In these regions the network capacity can be increased by using the BGP multipath feature, simultaneously keeping the size of routing tables limited. With the proliferation of SDN orchestration of networks, BGP multipath management will be more simple and manageable. As a result, there is space for BPG Add-Paths and BGP-XM application in the context of SDN.

YAMR and STAMP, in our opinion, do not have much chance for implementation in operators' networks. They attract few researchers and network device manufacturers. DIMR, on the other hand, as an extension of PDAR, attracts researchers and, as a mechanism based on BGP, may be interesting for manufacturers. PDAR can be implemented in a relatively easy way and the functionality of BGP-based interdomain routing will not be deteriorated.

Segment Routing is one of the multipath interdomain solutions which have the biggest chance to be implemented on a wider scale. Although the standardization process is not finished, some vendors already have SR ready devices in their portfolio. Adoption of BGP extensions, so far specified in IETF draft versions, may greatly speed up usage of SR by network operators.

Although the main design goal of LISP was not to support multipath transmission, it seems to be a promising solution. An important feature of LISP is that it can be deployed incrementally without the need to change current operations in the core network (in particular, no changes are needed to the BGP protocol) and the intradomain routing. Only border routers in a LISP capable autonomous systems need to be upgraded to support LISP. LISP does not need to be deployed at once by all ISPs. LISP based multipath can be introduced with no overhead to routers' forwarding tables; only a small signalling overhead related to the mapping service may appear. All available paths between two stub ASes are stable and created by the BGP protocol. The decision on using two or more concurrent paths is left to the operator of stub AS and will reflect its policy. Increasing deployment of LISP seems to be highly possible in the future. Operators of stub-networks will take advantage of its multipath capabilities if it offers some benefits (e.g. better quality, cost savings etc.).

Network equipment with GMPLS-capabilities has been offered for years by main equipment vendors. However, the proliferation of such GMPLS-capable equipment is somehow not relevant to the plethora of universal and flexible functions being offered by this control plane (and to some extent also management plane) framework. In spite of this limited and quantitatively unimpressive applicability, GMPLS offers an excellent set of reference concepts and functions that give receipts for resource provisioning, traffic engineering, protection, restoration, service-oriented routing and management.

MIRO, Pathlet routing, Routing deflection, AMIR and Path Splicing have not developed into commercially available solutions. One or two publications which have appeared did not spark interest. This does not negate the idea; however, the operators, who would be a driving force in implementing these ideas, did not see a strong enough incentive.

BANANAS, Platypus and WRAP extend source routing with a set of new features. Each of them can be introduced in-crementally, i.e., not all routers have to be aware of mechanism specific operations. Despite this, there is marginal interest from researchers and the market. NIRA was proposed in 2003, and since that time has not attracted vendors and operators. In our opinion, this mechanism will not be implemented on a wide scale in the future.

One of interesting concepts which can be considered in future as possible solutions for interdomain architectures are Software-Defined Networks (SDN). While SDN is not a new concept, its renaissance has been observed in recent years. However, so far, its popularity arises mostly in single domain networks. We believe that this concept can and will be used successfully in future for multipath transmission on the Internet. Before that, the scalability problems observed for SDN must be resolved. One of the most popular elements of the SDN architecture is the OpenFlow protocol responsible for communication between controller and forwarding elements. This protocol opens new possibilities for routing modifications by users or operators without using expensive devices. SDN should ensure easier network control, new services, and decreasing operational costs.

Moreover, Information Centric Networking (ICN) paradigm, that shifts from the current host-centric approach to information-centric one, can be considered as a future solution implementing multipath interdomain transmission. There exists very good survey papers addressing *state-of-the-art* of ICN [72–75]. In the ICN concept data routing can be closely related to the name resolution mechanism and those two functions can be integrated (*coupled*) or independent (*decoupled*). Over the years researchers proposed many differentiated architectures in which data routing may relay on well known protocols (e.g. OSPF, BGP, source-routing) or may implement novel approaches (e.g. Bloom filters, DHT). However, as addressed in [73] in the case of interdomain transmission many of ICN-based mechanisms require a global view of the network, have scalability issues and introduce quite big overhead of signalling packets into network.

