Network Working Group
Internet-Draft
Expires: April 27, 2003

H. Berkowitz
Gett Communications
E. Aman
T. Eriksson
Telia Research AB
October 27, 2002

Routing Architecture Building Blocks: an Informational Taxonomy
draft-eriksson-rabbit-00.txt

Status of this Memo

Copyright Notice

Abstract

This document identifies and categorizes the components of routing, switching, forwarding, and addressing that may be used in routing architectures. The intention is to support the development of a new routing architecture for the Internet.

The addressing architecture, address allocation and assignment principles, and possibilities for renumbering are important aspects when designing a routing architecture. How routing information is learned and methods for distributing it are other issues discussed in

this document. A number of methods for data traffic forwarding are also described and evaluated.

Table of Contents

1. Introduction

Work has started on developing the next generation routing architecture for the Internet. The intention of this paper is to support this work by identifying and categorizing the components of routing, switching, forwarding, and addressing that may be used in routing architectures. Knowing what building blocks are available and knowing the strengths and drawbacks of each of them is essential when designing a new architecture.

This document avoids specifying any demands on future routing architectures or protocols. Such requirements can be found in [15] and [26]. The design decisions made when developing a routing architecture are governed by what is expected from the network. This document can give some guidance in selecting among available design choices based on requirements following from these expectations.

Even though we assume IPv4 and IPv6 to be the main transport protocols of the future Internet, we try not to limit ourselves in looking only at what has so far been used in IP routing. When possible, we evaluate designs from several suggested or implemented architectures. Current technologies are often cited as examples, but we do not take for granted that current protocols (as for example BGP, OSPF, or IS-IS) will continue to be used in a next generation routing architecture.

The document is written with as little prejudice as possible. Therefore we do not always adhere to describing techniques suitable for the current Internet architecture. It is already a reality that there are other forwarding paradigms in use in the Internet than the traditional hop-by-hop model. What we do take for granted is that the Internet will continue to consist of a large and probably growing number of networks under different administrative control where different actors do not always trust one another.

Different components of a routing architecture may to a large extent be selected independently from one another. For example, the mechanisms for distributing information may be independent from how routing decisions are made. A routing protocol, if well designed, can also be used for multiple network protocols including both connectionless and connection-oriented approaches.

2. Terminology

Terminology is a distinct challenge in developing new architectures. Many concepts have overlapping, and indeed contradictory, meanings developed over time. Other concepts are overloaded with multiple usages, leading readers to assume interdependencies that really do

not exist.

Some of the terms below were inspired by [7], which is an exhaustive BGP-centric terminology, and the Definitions section of [29] from the ForCES Working Group.

Butler Lampson said "no problem in computer science is insoluble with a sufficient level of indirection." Applying this philosophy to scalable networking, the keys to solubility are flexible and appropriate abstraction, scalable information distribution, and appropriate scoping/information hiding.

So, we begin with a set of abstractions:
Topological abstractions and labels, architectural planes, control plane abstractions, forwarding plane abstractions, and administrative abstractions. Use of abstractions is futher discussed in Section 7.2.

2.1 Topological Abstractions and Labels

Forwarding Element:
A logical entity that performs forwarding (q.v.) of traffic.

Control Element:
A logical entity that runs routing and/or signalling protocols and uses the information to instruct one or more forwarding elements how to process packets.

Network element:
An entity composed of one or more control elements and one or more forwarding elements. Usually referred to as either "router" or "switch".

Router:
A device that forwards data at the network layer. Routers may run some kind of routing protocol in the control plane for communicating with other routers.

Switch:
A device that performs forwarding of data traffic at the link layer. It also communicates with other switches for path establishment and possibly for exchanging routing information.

Host:
A device that can originate and/or receive traffic. A host does not perform forwarding or switching, but may send traffic that it originates to different forwarding elements depending on the destination.

Node:
A host, a network element, or a topological aggregate of nodes.

Interface:
A device a node can receive traffic from and send traffic to.

Network:
A set of interconnected nodes.

Link:
A device that provides communication between two or more nodes. Something that interconnects interfaces.

Flow:
A unidirectional association between a data source and one or more data sinks. Connections (q.v.) are usually bidirectional flows that involve the commitment of resources. All connections, therefore, are flows or sets of flows. Not all flows are connections, because a flow may only establish an end-to-end relationship with no resource commitment. Flows can be unicast or multicast.

Flow identifier:
A label used in the data traffic in order to identify the same flow. "Wild card" labels or matching rules may be used for flow aggregation.

Connection:
An association between two or more communicating parties used for communication between them. A connection may be implemented as a circuit. "Flow" also implies an association between parties, but "connection" has the additional implication that resources are explicitly committed, though not necessarily in the path.

Circuit:
A permanent or signalled path through a network, which requires state in the forwarding elements. A unidirectional circuit is also a flow, but not all flows are circuits. Bidirectional circuits consist of at least two flows.

Circuit identifier:
A label used to identify a circuit. The value of the label might be the same over all hops the circuit traverses or be local to each hop.

Route or Path:
The path from one point to another in the network. The two words are considered interchangeable in this document. The route may be

between physically connected devices or go through intermediate forwarding elements.

Next hop:
An address in a routing or forwarding table entry to which packets to a given destination should be sent by a network element. The next hop can either be connected to the same link as an interface belonging to the network element, or be an address located somewhere else in the network. The latter case is referred to as "indirect next hop" and implies that the network element has to perform one or more additional lookups in order to be able to forward traffic to the destination.

2.2 Architectural Planes

The tasks performed by network elements can generally be split into two logical parts; the control plane and the forwarding plane. These planes can be implemented in one physical device or be split into different components. Control plane functionality and forwarding plane functionality are implemented by a control element and forwarding element, respectively. For example a general-purpose "router" has both, while a route server has only control, and a distributed forwarder has only forwarding.

Control Plane:
Tasks performed in the control plane are the discovery of routing information, exchange of routing messages between control elements, and calculation of forwarding tables. Connection set-up is also included if applicable. Operations performed by the control plane are usually referred to as "routing".

Forwarding Plane:
The forwarding plane handles only the per-flow or per-packet forwarding as calculated by the control plane. It also performs policing and scheduling as well as services such as protection and load sharing. The forwarding plane is often said to perform "forwarding" or "switching".

2.3 Control Plane Abstractions

Routing:
The process of selecting paths through a network and exchanging the information used for selecting these paths. Note that a path may have different amount of detail; it may specify all the nodes in a path, some nodes, or a number of topological aggregates.

Signalling:
In this document the term "signalling" is used for the process of setting up a path. The path may be between a host and an ingress/egress device, or between forwarding elements. Different protocols may be used for these functions.

Routing Table or Routing Information Base (RIB):
A database in a control element containing information about reachability to destinations. The information in a RIB may be selected from several sources of information. A routing protocol may have more than one logically (not necessarily literally) distinct RIBs.

Policy:
Policy is "the ability to define conditions for accepting, rejecting, and modifying routes received in advertisements." [25].

2.4 Forwarding Plane Abstractions

Forwarding:
The process of receiving data traffic from an interface, determining the outgoing interface(s) and sending the traffic there. The outgoing interface is determined using a forwarding table and network layer or link layer information associated with the data traffic. Forwarding may include changing information in the header.

Forwarding Table or Forwarding Information Base (FIB):
The FIB is the data structure that is used for each forwarding decision that a forwarding element makes. It is generated from the RIB, but in contrast to the RIB it is generally optimized for high-speed lookup.

