RTP media failover: problem statement
draft-taylor-mmusic-rtp-failover-problem-01.txt

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

This Internet-Draft will expire on February 31, 2009.

Copyright Notice

Copyright (c) 2016 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.
Abstract

Network-based functions that terminate large numbers of RTP media streams and that offer high availability, such as session border controllers or conference bridges, typically preserve the same IP address towards sources of RTP media across a failover event because it is impractical to signal a change of IP address towards large numbers of RTP sources sufficiently rapidly to keep media interruption intervals within acceptable limits. The need to preserve the IP address of RTP media terminating functions across a failover event imposes architectural requirements that can be difficult or costly to meet, particularly in network function virtualization environments. This document describes the problem, outlines the key requirements for a solution, and discusses the merits and shortcomings of various existing approaches to solving the problem, before arguing that a new solution is needed.

Table of Contents

1. Introduction...................................................3
2. Problem Space..................................................4
   2.1. Geographic Redundancy....................................4
   2.2. Resource Efficiency.......................................5
   2.3. Evolution to Cloud-Centric Virtualized Network Functions..6
   2.4. Absence of Layer 2 Connectivity........................6
3. Requirements for Improved Failover of RTP Media Streams........7
   3.1. Upper Limit on Media Interruption Time....................7
   3.2. Geographic Redundancy....................................8
   3.3. Resource Efficiency.......................................8
   3.4. Network Compatibility....................................8
   3.5. Backwards Compatibility...................................8
   3.6. Compatibility with Hosted NAT Traversal...................8
4. Available Solutions and Their Limitations.......................9
   4.1. Use of SIP Re-INVITE or UPDATE to Update SDP..............9
   4.2. Restriction of Size of Fault Zone........................10
   4.3. Re-Routing at the IP Layer Using BGP....................10
   4.4. Re-routing at the IP Layer Using Link-State Protocols....11
   4.5. Anycast..................................................12
   4.6. RTP Proxy / Load Balancer................................14
   4.7. Multipath RTP............................................14
5. Proposed New Approach to RTP Media Failover....................15
6. References.....................................................15
   6.1. Normative References....................................15
   6.2. Informative References...................................16
7. Change Log.....................................................17
   7.1. Changes in draft-taylor-mmusic-rtp-failover-problem-01...17
1. Introduction

Session Description Protocol (SDP) [RFC4566], typically conveyed via Session Initiation Protocol (SIP) [RFC 3261] requests, provides a means for Real Time Protocol (RTP) [RFC3550] endpoints to negotiate via the Offer/Answer Model described in RFC3264 [RFC3264] the details of media sessions to be established between them. An endpoint conveys the specific IP address and port number on which it wishes to receive a given media stream via the c= (connection information) and m= (media description) lines defined by SDP. An endpoint that wishes to change the IP address and port number on which it is to receive a given media stream needs to send updated SDP to the transmitter of that media stream.

Some services that make use of SIP and SDP to negotiate the establishment of media sessions for voice, video or real-time streaming purposes employ RTP media relay functions in the network, for example associated with a SIP back-to-back user agent in the form of a session border controller. A single such RTP media relay instance may support the relaying of tens of thousands of concurrent media streams. Likewise, a large-scale conference bridge may support many thousands of concurrent RTP sessions.

With network functions that terminate such large numbers of RTP sessions (referred to in the remainder of this document as "RTP-terminating network functions"), it is desirable to provide some means to protect against hardware or software failures in a manner that preserves the RTP sessions, such that failover can be accomplished with minimal transient impairment to the audio or video streams as perceived by users of the service. This may be accomplished by deploying a second, identical instance of the network function to act as a backup. The two instances work together as a pair, with one instance actively performing RTP session termination and the other instance standing by, ready to take over if the active instance fails. Some means is provided to enable the backup instance to detect failure of the active instance, for example by means of a heartbeat protocol between the two instances. On detecting failure of the active instance, the backup instance becomes active and can take over the processing of all the media streams that were previously handled by the active instance.

The handover between active and standby network function instances is typically handled in a manner that is transparent to the RTP endpoints that are currently sending media to the active instance. This may be accomplished by assigning a virtual IP address that is shared between the active and standby instances of the network.
function. It is this IP address that is conveyed over SDP to the set of RTP endpoints served by the network function as the destination to which they should send their media streams. Under normal operating conditions, the virtual IP address is associated with the currently active member of the RTP-terminating network function pair. When the standby member of the network function pair detects a failure of the active member, it becomes active and claims the virtual IP address, for example by issuing a gratuitous Address Resolution Protocol (ARP) message. By this means, all the media streams that are currently being transmitted to the formerly active member of the network function pair may be re-directed to the newly active member without any of the transmitting endpoints being aware of the change.

Fault tolerance schemes that take advantage of IP address swapping in the manner described above are widely employed by network functions that terminate large numbers of RTP streams and are often embodied in physical appliances such as session border controllers.