In next section, we analyse challenges identified by us, which should be taken into consideration before implementing a new multipath interdomain transmission mechanism.

22.1. Challenges and requirements for optimal inter-AS multipath architectures

Several challenges have been identified in term of inter-AS multipath transmission. They are listed in Table 5. Addressing these challenges can limit the barriers for the implementation of interdomain multipath solutions. Operators usually use BGP which allows them to select one path between a pair of ASes. Load balancing can be used between two directly connected ASes. Operators are reluctant to implement more advanced solutions. They are aware of the importance of interdomain traffic and they do not want to incur risks related to the implementation of such mechanisms. In particular, operators cannot risk loops appearing in their networks.

The first, and one of the most important challenges, is scalability. Most of the mechanisms described in this paper can work successfully in relatively small networks. It is difficult to ensure the efficient operation of an interdomain multipath mechanism. Particularly in solutions which need a controller to set up multiple paths, e.g., in SDN, it is a challenge to coordinate information exchanged between ASes. A set of controllers from different domains composing a control plane can be considered as a possible solution to the scalability challenge. Such an organization of the network may need to upgrade or even replace existing hardware, which in turn should result in increased computing power and elasticity.

One of the most important problems relating to scalability is the growing size of the routing tables of Internet routers. Maintenance of huge routing tables and lookup procedures becomes real challenge. Provision of multipath mechanisms on the Internet scale increases resilience, but it simultaneously results in routing table size explosion. Resilience cannot be a strong argument for introducing multipath transmission at the inter-AS level. Current BGP has enough abilities to provide resilience.

Another factor which prevents vendors and operators from implementing multipath solutions is route stability. Route flapping is a very undesirable effect. Some of the presented mechanisms have been inspected for stability, and simulation shows that route stability is preserved. BGP-related mechanisms usually show good stability (ex. BGP-XM). Some of the presented mechanisms are dedicated to limiting routing table size - for instance LISP or Segment Routing. The fact that multipath transmission may increase the bandwidth and Internet capacity is not enough to strongly interest vendors and operators. Perhaps more mature solutions which combine strategies limiting table size together with multipath transmissions will attract more attention. Similar objections are also indicated to the solutions based on flow switching, and potential flow table explosion is considered. The SDN based solutions are vulnerable to this problem.

The next crucial challenge of multipath interdomain transmission is the ability to prevent the creation of a loop and to break already established loop(s). The existence of a loop has a far reaching consequence in a multi-domain multi-operator scenario. If a loop is created, then data conveyed by a transit AS may be caught in the loop and, as a result, due to the TTL limitation, removed from the network. This means that a service delivered by a provider in a distant part of the network may be blocked. Such an incident not only has technical consequences but also financial and legal ones. Increasing the number of paths which may be used in a network increases the difficulty in managing the network, data flows and used paths.

Manageability is a significant feature for interdomain multipath transmission. It is not easy to manage a huge amount of traffic, especially sent in a network through different paths. The control plane should not only be scalable but also efficient, and should ensure the proper operation of many entities. The management process should take into consideration also possible frequent changes being observed in the network.

Additionally, a growing number of paths increases the burden placed on the ability to prepare, use and coordinate routing polices. With contrary goals and preclusive business, legal and technical environments of operators of autonomous systems, it may be hard to introduce multipath. Broad use of multipath transmission needs to be accepted, understood, and used by managers and technical staff with a clear advantage seen by financial departments. It seems that enforcement on routing policies should be performed smoothly and automatically with more self organization than in current networks.

One of the fundamentals of BGP and its popularity is stability and reliability. BGP allows the maintenance of connectivity among ASes even in the case of link failures. It can be more difficult to ensure network reliability when multipath interdomain transmission is allowed. On one hand, a network controller may have to observe the state of a higher number of paths than in a network without multipath mechanisms. On the other hand, in the case of a link failure it may be easier to use another path to carry user data.

When we provide new mechanisms, especially new control planes, fault tolerance is a real challenge. Not only network links

Table 5
Challenges and requirements for optimal inter-AS multipath architectures/mechanisms.