2.5 Administrative Abstractions

Network operator:
The organization that has administrative control of a part of the network.

Customer:
The organization or individual for which a network operator conveys and supplies traffic. The customer may be another network operator.

Allocation:
Long-term delegation by a registry (q.v.) of a resource such as a

block of addresses.

Assignment:
Short-term delegation of resource (e.g., address space) from the recipient of an allocation to its internal use or its customers.

Registry:
1. An organization responsible for stewardship of part of an address, name, or route space [20].
2. A database, often distributed, containing information on allocation/assignment of resources, and specific policy on their use [48].

3. Addressing

While addressing is one of the most common terms in network architecture, it is also a term with a wide range of definitions, some overlapping and some contradictory

This section will cover addressing fundamentals, alternative models for address binding, and allocation and assignment methods.

3.1 Names and Addresses

A name represents an entity and is used for referencing that entity independent of its actual connectivity in a network. An entity may be assigned different types of names. A name can for example have the form of a DNS name or a PSTN telephone number [23]. IPv4 addresses are sometimes de facto names, but that is not the primary motivation, in the modern Internet, for their use.

One confusing historical artefact of the current IPv4 architecture is that the IPv4 address has become semantically overloaded. It may be used as a persistent endpoint identifier, a locator, a dynamically assigned endpoint identifier, or even as a security identifier. In this document, the word "address" signifies any of the above or a combination of them. If the meaning cannot be concluded from the context, it will be specified.

In different architectures addresses are be assigned in different ways. Addresses can be assigned to different kinds of entities; an address may represent an interface, a node, or a specific function. Different types of addresses can be used for different kinds of entities. An entity may also be known simultaneously under several different addresses of the same kind for reasons such as multihoming and renumbering [6].

There are a number of alternative approaches regarding address structure, uniqueness, et cetera. Different parts in a hierarchy may use different hierarchical assignment schemes, or none at all. The assignment scheme used may not always be known by parties using the address. In a given host or network element, there can be independent addressing at different layers. In [30] Lear outlines a number of alternative properties of addresses:

- The length of an address may be fixed or variable.
- Addresses may or may not be structured.
- An address may be required to be globally unique.

These alternatives can be applied to any type of address. Below are some examples of interesting aspects from existing addressing architectures.

Topology Dependence
: Telephone numbers in the PSTN were originally largely topology-dependent, as that was required when using mechanical switches. Nowadays telephone numbers are often topology-independent below the country code, although there typically is at least administrative hierarchy in the national part.

Length
: For performance reasons, addresses used in packet headers generally have a fixed length.

Structure
: ISO NSAP addresses [36] are examples of structured addresses. An address starts with the initial domain part (IDP) which consists of an authority and format identifier (AFI) and initial domain identifier (IDI). The AFI designates mainly the syntax of the address and the format of the IDI, which specifies the network addressing authority. The rest of the address, the domain specific part (DSP), can in turn be structured and its format depends on the IDI value.

Uniqueness
: MAC addresses [18] are examples of globally unique addresses although strictly spoken uniqueness is only necessary on the same link for communication purposes. Each vendor of network cards is assigned an address block and may internally structure that address block among its products according to their preferred scheme. None of these hierarchy levels generally needs to be known when the MAC address is used by communicating parties.

3.1.1 Locations and Location Names

A location is a point in a network, for example the attachment point of a host to a network. A location name is a reference to a location and designates where something is located in a topology.

The Location Area found in the GSM system [34] is one example of a location. The Location Area Identity (LAI), i.e. the location name, identifies the location to the GSM network. An endpoint associated with one location might move and may need to change its Location Area over time because of this.

3.1.2 Endpoints and Endpoint Names

An endpoint is defined in [30] as one of the participants of an end-to-end communication. Some examples of useful characteristics of endpoint names discussed in a paper by Chiappa [13] are global uniqueness, topological insensitivity and portability.

An example can again be found in the mobile telephony system GSM [34]. The mobile terminal can be seen as an endpoint which has an endpoint name associated with it, the telephone number. Another example is the two endpoints of a TCP session for which the IP addresses constitute endpoint names that are used as a part of their identification.

3.1.3 Address Binding Models

As mentioned earlier one address might bind to more than one entity. An alternative approach is to have a one-to-one mapping between address and entity. In the following paragraphs we exemplify two alternative models regarding naming of endpoints and their locations.

One Address to Identify both Endpoint and Location
: In the current Internet architecture the address is used as both an endpoint name and a location name. This has implications on how straightforward things are to implement. Ideally, a mobile node would change its location name but keep the same endpoint name as it moves from one location to another. This is not possible in the current Internet architecture. A benefit from using the same name for both purposes is that there is no need for mapping between the two.

Two Separate Addresses to Identify Endpoint and Location
: There are Routing Architectures that use separate naming of endpoints and their location. Nimrod [11] for example, uses the concept of locators (location names) and EIDs (endpoint names).

3.2 Allocation and Assignment

The prerequisites for routing and forwarding depend to a large extent on how the addresses on which to make forwarding decisions are assigned to network nodes.

3.2.1 By Registries or Others

Registries represent a centralized approach to address assignment. An organization applies for a block of addresses from a registry and may in turn run a local registry for purposes of internal assignments.

Other ways of assigning address space could be geographical [19] or statistical approaches.

3.2.2 According to a Hierarchy or Not

A location name should preferably be hierarchically assigned. If hierarchically assigned, forwarding elements can make some form of longest prefix match without having to know about the existence of every host, but rather of larger aggregated prefixes for networks.

Antonov suggests in his Trivial Routing Architecture Proposal (TRAP) [2] a scheme where every network receives a power-of-2 sized block of addresses from a higher-level assignment. Such extremely hierarchical schemes can contribute to reducing the sizes of routing and forwarding tables. The suggested scheme includes automated renumbering (see Section 3.2.3) to maintain the address assignment hierarchy as networks grow or move. It should be noted that such schemes may be impossible due to political restrictions.

Using large addresses increases the possibility for hierarchical assignment somewhat; if addresses are not a scarce resource, a network can be assigned a larger address block than needed and chances are that the network will never need to apply for additional addresses as it grows.

3.2.3 Manual or Automatic Methods

Address allocation and assignment for networks in today's Internet is still a largely manual process whereby a customer gets its address blocks from its operator's larger block, allocated by a Local Internet Registry (currently RIPE NCC, ARIN and APNIC) which in turn has been allocated address blocks by IANA. This method enables the control that is deemed necessary as addresses are considered a scarce resource.

The assignment of addresses could be handled by an automated system. TRAP [2] suggests a hierarchy of address assignment servers providing dynamically assigned address blocks based on the utilization in a network. Design principles from the multicast address allocation architecture MALLOC [45] can be of interest if designing a similar scheme for unicast.

Automatic assignment schemes have obvious advantages as the human intervention is by necessity minimised, but every element in a network needs to support the scheme

3.2.4 Validity Time

An important aspect in address assignment is for how long the assignment is valid. Traditionally an IPv4 address block assignment has been considered more or less permanent, which has been a contributing factor in address space depletion and routing table growth. The current IPv6 address allocation and assignment policy [24]instead stipulates that address space licenses should be subject to renewal on a periodic basis. This results in a possibility to force renumbering of networks, which may be used for the benefit of the rest of the Internet.

3.3 Renumbering

Having the possibility to renumber hosts and whole networks in order to reflect a changed topology may be very useful for reducing the size of routing tables.