2. Problem Space

The following points describe problematic aspects of highly available network functions that terminate large numbers of RTP media streams, for which an improved solution (or solutions) is sought. A common theme among these problems is the fact that failure recovery of the network function needs to be transparent to the sources of the RTP media streams handled by a failed RTP-terminating network function instance, in the sense that such sources are not aware that failover of the RTP-terminating network function instance serving them has taken place, except to the extent that they may experience some momentary interruption of received media. In particular, RTP endpoints continue to send media to the same IP address before and after an RTP-terminating network function failover event.

2.1. Geographic Redundancy

A pair of RTP-terminating network function instances deployed in the same physical location in active-standby mode and sharing the same virtual IP address can provide protection against equipment failure such as the failure of the active instance itself or the failure of network connectivity to the active instance. However, this arrangement does not protect against failure of the site at which the RTP-terminating network function is deployed, or failure of network connectivity to the site as a whole.

Network operators typically protect against site failure or site connectivity failure by implementing some form of geographic
redundancy. This usually involves replicating the equipment needed to support a given service on at least two sites, such that in the event of the failure of one site, the service can continue to be supported by making use of the equipment on one of the other sites. Since redundant equipment is deployed within each site to protect against equipment failure, protection against site failure requires yet more equipment to be deployed, which has obvious cost implications. Note that site failure is considered to be a far less frequent event than equipment failure, and typically no effort is made to preserve active real-time media sessions across a site failover, unlike the case of equipment failover.

Network operators can potentially reduce the cost of meeting service availability targets by protecting against both equipment failure and site failure with a single common failure recovery mechanism. For example, a pair of RTP-terminating network function instances could be deployed with one member of the pair located in one site and the other member located in another site. If the network operator determines that real-time media sessions must be preserved across equipment failure, then we need to be able to switch all of the media streams addressed to the failed RTP-terminating network function instance to the standby instance located in the backup site sufficiently quickly that users experience no more than brief transient interruption to their incoming media streams.

This can be accomplished quite efficiently at Layer 2 (by swapping a virtual IP address from one member of the network function pair to the other), but this approach requires the establishment of a Layer 2 connection between sites, which can be complex and inconvenient to accomplish. Other methods for preserving real-time media streams across geographic failover are discussed below in Section 4.

2.2. Resource Efficiency

While it is common practice to deploy RTP-terminating network functions as active-standby pairs to provide high availability, this arrangement is relatively wasteful of hardware resources because, at any one time, only half the hardware supporting the RTP-terminating network function is doing useful work. The cost of hardware to support RTP processing can be relatively high, particularly if the function is required to perform compute-intensive work on media streams such as encryption/decryption of Secure Real Time Protocol (SRTP) [RFC3711], or audio or video transcoding.

The amount of hardware resources required to support any given capacity of RTP-terminating network function could be very considerably reduced if it were possible to provide protection against hardware or software failure by means of a pooling
arrangement. This could be in the form of a group of RTP-terminating network function instances, all of which are active all of the time, and where their total aggregate capacity exceeds the maximum expected load by a sufficient margin that the load carried by any given instance can be successfully load-balanced across the remaining instances in the event of the failure of this instance. An alternative approach would be to deploy a small number of standby instances to protect a much larger number of active instances, and to switch all of the RTP sessions carried by a failed active instance over to one of the standby instances.

This type of high availability scheme is often known as N+k redundancy. While the latter example above of N+k redundancy (N x active, k x standby) is compatible with the swapping of virtual IP addresses, the former example (active-active load-balanced) is not. Most network operators express a strong preference for active-active N+k schemes, regardless of any consideration as to whether active-active N+k can actually be shown to deliver higher availability than active-standby N+k.

2.3. Evolution to Cloud-Centric Virtualized Network Functions

Many network operators are embracing network functions virtualization (NFV), whereby network functions that would previously have been embodied as physical appliances are now embodied as software components deployed in a virtualized cloud computing environment. With the move to NFV, network operators are expressing a strong preference for cloud-centric approaches to network function design. This tends to imply the deployment of relatively large numbers of relatively small instances of network functions, where all instances are active, and protection against failures at any level from individual instance through physical host up to a complete site is provided by means of active-active N+k redundant pools of virtualized network function instances.

It is difficult in practice to architect highly available solutions for RTP-terminating network functions based on active-active N+k redundancy that meet the requirement that failover must be transparent to sources of RTP media. Possible solutions and their limitations are discussed later in this document.

2.4. Absence of Layer 2 Connectivity

Widely used active-standby techniques for RTP-terminating network functions that involve the sharing and swapping of a virtual IP address typically require that the active and standby members in a high availability arrangement are directly connected via a Layer 2 network segment.
As discussed in section 2.1 above, this can be problematic if the active and standby RTP-terminating network function instances are located in different geographic sites, although this problem is soluble, for example with the aid of a Layer 2 Virtual Private Network.

A more intractable problem arises when a network operator chooses to design a network functions virtualization infrastructure with a Layer 3 centric fabric that does not provide L2 connectivity between virtualized workloads. While this is not yet a common approach to cloud network design, scaling issues with L2-centric fabrics are expected to drive increasing popularity of L3-centric approaches in the future. In L3-centric cloud network fabrics, failover of RTP-terminating network functions based on virtual IP address swapping cannot be supported with the usual approach based on gratuitous ARP [RFC826].