	Challenges to existing inter-AS multipath architectures	Requirements for optimal inter-AS multipath architectures
Scalability	to upgrade or replace existing hardware	to change devices to increase computing power and elasticity
Size of routing tables	to react to increasing size of routing tables in hardware	to limit routing table registrations or to aggregate flows
Routes stability	to provide stability in network	to minimize route flapping
Loop prevention/mitigation	to prevent creation/usage of routing loops	to ensure smooth transport of data
Manageability	to react effectively to changes in the network	to manage a higher number of outgoing inter-AS links
Operation & Administration Maintenance	different paths for the same transmission make it harder to troubleshoot/diagnose problems	more memory and processing power needed to manage a single transmission
Reliability	to ensure co-existence of protection/restoration mechanisms with inter-AS multipath algorithms	to ensure sufficient network resources and efficient reliability mechanisms
Fault tolerance	to eliminate or reduce failures of network elements	protection/restoration mechanisms ensure short breaks in transmission and high performance when failure occurs
Cost	to minimize cost of implementation of new mechanisms and network operation	to ensure effective network devices at a rational cost
Power consumption	to reduce power consumption by usually overprovisioned network resources	to ensure that power consumption is rational

but also the control plane must be reliable. For example, redundancy for network controllers has to be assured. Network nodes may have many more operations to conduct. As a result they are subject to potential failures. Protection/restoration mechanisms implemented for network devices should ensure short breaks in transmission and high performance when failures occur.

The cost of implementation of new mechanisms for multipath inter-AS transmission may be very high. Core network devices are very expensive and should be changed rationally. Moreover, implementation of a control plane (especially with central controllers) and providing inter-operator agreements can increase the cost of implementation of new mechanisms. All new proposals have chances for implementation if the cost of this implementation is lower than potential profits. In most cases, it is not easy to estimate the cost of implementation on a global scale. What is clear is that network devices should work effectively at a rational cost.

Implementation of additional mechanisms usually results in additional demand on power (both computational and electricity). The same is observed for most mechanisms analysed in this paper. A separate control plane with network controllers, additional algorithms to be used by network devices, and usage of additional links, may mean that more power will be consumed. On the other hand, some links may not be overprovisioned, as they are currently. Network operators should balance network operation costs in their networks to enable rational transmission expenses.

In spite of all these challenges and difficulties it seems that in the end all these obstacles will be removed and automatic control of a multipath multi-domain network will be implemented. Looking at the previous twenty years and all the improvements visible in networking, it seems that it is fully possible.

23. Conclusion

In our previous work [1] we have shown that operators can provide multipath transmissions inside their domain quite easily, by choosing one of the available solutions. However, the solutions are not popular due to their complexity and the fact that operators rarely observe the necessity to use multiple paths inside their domain.

This survey shows that there are numerous proposals for providing multipath transmission between domains. These can be more appealing for both operators and end-users. Therefore, the incentive to use them is greater than in the previous case. However, the problem with introducing interdomain multipath transmissions is similar to the transition from IPv4 to IPv6. IPv6 was specified in 1998, and 17 years later, even though most currently used devices support it, its introduction on a global scale has still not happened. Obviously IPv6 works here and there, but because it is a ubiquitous protocol, its replacement is a very complicated operation.

The situation is the same with interdomain routing. As it all relies on one protocol (BGP), any changes are almost impossible to introduce. In the case of IPv6, a tremendous amount of effort and a lot of time has been devoted to providing strategies for seamless, incremental deployment. If there is ever a will to implement an interdomain multipath solution, after having decided on a single one, a similar amount of effort will need to be made to assure smooth deployment. Furthermore, routing is also connected with network politics - solutions that were established over the years by all operators. Forcing anyone to make changes might not be beneficial for all the parties involved. Therefore, there is definitely a long road ahead.

Acknowledgment

The research was carried out with the support of the project "High quality, reliable transmission in multi-layer optical networks based on the Flow-Aware Networking concept" founded by the Polish National Science Centre under project no. DEC-2011/01/D/ST7/03131.