Host
: When hosts change their location in a network, mechanisms for automatically changing addresses might be used. Examples of such mechanisms include the stateful Dynamic Host Configuration Protocol (DHCP) [16] and the IPv6 stateless address autoconfiguration [47]. Special cases include host addresses with demand access (e.g., PPP with IPCP, with or without DHCP proxy), and host mobility.

Network
: Despite the availability of dynamic configuration protocols such as DHCP for hosts, renumbering a whole IPv4 network is still considered a painful and time consuming task. This is partly due to that the original IPv4 architecture assumed renumbering to be infrequent.

4. Sources of Routing Information

The first stage for the control plane in a network element is to find the information that it should spread to other parts of the network and the information to use for its forwarding decisions. Different kinds of routing information are discussed further below.

4.1 Information Discovery

Many types of routing information can either be configured manually or found automatically by a network element through probing, measuring or database lookup. In general, automatic information discovery facilitates the work done by the management staff and can eliminate human configuration errors. On the other hand automating can also lead to loss of exact control of the system resulting in that things may not work as intended. With manual configuration of all parameters and pieces of information, some of the dynamic nature of routing and ability to react to changes can be lost. Different methods may have to be chosen depending on the type of information. The design of the routing architecture can influence which is the best alternative.

4.1.1 Static

There are some cases where it is preferable to configure information statically when setting up a network. Reasons may be to get better control over the network and the information announced.

One example is statically configuring BGP prefix announcements instead of basing announcements on actual circumstances, i.e., internal reachability. The latter can cause unnecessary instability visible to the rest of network.

Static configuration, perhaps driven by an human-controlled provisioning system with varying degrees of automation, is the norm for SS7, the PSTN method of exchanging topological information.

4.1.2 Dynamic

Neighbour Discovery
: An interesting aspect of a routing system is how a network element discovers other network elements in the vicinity and how they decide to establish a connection between them. A network element can easily use a discovery protocol to find and start exchanging routing information with other directly connected network elements, subject to the constraint that they for example belong to the same area. Too much automation can contribute to unintentional adjacencies being formed, which in turn can cause

unintentional traffic paths.

Detection of Lost Neighbour
Network elements need ways to determine whether the link between them is still functional. Typically the link layer will inform the routing process if the link is lost and in addition some form of keepalive packets are often used for determining if a session with a peer is still up. If the path used for session communication is not the same as that used for traffic, additional complexity is added.

4.2 Information Export between Protocols or Protocol Levels

Future routing architectures may not have different protocols for reachability within domains and between domains (today's IGP/EGP split), but it may have similar constructs in its design or during a transition phase from the current architecture. In such cases it may be an issue how export between routing protocols or different levels of the same routing protocol should work. Also, should a network element run several routing protocol instances which are at different levels (such as having both an OSPF process and a BGP process in the same router which is common today) or should the necessary information only be exported between levels at the borders between domains?

It is possible to export information between different routing protocols in many of today's router implementations, but it is often avoided because of concerns that it can cause problems such as instability.

5. Information to be Distributed

The network elements of a modern routing system need different kinds of information on which to base their decisions. The various types of information have different properties with regard to how often the information needs to be updated, how far and fast it needs to be propagated, and where it is originated. In this section some types of information worth taking into consideration are discussed. Their properties such as reasonable time frames for update frequency and information propagation are also mentioned. These properties are of interest when deciding upon relevant distribution methods, listed with their properties, advantages in Section 6.

The most obvious information for a routing protocol to distribute are reachability information and/or information about the network topology. Recent protocols and extensions to existing protocols have added support for distributing other types of information useful for

controlling and optimising routing decisions.

5.1 Topology

Topology state routing protocols work by distributing information about the existence of connections between networks or nodes. They can also be referred to as map distribution protocols or link state protocols. Only a small amount of information needs to be distributed when there is a change in network connectivity.

The time it takes to distribute topology information affects the convergence time, i.e. the time it takes to move traffic to a new path when the one in use becomes unavailable. A convergence time of several seconds is quite acceptable for many of the services that the Internet is used for today, but sub-second convergence time at the forwarding level is required by telephony and other real-time applications, see for example Chapter 6 of [5] for a more detailed discussion. The convergence time for a particular event may be different in different parts of the network, as it is affected by the time it takes to propagate the information. See [7] for futher discussion of control plane convergence, in particular regarding BGP.

In particular, information about lost connections needs to be propagated fast in order for the traffic to converge onto another path. In many cases, however, only nodes that are close to the topology change need immediate knowledge about it, as nodes further away do not need to update their forwarding tables.

5.2 Reachability

Reachability information is distributed by for example distance vector protocols. Typically a prefix and a cost to reach that prefix is sent. In this discussion we regard a path vector protocol as a special case of a distance vector protocol; the main reason for including a path is to avoid loops (see Section 7.4) and the length of the path can be one of the routing decision criteria.

5.3 Shared Risk Links Groups

Paths that look separate to the networking layer may in reality be destroyed by the same backhoe or the same power outage. Sometimes it is desired to have guarantees that two different communication paths are really independent of each other. In order to make it possible to give such guarantees, either information from the lower layers or out-of-band configuration is needed. Such information may have different granularity; it may be necessary to determine if two paths pass through the same fibre, the same duct, or the same building. Further description of the Shared Risk Link Group (SRLG) concept can

be found in [38].

Information of this kind is expected to be mostly static, but any change in dependencies caused by rerouting on the link layer should preferably be visible to the routing system without significant delay.

5.4 Traffic Engineering

The objective of traffic engineering can be both to optimise resource utilization in a network and to enhance traffic performance [3]. Traffic engineering information include the currently available bandwidth and assigned administrative groups of links. Information of this kind may mainly be intended for use within the same domain.

It is recommended for stability reasons that router implementers make sure that rapid changes in available bandwidth do not cause rapid generation of new information [3]. The update frequency should thus usually be in the order of minutes to hours and it may not be necessary to propagate the information faster than within minutes.

Current examples are the traffic engineering extensions (mainly intended for use by MPLS) that have recently been added to OSPF [27] and IS-IS [31].

5.5 QoS

In order to offer routing based on QoS demands, it is necessary to exchange information about QoS parameters, such as the expected or guaranteed delay, jitter and throughput. The exchange of QoS parameters could be done in many ways. Examples include through signalling as done in RSVP [9] or by receiving service description messages as proposed in [8]. Nevertheless, spreading QoS information before traffic flow establishment can help network elements in making at least tentative decisions for which path to choose.

QoS parameters may change more or less regularly depending on the granularity of the information.

5.6 Policy

We think of routing policy as how to select routes, and information distribution policy as rules for what routing information might be sent to someone. These concepts are discussed in more detail in Section 8.2 and Section 7.1, respectively. Future routing protocols may need the possibility to formalize and distribute information about advanced policies.

The Routing Policy Specification Language (RPSL) [1] is an example of a language for specifying routing policies. Network operators can use it for sending information to routing registries about which routing information they accept from other networks and for whom they provide transit. There exist tools for converting the information in the routing registries into router configurations, but they are not widely used.

Transit agreements and similar policies typically change seldom (months to years) and usually through manual intervention in today's Internet. Future bandwidth broker and transit trading systems may change that, but we still expect such information to be reasonably stable. An update propagation time of minutes to hours should be enough.

5.7 Destination Location

Routing protocols to some extent distribute both information about what the network looks like and where different destinations can be found. In protocols such as IP a destination in the form of a number of consecutive addresses is represented by a prefix and a network mask.