References

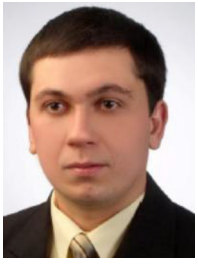
[1] J. Domzal, Z. Dulinski, M. Kantor, J. Rzasa, R. Stankiewicz, K. Wajda, R. Wojcik, A survey on methods to provide multipath transmission in wired packet networks, Comput. Netw. 77 (2015) 18–41.
[2] A. Ford, C. Raiciu, M. Handley, S. Barre, Architectural guidelines for multipath TCP development, IETF RFC 6182 (2011).
[3] A. Modirkhazeni, N. Ithnin, O. Ibrahim, Secure multipath routing protocols in wireless sensor networks: A security survey analysis, in: Network Applications Protocols and Services (NETAPPS), 2010 Second International Conference on, 2010, pp. 228–233, doi:10.1109/NETAPPS.2010.48.
[4] S. Adibi, S. Erfani, A multipath routing survey for mobile ad-hoc networks, in: CCNC 2006. 2006 3rd IEEE Consumer Communications and Networking Conference, 2006., 2, 2006, pp. 984–988, doi:10.1109/CCNC.2006.1593185.
[5] S.K. Singh, T. Das, A. Jukan, A survey on internet multipath routing and provisioning, IEEE Commun. Surv. Tutorials 17 (4) (2015) 2157–2175, doi:10.1109/COMST.2015.2460222.
[6] C. Xu, J. Zhao, G.M. Muntean, Congestion control design for multipath transport protocols: A survey, IEEE Commun. Surv. Tutorials PP (99) (2016), doi:10.1109/COMST.2016.2558818. 1–1
[7] E. Mannie, Generalized multi-Protocol label switching (GMPLS) architecture, IETF RFC 3945 (2004).
[8] D.W. Fedyk, L. Berger, Generalized MPLS (GMPLS) data channel switching capable (DCSC) and channel set label extensions, IETF RFC 6002 (2015).
[9] MARBEN GMPLS, MARBEN GMPLS https://www.marben-products.com/gmpls/gmpls-overview.html, 2015,
[10] D. Walton, D. Cook, A. Retana, J. Scudder, Advertisement of multiple paths in BGP, IETF Internet Draft (2002).