The location of particular destinations can be separated from topology information. The information about which destinations are advertised by a node will typically be more stable than the information about how to reach the node. Separating topology information from destination location facilitates multi-protocol routing and enables topology changes to be propagated faster.

Adding a bit more complexity, an announcement of the location of a destination can also specify if the destination information may be aggregated at another location and if it can be multihomed, i.e. announced at other nodes as well. This can facilitate optimisations in other parts of the network.

If networks are designed in a way that the information about where a destination is attached only changes when there is a deliberate renumbering and all unexpected or sudden changes are regarded as topological, it is not unreasonable to allow up to several hours for updating information about destination location. The frequency of updates depends on the address allocation and update strategy as discussed in Section 3.

OSPF and IS-IS are examples of link-state protocols, which is a type of map distribution protocol. They separate the location information from the topology, but the two types of information are propagated the same way.

5.8 Indicating Unreliability or Insufficient Information

In the current Internet issues can arise when a router is known in the IGP before it has received all the necessary reachability information from BGP. This can cause blackholing of traffic if other routers try to use it for transit, see RFC 3277 [32]. Even if future routing architectures don't include the IGP/EGP split, similar effects may have to be taken into consideration.

The use of the IS-IS overload bit, further discussed in Section 11.1, is a an example from current practice. Its use is in this situation is described in [32].

5.9 Time

In the list of information that needs to be distributed in a routing system we would also like to mention that time-synchronization between routers (see Section 11.1 for an example of how it can be useful). Accuracy of a few milliseconds is reasonable to achieve and many routers already today use NTP for timekeeping.

6. Information Distribution

The information that a routing system distributes between its elements can follow different distribution models. These are discussed in this section as well as the topologies that can be used for information distribution.

In Section 6.1 we describe distribution models for routing information. This is an important aspect of any routing architecture and the same architecture may even use several distribution methods.

An architecture might be built with a hierarchy of protocols where one routing protocol depends on another in order to reach its peers. For example this is how BGP is typically used internally within an autonomous system. In this case different information distribution topologies might be formed. These are discussed in Section 6.2

6.1 Distribution Models

Different kinds of information have different properties with regard to how often updates are expected to occur, what part of the network needs to know about the change, and how fast those network elements need to know about it. Some kinds of information may also need to be propagated only to a small part of the network, while other information has to be known by a large amount of network elements. A routing system could potentially utilize different distribution models for different types of information.

6.1.1 Propagation of Local Decisions

This way of distributing information is used in today's distance vector protocols. A network element receives information from its peers, calculates its own routing decisions (such as the best path to a destination) and propagates its decision together with updated calculated information to its peers.

An inherent drawback of this distribution model is that information has to be processed and a decision has to be made at each hop. Advantages are that each node does not need to make a calculation based on the whole network topology, which can reduce memory and CPU requirements.

BGP [43] is an example of a protocol that works in this way. It receives potentially several paths to the same destination, selects the best one of those, and redistributes that path with its own AS number prepended.

6.1.2 Flooding

Flooding is a communication scheme in which a network element receives a message from one peer and sends the message unaltered to all its other peers unless the message has been received before.

In contrast to the previous method, no calculation has to be made in each step before propagating the messages further. This speeds up the message propagation. In practice there is a limit on the size of a flooding area because the use of CPU, memory, and link capacity. Flooding may be limited to the same hierarchy level and optimisations such as mesh groups [4] can be used.

Examples of current routing protocols that communicate through flooding are OSPF and IS-IS. Implementations of both protocols can usually propagate information to hundreds or more nodes in below a second.

6.1.3 Piggybacking

One possibility for transmitting routing-related information is to attach messages to the packets passing by. One kind of information that can be suitable to transport in this way that there is congestion in the network, see for example the ECN [40] bits in the IP header. Advantages are that it is an easy way to reach network elements further down the path and that no additional routing traffic has to be sent. The drawbacks are that the added complexity to the forwarding plane could be significant that and the message may not reach the intended recipient.

Another form of piggybacking can be said to occur when for example a routing protocol is used for transporting a type information that it was originally not intended for. Opaque LSAs in OSPF and new optional attributes in BGP are examples of ways to achieve this. Such methods can be useful when introducing a new routing protocol which requires information to be transited through network elements which during a transition phase do not yet support the new protocol.

6.2 Information Distribution Topologies

Information distribution topologies are the structures over which routing information is propagated. A topology can either be the same as the network layer topology between the network elements or be a logical topology consisting of for example TCP connections between the elements.

6.2.1 Full Mesh

The basic logical topology allowing any to any flow of information is the full mesh. Here every network element in a routing domain form peering sessions with all other network elements in the domain. An example where full mesh is used is IBGP when distributing reachability information between routers in the same autonomous system.

This method can work between a limited number of network elements. Bandwidth usage and scalability are obvious problems in large mesh topologies.

6.2.2 Hub and Spoke

Managing a full mesh of sessions quickly grows in complexity when the number of network elements involved increases. A natural way to improve scalability is by using hierarchy. This is done by letting the network elements send routing information to one (or several) control element(s). These control elements might then make certain routing or policy decisions before distributing information back to the network elements. A natural extension of this scheme is to allow a hierarchy of these control elements.

In comparison to the full mesh topology we gain in scalability but we might introduce the possibility of forming loops. If so, special measures have to be taken to avoid these loops since they might cause unnecessary load and instability to the network.

An example of such a hierarchy is route reflectors as used for internal BGP. These control elements receive routing information from members of a group of network elements, calculate routing

decisions, and distribute the result to the group. Route reflectors can also form a hierarchy and use of redundant units can ensure high availability.

Another example is route servers as used for external BGP at Internet exchange points. A router connected to the exchange point sends routing information to the route servers (typically more than one for redundancy). The route servers might then apply policy decisions on the routing information before distributing it to the other routers connected to the route servers.

6.2.3 Multicast

Much of the information relevant to a routing system is by its nature suitable for one-to-many communication. Multicast distribution using the network layer protocol or link layer protocol can be useful for some types of information. Multicast distribution at the network layer in practice requires some form of routing already working in the domain where information is transported using this method. Communication is likely to be faster than most other methods, as information does not need to be processed by each network element in the path, but acknowledgement of received information can be difficult.

As far as the authors know, there is no routing protocol implemented today where communication between network elements takes place using network layer multicast over several hops.

7. Limiting Information Distribution

There are many cases when it is desirable to limit the distribution of information in the routing system. The most obvious is scalability. Only by hiding some of the detailed information in different parts of the network from each other, can we build a routing system that will scale to the size of the Internet.

Closely related to scalability is the question of stability. On top of this, the routing information in different parts of the network must be consistent enough not to cause oscillations or persistent traffic loops. To achieve this, the amount of routing information and the speed at which it is distributed must be in tune with the network resources.

A common reason to limit information distribution is the desire of the network administrators to control the routing for different reasons (for example business agreements). This is done by formulating and implementing a routing policy.

7.1 Information Distribution Policy

In large networks, particularly where not all network elements are under the same administrative control, it is useful to have policies for what information to provide to whom. This protects internal information about a network and may contribute to enforcing routing policies. For example, correct information distribution policies can make sure that a network does not accidentally become a transit network. Policies can also be used for keeping reachability information for addresses that should only have a local scope within a limited domain.

Rules regarding which information to take into account and from where to accept information represent a related type of policy.

An example of routing policy in practice can be found in the use of BGP between autonomous systems [42].

7.2 Hiding the Details

Information that has only regional scope can be hidden from other parts of the network. In some cases it is enough not to propagate the information and in other cases we might need a new reference for the region that can be used by the rest of the network. This can be achieved by forming abstractions. Another way to hide the details of local routing information that can be used in certain cases is summarization. All these lead to better scalability properties and sometimes also aid stability. By allowing network elements to know little about the global topology and sufficiently about their own neighbourhood, the calculations that have to be performed by network elements are simplified.

Some examples of information hiding are discussed below.

7.2.1 Scoping

Protocols increasingly include mechanisms for limiting the propagation of information, as a means of implementing abstraction. In IGPs, this takes the form of link-local, area-local, and domain-wide information, as well as tags and other information to be used in implementing routing filters. At the exterior level, we have an increasing number of well-known communities (i.e., attributes of groups of routes) such as router-only (NO-ADVERTISE), AS-only (NO-EXPORT), and recently proposed scoping based on economic relationships (NOPEER) [21]

7.2.2 Creating Abstract Nodes

A global link-state protocol for the Internet would be impossible to deploy without some kind of aggregation or information hiding. One plausible solution to this would involve summarizing several (topologically close) nodes into one abstract node. Several of these abstract nodes can in turn be summarized into larger ones.

Recent results have shown that the Internet topology is getting more and more "meshy". Some argue that this fact makes it increasingly difficult to use abstractions as described here.

An autonomous system (AS) in BGP is a kind of abstract node. Another example is PNNI which has wider possibilities than IS-IS and OSPF for summarizing several nodes into one abstract node at higher levels.

7.2.3 Address Prefix Summarization

When address prefixes are hierarchically distributed, it is beneficial to summarize smaller prefixes into larger ones in order to limit routing table sizes and the number of routing updates. The situations in which this is allowed have to be specified in a routing architecture.

In BGP prefixes can in practice only be summarized explicitly by configuration, which requires the prefixes to be known beforehand. At higher levels in the routing hierarchy it should be possible to perform automatic summarization.

In future protocols, aggregation may operate on sets of abstractions more general than address ranges.

7.2.4 QoS Parameter Summarization

If routing choices are not based on shortest path but on other QoS parameters (like delay or loss) it would be preferable to summarise these parameters as well. If the routing is based on more than one parameter, e.g., both shortest path and delay, this can result in quite complex functions as the relations between the parameters need to be taken into account as well.

7.3 Authentication

Authentication can be used for avoiding intentional and to some extent unintentional misinformation in the routing system.

Session authentication is used to ensure that a communicating party is the one that it claims to be. For example, all messages can be

signed using a shared secret or with the private key in an asymmetric key pair.

In addition to session authentication, a routing system may need the capability to verify the authenticity of routing information originated by another entity than the communicating parties. A possible way to implement authentication of information about nodes and prefixes is to have an asymmetric key pair assigned to each element. The public key then has to be transmitted in a way that guarantees that it is unmodified, for example by having it signed by a trusted third party.

Authentication can make various kinds of summarization and aggregation more problematic. A well thought-out design is necessary in order to make authentication useful together with other features of the routing architecture.

Today's inter-domain routing unfortunately completely lacks authentication of the information inserted into the routing system.

7.4 Forwarding Loop Avoidance

Particularly distance vector protocols need methods for avoiding forwarding loops. This can be solved by including a path showing through which entities that a path has been propagated. BGP uses AS paths for this purpose.

Short-lived forwarding loops in topology state protocols can occur mainly because not all network elements use the same information when calculating paths. This can most of the time be avoided by making sure that information is distributed quickly and that forwarding tables are updated simultaneously.

Loops may also occur because of conflicting policies or if routers use different algorithms when making their routing decisions.

7.5 Rate Control of Information Distribution

For stability and scalability reasons it often necessary to control the rate at which routing information is distributed. This is common practice both in IGPs and in BGP-4 used in the Internet today. For example "throttling" can be applied to how often a network element generates, resends or forwards routing information.

A special case of controlling the rate of information distribution is route flap damping in BGP. Experience during the 1990s showed that in order to keep the global routing instability down, there needs to be a way to damp routing oscillations. RIPE has issued a set of

recommendations [37] for operators to use in their BGP routers. As pointed out in the recommendations document, it is important that the damping parameters are coordinated in order for routing to be consistent.

A future routing system could improve on this feature by applying damping per node or link instead of just on a per-prefix basis. The damping behaviour can also be dependent on the size or "importance" of the object to or through which the advertised path is flapping. Common practice [37] today is to avoid damping the prefixes of the root and G-TLD name servers. The possibilities for how implementing route flap damping depends on how information is propagated through the network.

8. Path Selection

The process of selecting the right path for a packet is based on the routing algorithm used and the selection criteria this algorithm is using. This section describes these concepts in more detail.

8.1 Routing Algorithms

Routing algorithms are the calculations made on routing information in order to create a routing table. The most common algorithms are topology state and distance vector.

Care may have to be taken in order to ensure that different network elements do not make contradictory decisions that cause forwarding loops or blackholing of traffic.

There are a number of more or less well known algorithms (and possibly unknown) to use for routing calculation. Some of these are briefly explained below. The list is a subset of potential algorithms that where discussed at the Midnight Sun Routing Workshop [33].

DHT (Distributed Hash Tables)
: Distributed hash tables is a common name for algorithms where for example files are associated with a key. The key is produced by hashing the file. Keys and associated files are then distributed over a number of nodes. A lookup function returns the node where the file is located when given the key as input. In [41], a number of current DHT algorithms are reviewed and some open questions are discussed. These ideas might come to use in a future routing architecture.

SPF variants
: The shortest path first algorithm, first described by Dijkstra, is

used by the two most common link state IGPs in the Internet, OSPF and integrated IS-IS.

In these protocols all participating network elements in the same area have identical link state databases where the network elements and their neighbour connections are represented.

Each network element builds a shortest path tree with itself at the root by recursively finding the next closest network element and adding this to the tree. (This is the Dijkstra algorithm.) From this tree the next hop to reach each network element in the area can easily be derived. Since all network elements perform the same calculation on the same data, the routing tables should be coherent and loop free.

Bellman-Ford
Bellman-Ford protocols can be said to use a distributed route calculation approach. An example of such a protocol is RIP. Each network element calculates a cost to other network elements it knows of and distributes this information to its neighbours. The neighbouring routers can then include this reachability in their own routing calculations and in turn distribute the resulting reachability to their neighbours etc.

Geo-based routing
TBD.

Link vector
TBD.

Other potential routing algorithm candidates are for example, Synchronous, Dual, Hot potato, Worm hole, Electric flow, Ant, Swarm, Genetic, Map abstraction, and AI.

8.2 Selection Criteria

A network element can have access to different types of information on which to base its decisions. We note that solving problems with multiple constrains might be computationally hard. Improvements are possible using heuristic methods.

Shortest Path or Lowest Cost
The main criterion when selecting a path is usually minimizing the number of hops or the sum of the assigned cost for the hops in the path. A network element can associate links with different costs depending on how preferable that link is.

Examples from today's routing can be found in OSPF [35] and IS-IS

[22] which explicitly assign costs to links. BGP has no formal support for assigning costs to inter-AS links, but it is common practice to add multiple instances of the same AS number to the path in order to increase the "cost" and thus make a path less attractive.

QoS Information

Increased demands on the network has made route decisions based on QoS parameters more relevant. The routing process may select different paths for different traffic classes. If traffic classes is defined by more than one parameter, for example both delay and packet loss, the complexity of the selection process is of course increased. For some traffic classes paths which are known to be unsatisfactory may be totally excluded when making a routing decision.