[11] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, The locator/ID separation protocol (LISP), IETF RFC 6830 (2013).
[12] L. Iannone, D. Saucez, O. Bonaventure, Implementing the locator/id separation protocol: design and experience, Comput. Netw. 55 (4) (2011) 948–958. Special Issue on Architectures and Protocols for the Future Internet
[13] OpenLISP webpage, 2015, URL http://www.openlisp.org.
[14] LISP-Lab Project, 2015, URL http://www.lisp-lab.org.
[15] 2016, Open Overlay Router. URL https://www.openoverlayrouter.org.
[16] LISPmob, 2015a, URL http://lispmob.org.
[17] CISO LISP webpage, 2015b, URL http://lisp4.cisco.com.
[18] 2015, Open Networking Foundation. URL https://www.opennetworking.org.
[19] X. Yang, Nira: A new internet routing architecture, SIGCOMM Comput. Commun. Rev. 33 (4) (2003) 301–312.
[20] X. Yang, D. Clark, A. Berger, Nira: A new inter-domain routing architecture, Netw. IEEE/ACM Trans. 15 (4) (2007) 775–788.
[21] K. Shiomoto, D. Papadimitriou, J.L. Roux, M. Vigoureux, D. Brungard, requirements for GMPLS-Based multi-Region and multi-Layer networks (MRN/MLN), IETF RFC 5212 (January 2008).
[22] E. Rosen, A. Viswanathan, R. Callon, Multiprotocol label switching architecture, IETF RFC 3031 (2001).
[23] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, RSVP-TE: Extensions to RSVP for LSP tunnels, IETF RFC 3209 (2001).
[24] L. Andersson, P. Doolan, N. Feldman, A. Fredette, B. Thomas., LDP Specification, IETF RFC 3036 (January 2001).
[25] B. Jamoussi, L. Andersson, R. Callon, R. Dantu, L. Wu, P. Doolan, T. Worster, N. Feldman, A. Fredette, M. Girish, E. Gray, J. Heinanen, T. Kilty, A. Malis, constraint-Based LSP setup using LDP, IETF RFC 3212 (January 2002).
[26] J. Lang, link management protocol (LMP), IETF RFC 4204 (October 2005).
[27] K. Kompella, Y. Rekhter, OSPF extensions in support of generalized multi-Protocol label switching (GMPLS), IETF RFC 4203 (2005).
[28] K. Kompella, Y. Rekhter, IS-IS extensions in support of generalized multi-Protocol label switching (GMPLS), IETF RFC 5307 (2008).
[29] J. Scudder, A. Retana, D. Walton, E. Chen, Advertisement of Multiple Paths in BGP, Internet-Draft draft-ietf-idr-add-paths, URL http://tools.ietf.org/html/draft-ietf-idr-add-paths.
[30] A. Retana, Advertisement of Multiple Paths in BGP: Implementation Report, Internet-Draft draft-ietf-idr-add-paths-implementation, URL http://tools.ietf.org/html/draft-ietf-idr-add-paths-implementation.
[31] V. van˜den Schrieck, P. Francois, O. Bonaventure, BGP add-Paths: the scaling/performance tradeoffs, IEEE J. Sel. Areas Commun. 28 (8) (2010) 1299–1307, doi:10.1109/JSAC.2010.101007.
[32] P. Mohapatra, A. Simpson, P. Francois, K. Patel, R. Fragassi, J. Uttaro, Best Practices for Advertisement of Multiple Paths in IBGP, Internet-Draft draft-ietf-idr-add-paths-guidelines, URL http://tools.ietf.org/html/draft-ietf-idr-add-paths-guidelines.
[33] R. Fernando, P. Mohapatra, R. Raszuk, C. Filsfils, Fast Connectivity Restoration Using BGP Add-path, Internet-Draft draft-pmohapat-idr-fast-conn-restore, URL http://tools.ietf.org/html/draft-pmohapat-idr-fast-conn-restore.
[34] J. Scudder, A. Retana, D. Walton, E. Chen, BGP Persistent Route Oscillation Solutions, Internet-Draft draft-ietf-idr-route-oscillation-stop, URL http://tools.ietf.org/html/draft-ietf-idr-route-oscillation-stop.