Load

The current load can be a factor in deciding which path to use[28]. Care should be taken not to create an oscillating system; if a lighter loaded link is preferred in favour of a heavily loaded one, traffic flows may move quickly between the links.

Policy

Policies for which paths to prefer may be used for increasing the control of traffic flows.

In today's Internet it is becoming increasingly obvious that the possibility to forward traffic is determined not only by the existence of a theoretically possible path, but also on whether that path is allowed to be used or not. A routing policy may state things like "traffic from network X may pass through network Y only if it is destined for network Z".

This policy information may be distributed between administrative regions and used by the local routing processes to ensure that traffic is sent on a usable path towards the receiver.

Policies are not common in connection with IGPs in today's Internet, but used at a large extent between network operators. For example, routers running BGP are often configured as to which routes to accept and which to propagate, but the protocol itself does not spread policy information (some would argue that it does). Thus the routing policy is achieved to a large extent by hiding information.

The enforcement of routing policies is discussed further in Section 12.3.4.

9. Connection Set-up

Connection set-up is used when establishing an explicit connection to send data traffic over. State for the connection may be set up in the network by signalling over the intended traffic path. Alternatively explicit state for the connection may be kept only at the connection endpoints which means that traffic will be delivered by means other than by explicit connection state in routers.

In addition to the above this section will discuss how signalling is performed, selection of path to signal along, which entity initiates the signalling, and ways to repair connections.

9.1 Initiation

Connection set-up signalling can be initiated by different entities; by a host, the first network element in the path or at the border entering a signalled domain. Said more generally it may be initiated by any host or network element in the network.

9.2 Connection state

State might be present in the endpoint entities of a connection only. One example of this is IP-in-IP encapsulation [44]. MPLS and ATM are examples where state is kept along the path between the connection endpoint entities. In cases where connection state is kept in the network elements along the path this state might be aggregated for several connections.

9.3 Explicit versus Implicit Signalling

In explicit signalling, a signalling protocol is used to reserve resources for a flow by installing state in the network elements in the path. Basic functions such as set-up and teardown of circuits can be supported. With this model the state can be either hard or soft, see Section 12.1.4.

Implicit signalling is performed when network elements look at information in the header of data packets to establish state for a flow. The state is typically soft in this model.

9.4 On-path versus Off-path Signalling

On-path signalling means that the signalling follows the data path of the flow. The advantage of this is that network elements along the data path can acquire configuration data just by receiving and forwarding the signalling messages.

In off-path signalling, on the other hand, the signalling does not follow the signalled flow. This can be the case either when signalling is not initiated by a host, or when the signalling is controlled by network entities not on the data path. Benefits of off-path signalling include a natural separation of signalling functions from forwarding functions and flexibility in signalling entity placement.

9.5 Selection of Path

The path can be selected at the originating side of the connection set-up. A source route can for example be used to describe the path. Another approach is to use a hop-by-hop scheme; the path is selected on a per hop basis and the originator just uses the destination address to indicate where the connection should be terminated. A combination of these two methods can also be used, or there can even be a central control element that makes the choice of path (e.g., a bandwidth broker).

The above schemes applies both to the connection set-up for a signalled connection as for a connection where state is only kept in the endpoint entities.

9.6 Repairing a Connection

When a link or node used by a connection becomes unavailable, a new path has to be established. The repair can be global, i.e. initiated by one or both of the connection endpoints, or it can be local, i.e. initiated by network elements close to the failure.

The global repair is relevant only if the connection is signalled. The endpoint entity can choose between having a pre-established backup path or to signal a new path when the connection loss is discovered.

Local repair can be used both when the connection is signalled and when it is not. The mechanisms to perform the repair are different in the two cases however. For a signalled connection, re-signalling is performed locally to avoid the error. Convergence in routing handles the repair for the non-signalled connections.

10. Sub-IP Capabilities

The previous section described connection set-up, which can take place either at the network layer or at the link layer in order to support upper-layer communication. This section, in contrast, deals mainly with the capabilities that the network layer may expect from lower layers.

10.1 Detection of Lost Link

Network elements need ways to determine whether the link between them is still functional. A link failure event can typically be propagated to higher layers if needed. If the path used for session communication is not the same as that used for traffic, additional complexity is added. See also Section 4.1.2.

10.2 Protection

In order to increase resistance to link faults seen by higher layers, operators may choose to let two forwarding elements be connected by more than one link and even send the same data simultaneously on both links. This increases the tolerance to link faults and the switch-over time from one link to the other can be very low or even zero.

There are different kinds of protection. The level of protection achieved depends in large on what economical resources can be afforded. Berkowitz describes different levels of protection in [5], chapter 8. The best kind of protection is often referred to as 1+1, which means that all resources are duplicated. Even 1+1+1 or higher are used for extremely high demands on protection. 1:1 are less expensive and gives the possibility to use the backup link for traffic that may by preempted if needed. Even less expensive models include 1:n and n:m, where 1 protects n and n protects m resources, respectively.

10.3 Load Sharing

In order to utilize existing infrastructure in an optimal way, there is often a desire to load share traffic over a number of paths. It can be performed over several forwarding elements or just over several links between the same forwarding elements. Instead of doing an equal load sharing between paths of the same cost, it may even be desirable for a resource owner to be able to specify the exact share of traffic that should use different paths.

The consequences of load sharing over multiple paths have to be taken into consideration in other parts of the networking architecture. Implications on multicast and debugging tools are discussed in RFC 2991 [46].

The possibilities for load sharing in the current Internet depend on the routing protocol and the network element implementation. In general it is possible to split the load equally over equal cost paths in the same IGP domain. BGP is limited to announcing different prefixes on different links. In architectures with circuit switching the problem looks a bit different, as an alternative path can be

selected in the circuit establishment phase if there is a lack of resources in one path.

10.4 Fast Reroute

In some situations it may be useful to take special measures when a neighbouring node or link becomes unavailable in order to make interruptions as short as possible. Such measures may include pre-calculated alternative forwarding and tunnels using alternative paths.

11. Administrative Interaction

11.1 Preannouncement of Unavailability

Maintenance and downtime on network elements is sometimes scheduled (be it seconds or days in advance). As the detection of a failure will always take some time, it is advantageous if there is a well designed way to inform the routing system in advance so that traffic can be routed around the network element or link in question already before the planned downtime. This eliminates interruptions during forwarding convergence. Such a scheme could be deployable for both links, networks elements and topological aggregates of larger size that are scheduled to be unavailable.

A primitive but powerful example is the "overload" bit in IS-IS [22]. As its name indicates, its original intention was that a router should be able to tell other routers that it may be too overloaded to keep its forwarding table updated and therefore it should only be used for traffic to its directly connected networks. While it seldom needs to be used as originally intended in modern routers, some operators routinely set the overload bit shortly before taking down or restarting a router.

In an intermittently connected network, such as one used for interplanetary [12] or battlefield communication similar functionality can be used for announcing that a resource will be available only for a short prescheduled period of time.

11.2 Robustness to Configuration Errors and Attacks

Misconfiguration and intentional attacks on the routing system can have a devastating effect on the whole network. Well designed authentication schemes (as discussed in Section 7.3) are of great value when ensuring that the information in the routing system is correct. Network elements and cryptographic keys can still be compromised by an attacker. It may be desired that misinformation entered into the routing system should at least only have a local

effect and not cause global problems. The task of reaching consensus even if a small number of units give misleading information is referred to as "the Byzantine Generals problem". This kind of problems has been investigated by Pearlman in [39].

The routing protocols used in the Internet today have very little protection against misinformation. Based on previous events some router vendors have added features in their BGP implementations to reset the connection with a peer if too many prefixes are announced by the other router.

11.3 Graceful Restart

Currently deployed network elements sometimes have to perform a planned or unplanned restart of the control plane. This often causes instability in the routing system and interruption of traffic forwarding. Some modern routers can keep the forwarding plane functional while restarting the control plane and extensions are now being defined and implemented for several IP routing protocols that enable routers to restart their control plane without impacting the sessions that they have with other routers. When the software is back up, the neighbours can send information about any changes that have occurred during the restart. Any new routing protocol will probably benefit from including such functionality into the design.

12. Forwarding Plane

This section covers the functionality found in the forwarding plane part of a routing architecture. In the first subsection, a number of forwarding plane concepts are outlined. This is followed by a description of different forwarding models. The section is concluded with a list of mechanisms that can be useful to implement in the forwarding plane.

12.1 Concepts

12.1.1 The Concepts of Forwarding (and Switching)

In a broad sense forwarding and switching can be defined as the ways in which routers and switches respectively process incoming data traffic in order to transport it from a source interface to one or more destination interfaces.

The exact distinction between forwarding and switching is sometimes subject for discussion. We note that the process is often referred to as switching when performed at the link layer and forwarding when performed at the network layer. Here we will refer to and use the term "forwarding" to make the discussion easier.

More specifically we define forwarding as the process of taking data from an interface, determining the outgoing interface(s) and sending the traffic there. The outgoing interface is determined using the forwarding table and information found in the header or otherwise associated with the traffic. Information in the header may be changed during forwarding.

The term "lookup" can be used when referring to the general work done by the forwarding engine when determining the outgoing interface(s).

12.1.2 Forwarding Information Storage

There is a trade-off between the amount of forwarding information to store in the headers and in the forwarding table located in the forwarding element.

One possible scheme would be to let each packet be forwarded according to a number of predetermined hops, e.g. the next outgoing interface for each hop would be stored in a list in the header. This would require minimal amount of state in the forwarding elements but the possibility to adapt to changes in the underlying topology would not be a local problem but a global one. The scheme in its purest form would also be difficult to scale for large networks mainly because all network elements would have to know the state of all links in the world. The model described in this paragraph is often referred to as "strict source routing".

Another alternative is to put only a destination address in the header and let each network element make the forwarding decision based on the address and information in the forwarding table. This conserves the header space required, but more memory and processing power is needed in the network elements. This model is often referred to as hop-by-hop routing.

12.1.3 Destination Address Versus Flow/Circuit Identifier

When determining the outgoing interface based on the destination addresses, the network element typically has to perform a longest prefix match search. This process is relatively expensive to implement in hardware compared to using flow or circuit identifiers.

Flow/circuit identifiers can be globally or locally assigned in a network element. It is relatively easy to perform a lookup to find the outgoing interface and flow/circuit identifier based on the incoming flow/circuit identifier. The drawback here is that some type of signalling is needed to set up these lookup tables. Also, if each identifier is associated with a flow, there may be scalability problems when there are a lot of flows.

12.1.4 Soft Versus Hard Lookup State

Lookup state in a network element can be classified either as hard or soft. The general notion of soft and hard state has been discussed by Chiappa in a short technical note [14].

Hard state is here defined to mean that it must be removed explicitly by a signalling message.

Soft state is defined as a state that must be refreshed by some means to continue to exist. For example, a timer could be associated with the state which is removed when the timer expires. What resets the timer could for example be the traffic flow or some signalling refresh message.

12.2 Forwarding Models

In this section we give an overview of different models which network elements can use to handle data transport.

12.2.1 Hop-by-Hop

In hop-by-hop forwarding, each router uses a forwarding table for determining how packets to different destinations should be forwarded. Each router has to examine every packet header. Router implementers are however free to optimise forwarding decisions by caching results from previous lookups. Load sharing and forwarding based on other fields than the destination address is possible, but not in a very coordinated way between routers.

Hop-by-hop forwarding is the predominant forwarding technique used in the Internet so far.

12.2.2 Packet Carried Explicit Route

The path through the network is determined as the packet enters the area where packet carried explicit routes are used. This can be done either by the source host or by a network element. This information is inserted into the header of the packet and the following network elements use the path information for forwarding. Packet carried explicit route is also known as source routing.

Strict source routing means that the packet should take exactly the specified path, while loose source routing means that the packet may also traverse other forwarding elements between two addresses in the list.

This model is theoretically possible in IPv4, but for practical and

security reasons it is not used much.

12.2.3 Signalled Explicit Route

In a signalled explicit path, a connection has to be set up before any communication can take place. Network elements in the path have to store some state regarding how to forward packets in that particular flow.

The connection set-up may be initiated manually, by the source, or be driven by the traffic received by routers.

Examples of protocols using this technique today are MPLS and ATM.

12.2.4 Flow Based Forwarding

Flow labels can be used for identifying a number of packets that should be processed equally by the network. Routers may use the flow information in order to optimise their forwarding. All the information required for forwarding should also be available from the packet header without looking at the flow label. The flow label can be set by the source host or by a router in the network. In some architectures it may also be modified at the border between areas.

The use of flows is somewhat specified in IPv6 [10].

12.2.5 Combining the Different Models

Different forwarding models could be used in different parts of the network. In a hierarchy with two levels, for example, the top level could use the packet carried explicit route model while different sub segments may use any model they want. This of course has to be coordinated at the control plane level.

12.3 Forwarding Mechanisms

Some mechanisms for increased availability and resource utilization have to be implemented in the forwarding element.

12.3.1 Load Sharing

Issues in the control plane regarding load sharing over several links or paths was discussed in Section 10.3. When the control plane has informed the forwarding plane of several alternative paths, there has to be an algorithm for selecting which path to use. In a circuit-switched network this is typically done during connection set-up for each flow. Even though the current Internet architecture does not guarantee that reordering will not occur, it is avoided as it has a

negative effect on TCP traffic. Because of this, naive approaches such as random or round-robin selection cause problems, see RFC 2991 [46]. In the absence of a flow label in IPv4, a common practice is to load balance based on a hash calculated from values such as source address, destination address, source port, and destination port.

12.3.2 Simultaneous Forwarding over Multiple Paths

For some applications where loss of communication has a devastating effect (telemedicine comes to mind), intermediate networks may not be completely trusted despite any QoS guarantees. One solution for making sure that packets still reach their destination without interruptions is to send the same information through two or more paths that are guaranteed to be completely separate.

12.3.3 Fast Reroute

Functionality for fast reroute (see Section 10.4) needs support in the forwarding plane for optimized performance. This includes detecting failures momentarily and for example the ability to forward traffic through a tunnel whose next hop can be changed quickly.

12.3.4 Policy Enforcement

There often needs to be local policies for which traffic to forward and which to drop. This information may be collected from the routing protocol or locally configured.

One example of this is ingress filtering [17] which is often used in order to render it impossible for senders to spoof their source address.

Devices whose main purpose is forwarding policy enforcement are usually referred to as firewalls. General-purpose forwarding elements also usually have some basic firewalling mechanisms, such as filtering based on for example source address and port number.

13. Acknowledgements

The authors would like to thank (in alphabetical order) Niklas Borg, Elwyn Davies, Dmitri Krioukov, Per Lindberg, Olle Pers, Henrik Villfor, and Kristofer Warell for reviewing this material and giving us valuable feedback.