[35] M. Coudron, S. Secci, G. Pujolle, Augmented multipath tcp communications, in: Network Protocols (ICNP), 2013 21st IEEE International Conference on, 2013a, pp. 1–2.
[36] M. Coudron, S. Secci, G. Pujolle, P. Raad, P. Gallard, Cross-layer cooperation to boost multipath tcp performance in cloud networks, in: Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on, 2013b, pp. 58–66.
[37] Y. Benchaib, S. Secci, C.-D. Phung, Transparent cloud access performance augmentation via an MPTCP-LISP connection proxy, in: Architectures for Networking and Communications Systems (ANCS), 2015 ACM/IEEE Symposium on, 2015, pp. 201–202.
[38] O. Bonaventure, G. Detal, C. Paasch, Use cases and operational experience with multipath TCP, ietf-mptcp-experience-04 (work in progress) (2016).
[39] X. Wei, C. Xiong, MPTCP proxy mechanisms, wei-mptcp-proxy-mechanism-02 (work in progress) (2015).
[40] R. Moskowitz, P. Nikander, P. Jokela, T. Henderson, Host identity protocol, IETF RFC 5201 (2006).
[41] E. Nordmark, M. Bagnulo, Shim6: level 3 multihoming shim protocol for IPv6, IETF RFC 5533 (2009).
[42] R.J. Atkinson, S.N. Bhatti, Identifier-Locator network protocol (ILNP) engineering considerations, IETF RFC 6741 (2012).
[43] A. Gurtov, T. Polishchuk, Secure multipath transport for legacy internet applications, in: Broadband Communications, Networks, and Systems, 2009. BROADNETS 2009. Sixth International Conference on, 2009, pp. 1–8.
[44] Y. Wang, X. Li, D. Liu, M. Chen, Optimizing cost and performance for concurrent multipath transferring using extended shim6, in: Communication Technology, 2006. ICCT '06. International Conference on, 2006, pp. 1–4.
[45] A. Feldmann, L. Cittadini, W. Mühlbauer, R. Bush, O. Maennel, Hair: Hierarchical architecture for internet routing, in: Proceedings of the 2009 Workshop on Re-architecting the Internet, in: ReArch '09, ACM, New York, NY, USA, 2009, pp. 43–48.
[46] M. Menth, M. Hartmann, D. Klein, Global locator, local locator, and identifier split (gli-split), Future Internet 5 (1) (2013) 67.
[47] S. Previdi, C. Filsfils, A. Bashandy, H. Gredler, S. Litkowski, B. Decraene, J. Tantsura, IS-IS extensions for segment routing, IETF Internet-Draft (2015).
[48] P. Psenak, S. Previdi, C. Filsfils, H. Gredler, R. Shakir, W. Henderickx, J. Tantsura, OSPFv3 extensions for segment routing, IETF Internet-Draft (2015).
[49] C. Filsfils, S. Previdi, J. Mitchell, E. Aries, P. Lapukhov, G. Gaya, D. Afanasiev, T. Laberge, E. Nkposong, M. Nanduri, J. Uttaro, S. Ray, BGP-Prefix Segment in large-scale data centers, IETF Internet-Draft (2015a).
[50] C. Filsfils, S. Previdi, B. Decraene, S. Litkowski, R. Shakir, Segment routing architecture, IETF Internet-Draft (2015b).
[51] C. Filsfils, S. Previdi, K. Patel, E. Aries, S. Shaw, D. Ginsburg, D. Afanasiev, Segment routing centralized egress peer engineering, IETF Internet-Draft (August 2015).
[52] S.B. S., G. A., G. P., K. A., W. J., S. A., Scalable segment routing a new paradigm for efficient service provider networking using carrier ethernet advances, IEEE/OSA J. Optical Commun. Netw. (2015).
[53] S. A., P. F., G. A., C. F., C. P., Experimental demonstration of segment routing, J. Lightwave Technol. 34 (2016) 205–212.
[54] B. R., H. F., K. M., L.T. V., Optimized network traffic engineering using segment routing (2015) 657–665.
[55] B. Raghavan, A.C. Snoeren, A system for authenticated policy-compliant routing., in: R. Yavatkar, E.W. Zegura, J. Rexford (Eds.), SIGCOMM, ACM, 2004, pp. 167–178.
[56] B. Raghavan, P. Verkaik, A.C. Snoeren, Secure and policy-compliant source routing, IEEE/ACM Trans. Netw. 17 (3) (2009) 764–777.
[57] M. Motiwala, M. Elmore, N. Feamster, S. Vempala, Path splicing, ACM SIGCOMM Comput. Commun. Rev. 38 (4) (2008) 27–38.
[58] C. Page, E. Guirguis, M. Guirguis, Performance evaluation of path splicing on the géANT and the sprint networks, Comput. Netw. 55 (17) (2011) 3947–3958.
[59] Y. Liao, L. Gao, R. Guerin, Z.-L. Zhang, Reliable Interdomain Routing Through Multiple Complementary Routing Processes, in: Proceedings of the 2008 ACM CoNEXT Conference, in: CoNEXT '08, 2008, pp. 68:1–68:6.
[60] D. Wu, Z. Wang, X. Yin, X. Shi, J. Wu, M. Huang, Dimr: Disjoint interdomain multipath routing, in: High Performance Computing and Communications 2013 IEEE International Conference on Embedded and Ubiquitous Computing (HPCC_EUC), 2013 IEEE 10th International Conference on, 2013, pp. 1382–1390.
[61] F. Wang, L. Gao, Path Diversity Aware Interdomain Routing, in: INFOCOM 2009, IEEE, 2009, pp. 307–315.
[62] H.T. Kaur, S. Kalyanaraman, A. Weiss, S. Kanwar, Bananas: An evolutionary framework for explicit and multipath routing in the internet, In SIGCOMM FDNA Workshop, 2003.
[63] K. Argyraki, D.R. Cheriton, Loose source routing as a mechanism for traffic policies, in: Proceedings of the ACM SIGCOMM Workshop on Future Directions in Network Architecture, in: FDNA '04, ACM, New York, NY, USA, 2004, pp. 57–64.
[64] Internet Protocol - DARPA Inernet Programm, Protocol Specification, IETF RFC 791, Internet Engineering Task Force,
[65] X. Yang, D. Wetherall, Source selectable path diversity via routing deflections, SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 159–170.
[66] W. Xu, J. Rexford, Miro: Multi-path interdomain routing, SIGCOMM Comput. Commun. Rev. 36 (4) (2006) 171–182.
[67] P.B. Godfrey, I. Ganichev, S. Shenker, I. Stoica, Pathlet routing, in: Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication, in: SIGCOMM '09, ACM, New York, NY, USA, 2009, pp. 111–122.
[68] I. Ganichev, B. Dai, P.B. Godfrey, S. Shenker, YAMR: yet another multipath routing protocol, SIGCOMM Comput. Commun. Rev. 40 (5) (2010) 13–19.
[69] D. Qin, J. Yang, Z. Liu, H. Wang, B. Zhang, W. Zhang, AMIR: Another Multipath Interdomain Routing, in: Advanced Information Networking and Applications (AINA), 2012 IEEE 26th International Conference on, 2012, pp. 581–588.
[70] J.M. Camacho, A. Garcia-Martinez, M. Bagnulo, F. Valera, BGP-XM: BGP extended multipath for transit autonomous systems, Comput. Netw. 57 (2010).
[71] Y. Rekhter, T. Li, S. Hares, A Border Gateway Protocol 4 (BGP-4), 2006, (IETF RFC 4271).
[72] X. Jiang, J. Bi, G. Nan, Z. Li, A survey on information-centric networking: rationales, designs and debates, China Commun. 12 (7) (2015) 1–12, doi:10.1109/CC.2015.7188520.
[73] G. Xylomenos, C.N. Ververidis, V.A. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K.V. Katsaros, G.C. Polyzos, A survey information-centric netw. res. 16 (2) (2014) 1024–1049, doi:10.1109/SURV.2013.070813.00063.
[74] M.F. Bari, S.R. Chowdhury, R. Ahmed, R. Boutaba, B. Mathieu, A survey of naming and routing in information-centric networks, IEEE Commun. Mag. 50 (12) (2012) 44–53, doi:10.1109/MCOM.2012.6384450.
[75] B. Ahlgren, C. Dannewitz, C. Imbrenda, D. Kutscher, B. Ohlman, A survey of information-centric networking, IEEE Commun. Mag. 50 (7) (2012) 26–36, doi:10.1109/MCOM.2012.6231276.