References

[1] Alaettinoglu, C., Villamizar, C., Gerich, E., Kessens, D., Meyer, D., Bates, T., Karrenberg, D. and M. Terpstra, "Routing

Policy Specification Language (RPSL)", RFC 2622, June 1999.

[2] Antonov, V., "Trivial Routing Architecture Proposal (TRAP)", September 1995.

[3] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M. and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[4] Balay, R., Katz, D. and J. Parker, "IS-IS Mesh Groups", RFC 2973, October 2000.

[5] Berkowitz, H., "Building Service Provider Networks", John Wiley & Sons , ISBN 0-471-09922-8, 2002.

[6] Berkowitz, H., "Router Renumbering Guide", RFC 2072, January 1997.

[7] Berkowitz, H., Hares, S., Retana, A., Krishnaswamy, P., Lepp, M. and E. Davies, "Terminology for Benchmarking BGP Device Convergence in the Control Plane", draft-ietf-bmwg-conterm-03 (work in progress), July 2002.

[8] Borg, N., Holmberg, R., Fuzesi, P. and K. Nemeth, "NAIS - Network Architecture for Inter-Domain Services", 10th International Telecommunication Network Strategy and Planning Symposium (Networks 2002) Munch, Germany, June 2002.

[9] Braden, B., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[10] Carpenter, B., Conta, A., Deering, S. and J. Rajahalme, "IPv6 Flow Label Specification", draft-ietf-ipv6-flow-label-03 (work in progress), September 2002.

[11] Castineyra, I., Chiappa, N. and M. Steenstrup, "The Nimrod Routing Architecture", RFC 1992, August 1996.

[12] Cerf, V., "Delay-Tolerant Network Architecture: The Evolving Interplanetary Internet", draft-irtf-ipnrg-arch-01 (work in progress), August 2002.

[13] Chiappa, N., "Endpoints and Endpoint Names: A Proposed Enhancement to the Internet Architecture", 1999, <http://users.exis.net/~jnc/tech/endpoints.txt>.

[14] Chiappa, N., "'Soft' and 'Hard' State", <http://users.exis.net/

~jnc/tech/hard_soft.html>.

[15] Doria, A., "Future Domain Routing Requirements Group B contribution", draft-irtf-routing-reqs-groupb-00 (work in progress), February 2002.

[16] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[17] Ferguson, P. and D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing", BCP 38, RFC 2827, May 2000.

[18] IEEE, "Guidelines for use of a 48-bit Global Identifier (EUI-48)", October 2002, <http://standards.ieee.org/regauth/oui/tutorials/EUI48.html>.

[19] Hain, T., "Application and Use of the IPv6 Provider-Independent Global Unicast Address Format", draft-hain-ipv6-pi-addr-use-03 (work in progress), October 2002.

[20] Hubbard, K., Kosters, M., Conrad, D., Karrenberg, D. and J. Postel, "INTERNET REGISTRY IP ALLOCATION GUIDELINES", BCP 12, RFC 2050, November 1996.

[21] Huston, G., "NOPEER community for BGP route scope control", draft-ietf-ptomaine-nopeer-00 (work in progress), April 2002.

[22] International Organization for Standardization, "Intermediate system to intermediate system intra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode Network Service (ISO 8473)", ISO Standard 10589, 1992.

[23] ITU, "The International Public Telecommunication Numbering Plan", ITU Recommendation E.164, May 1997.

[24] APNIC, ARIN, RIPE NCC, "IPv6 Address Allocation and Assignment Policy", RIPE 246, June 2002.

[25] Juniper Networks, "Junos(tm) Internet Software Configuration Guide Routing and Routing Protocols, Release 4.2", September 2000, <http://www.juniper.net/techpubs/software/junos42/swconfig-routing42/html/glossary.html#1013039>.

[26] Kastenholz, F., "Requirements For a Next Generation Routing and Addressing Architecture", draft-irtf-routing-reqs-groupa-00 (work in progress), April 2002.

[27] Katz, D., Kompella, K. and D. Yeung, "Traffic Engineering Extensions to OSPF Version 2", draft-katz-yeung-ospf-traffic-09 (work in progress), October 2002.

[28] Khanna, A. and J. Zinky, "The Revised ARPANET Routing Metric", In Proceedings of ACM SIGCOMM , September 1989.

[29] Khosravi, H. and T. Anderson, "Requirements for Separation of IP Control and Forwarding", draft-ietf-forces-requirements-06 (work in progress), July 2002.

[30] Lear, E., "What's In A Name:Thoughts from the NSRG", draft-irtf-nsrg-report-06 (work in progress), August 2002.

[31] Li, T. and H. Smit, "IS-IS extensions for traffic engineering", draft-ietf-isis-traffic-04 (work in progress), August 2001.

[32] McPherson, D., "Intermediate System to Intermediate System (IS-IS) Transient Blackhole Avoidance", RFC 3277, April 2002.

[33] "Midnight Sun Routing Workshop", June 2002, <http://www.cdt.luth.se/babylon/msrw/>.

[34] Mouly, M. and M. Pautet, "The GSM System for Mobile Communications", ISBN 0945592159, 1992.

[35] Moy, J., "OSPF Version 2", STD 54, RFC 2328, April 1998.

[36] International Organization for Standardization, "Information Processing Systems - Data Communications - Network Service Definition", ISO/IEC Standard 8348, September 1996.

[37] Panigl, C., Schmitz, J., Smith, P. and C. Vistoli, "RIPE Routing-WG Recommendations for Coordinated Route-flap Damping Parameters", RIPE 229, October 2001.

[38] Papadimitriou, D., "Inference of Shared Risk Link Groups", draft-many-inference-srlg-02 (work in progress), November 2001.

[39] Perlman, R., "Network Layer Protocols with Byzantine Robustness", PhD Thesis, Department of EECS, MIT, August 1988.

[40] Ramakrishnan, K., Floyd, S. and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[41] Ratnasamy, S., Schenker, S. and I. Stoica, "Routing Algorithms for DHTs: Some Open Questions", 1st International Workshop on

Peer-to-Peer Systems (IPTPS '02) , March 2002.

[42] Rekhter, Y. and P. Gross, "Application of the Border Gateway Protocol in the Internet", RFC 1772, March 1995.

[43] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[44] Simpson, W., "IP in IP Tunneling", RFC 1853, October 1995.

[45] Thaler, D., Handley, M. and D. Estrin, "The Internet Multicast Address Allocation Architecture", RFC 2908, September 2000.

[46] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, November 2000.

[47] Thomson, S. and T. Narten, "IPv6 Stateless Address Autoconfiguration", RFC 2462, December 1998.

[48] Villamizar, C., Alaettinoglu, C., Govindan, R. and D. Meyer, "Routing Policy System Replication", RFC 2769, February 2000.

Authors' Addresses

Howard Berkowitz
Gett Communications
5012 S. 25th St
Arlington, VA 22206
USA

Phone: +1 703 998-5819
Fax: +1 703 998-5058
EMail: hcb@gettcomm.com

Erik Aman
Telia Research AB
SE-123 86 Farsta
Sweden

Phone: +46 8 713 81 71
EMail: Erik.G.Aman@telia.se

Thomas Eriksson
Telia Research AB
SE-123 86 Farsta
Sweden

Phone: +46 8 713 81 20
EMail: Thomas.A.Eriksson@telia.se

Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.