Robert Wójcik received his M.Sc. and Ph.D. (with honors) degrees in telecommunications from AGH University of Science and Technology, Kraków, Poland in 2006 and 2011, respectively. Currently, he works as an assistant professor at the Department of Telecommunications of AGH. He is the co-author of 10 international journal papers, 3 books, 3 patent applications and a number of conference papers. He has been involved in several international scientific projects, including: SmoothIT, (NOE) BONE and Euro-NF. His current research interests focus on Multipath routing, Flow-Aware Networking, Quality of Service and Network Neutrality.

Jerzy Domżał received the M.S. and Ph.D. degrees in Telecommunications from AGH University of Science and Technology, Krakow, Poland in 2003 and 2009, respectively. Now, he is an Assistant Professor at Department of Telecommunications at AGH University of Science and Technology. He is especially interested in optical networks and services for future Internet. He is an author or co-author of many technical papers, six patent applications and two books. International trainings: Spain, Barcelona, Universitat Politecnica de Catalunya, April 2005; Spain, Madrid, Universidad Autónoma de Madrid, March 2009, Stanford University, USA, May-June 2012.

Zbigniew Duliński received the Ph.D. degree in theoretical physics from the Jagiellonian University. He works at Faculty of Physics, Astronomy, and Applied Computer Science at the Jagiellonian University. He previously worked in the area of theoretical and experimental elementary particle physics. For 10 years he has been working on problems in telecommunication. He is currently working on management mechanisms in overlay networks and inter cloud communication. His researched interests include distributed computing, network management mechanisms and traffic engineering.



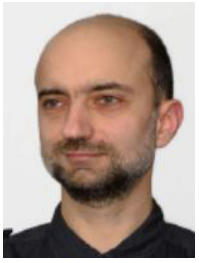
Grzegorz Rzym received his M.Sc. in Electronics and Telecommunications in 2012 and B.Sc. in Acoustic Engineering in 2013, both from AGH University of Science and Technology, Poland. Currently, he is a Ph.D. student at the Department of Telecommunications. His research interests cover management system design and implementation, traffic engineering and virtualization.

Andrzej Kamisiński is a research assistant in the Department of Telecommunications at the AGH University of Science and Technology in Krakow, Poland. He received B.S. and M.S. (with honors) degrees from the same University in 2012 and 2013, respectively. His current research goal is to develop new solutions to improve the reliability of telecommunication networks. He was involved in several scientific projects funded from Polish and European resources and worked with international teams during research stays in Oslo and Trondheim, Norway. He is the author or co-author of 5 book chapters, 10 conference and journal papers (including ACM Computing Surveys), as well as one patent application.

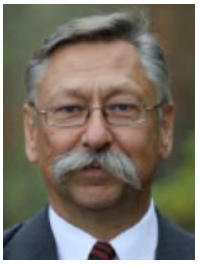
Piotr Gawłowicz received his M.Sc. and B.Sc. in Electronics and Telecommunications from AGH University of Science and Technology, Krakow, Poland in 2012 and 2014, respectively. Currently, he is working as researcher at TKN Group at TU Berlin, Germany. He is the author or co-author of several technical papers. He has been involved in national and European research projects. His research interests include software defined networking, wireless networks and simulation tools. He did a research internship in Panasonic R&D Center Germany in 2013, where he worked on enhancements for LTE. In 2014 under Google Summer of Code program, he contributed to development of ns-3 simulator.

Piotr Jurkiewicz received his B.Sc. and M.Sc. in Telecommunications from AGH University of Science and Technology, Krakow, Poland in 2012 and 2015, respectively. Currently, he is a Ph.D. student at the Department of Telecommunications. He was involved in many national and European scientific projects and is the co-author of several technical papers. His research interests include Software Defined Networking, flow aware networking, adaptive routing and fast packet processing. He is an open source software contributor.

Jacek Rząsa received an M.Sc. degree in telecommunications in 2001. Since then he has been working in the Department of Telecommunications at AGH University of Science and Technology. He has participated in research ordered by telecommunication operators and worked in many international projects. Jacek Rzasa is author and co-author of several research papers. His research interests focus on energy aware optical transport networks, traffic engineering in optical networks and Carrier Ethernet.



RafałStankiewicz received the M.Sc. and Ph.D. degrees in Telecommunications from AGH University of Science and Technology, Krakow, Poland in 1999 and 2007, respectively. He is employed as at the Department of Telecommunications of AGH. His current research interests focuses on networking techniques, QoS provisioning methods, performance modeling and evaluation, traffic management and optimization at network and overlay/application layers (including cloud traffic management) and information security. He is an author of several conference and journal research papers and co-author of two books. He actively participated in European research FP4, FP5, FP6 and FP7 projects. He is TOGAF 9 Certified.



Krzysztof Wajda received M.Sc. in Telecommunications in 1982 and Ph.D. in 1990, both from AGH University of Science and Technology, Krakow, Poland. In 1982 he joined AGH and is currently an assistant professor. He spent a year at Kyoto University and half year in CNET (France). He serves as a reviewer of a scientific journals and is TPC member of conferences. The main research interests include: traffic management, network reliability, control plane, network services. He participated in EU projects: COST 242, Leonardo da Vinci, BBL, LION, NOBEL, e-Photon/ONe(+), BONE, SmoothiT, SmartenIT. Dr. Wajda is the author of 6 books and over 100 technical papers.