Approaches based on Network Address Translation (NAT) [RFC3022] such as OpenStack’s Floating IP Address concept could potentially address this need, but the insertion of additional network elements into the RTP path to perform NAT introduces additional failure scenarios that need to be protected against. Also, such approaches require that the infrastructure management plane is capable of responding very quickly to a NAT re-configuration request, such that the interruption in incoming media streams experienced by users is perceived as no more than momentary. Practical experience suggests that this cannot currently be achieved with real-world cloud infrastructure solutions.

3. Requirements for Improved Failover of RTP Media Streams

For the reasons described in section 2 above, it is considered desirable to specify new behaviors of RTP endpoints so as to provide an improved method for failover of RTP media streams that supports high availability of RTP-terminating functions in the network.

When considering any new solution for failing over large numbers of RTP media streams, the following requirements should be met.

3.1. Upper Limit on Media Interruption Time

A new solution designed to preserve RTP media in the face of failure of an RTP-terminating network function instance MUST successfully re-establish a viable RTP media path for each and every flow that was previously handled by the failed instance within a maximum
elapsed time of two seconds, and SHOULD re-establish all media flows within 500 milliseconds.

3.2. Geographic Redundancy

A new solution for failover of RTP media streams MUST be capable of preserving media sessions across the failure of a physical site or the failure of network connectivity to a physical site, even when the two sites are separated by hundreds of miles.

3.3. Resource Efficiency

A new solution for failover of RTP media streams MUST support N+k redundancy of RTP-terminating network functions, where k << N.

3.4. Network Compatibility

A new solution for failover of RTP media streams MUST not assume the existence of Layer 2 connectivity between RTP-terminating network function instances that are protecting each other, and MUST not assume the existence of any network capabilities beyond basic IP unicast connectivity.

3.5. Backwards Compatibility

It will take time to upgrade the installed base of RTP endpoints to embody any new behaviors required to support a new solution for RTP media failover. RTP-terminating network functions that embody a new solution for failover of RTP streams MUST remain compatible with RTP endpoints that do not support the new behaviors. RTP-terminating network functions that support a new solution for failover of RTP media streams MAY continue to support legacy methods for failover of RTP media streams, but are not required to do so.

3.6. Compatibility with Hosted NAT Traversal

A new solution for failover of RTP media streams MUST be compatible with the method of Hosted NAT Traversal described in RFC7362 [RFC7362]. If the solution requires that, following failover, the RTP endpoint is to transmit RTP media streams to an RTP-terminating network function at an IP address and port number that is different than prior to failover, the RTP endpoint MUST commence transmission of RTP packets towards the new IP address and port number without waiting to receive RTP media packets from the new IP address and port number.
4. Available Solutions and Their Limitations

In this section, we discuss alternative ways of supporting high availability of RTP-terminating network functions without any change to the existing behavior of SIP- and SDP-signaled RTP endpoints. It will be seen that none of these methods meets the full set of requirements identified in Section 3 above.

4.1. Use of SIP Re-INVITE or UPDATE to Update SDP

A SIP User Agent in an active session state associated with a currently active RTP transmitter can be instructed to transmit RTP to a different destination IP address and port number by sending it an in-dialog re-INVITE or UPDATE request that includes SDP with the new connection details.

This use of a re-INVITE or UPDATE request to update SDP within an active session may be leveraged to manage failover of an RTP-terminating network function instance in the network. The SIP User Agent instance that is associated with the RTP-terminating network function instance, upon detecting the failure of said instance, could send a re-INVITE or UPDATE request to each and every SIP UA that is in an active session and sending RTP media to the failed RTP-terminating network function instance, with an SDP body that directs each RTP endpoint to send RTP media to a different RTP-terminating network function instance.

In practice, it is found that the processing resources required to transmit the required number of re-INVITE or UPDATE requests and process all of the responses so as to achieve resumption of all active RTP media flows within an acceptable elapsed time far exceed the processing resources that would normally be required to support the SIP signaling load associated with that number of concurrent sessions. It is therefore very costly to support RTP media failover by means of this technique.

One use case for RTP-terminating network functions is in peering arrangements for the connection of large numbers of concurrent RTP sessions between different networks. In this situation, if a SIP UA associated with an RTP-terminating network function were to send large numbers of in-dialog re-INVITE or UPDATE requests in a short elapsed time to its peer SIP UA in the other network so as to request that a large number of incoming RTP streams be sent to a different IP address and port number, the receiving SIP UA might easily be overwhelmed by the incoming load of SIP message traffic. This could have the doubly deleterious effect of failing to achieve the failover of many of the RTP streams in a timely fashion, and
failing to complete requests for the establishment of new sessions while the signalling overload condition persists.

4.2. Restriction of Size of Fault Zone

In a network functions virtualization environment, it is possible to terminate large numbers of RTP sessions by deploying large numbers of small scale RTP-terminating network function instances. These instances could be deployed without any form of redundancy, such that the failure of any instance causes the complete loss of all RTP media sessions currently being handled by it.

With this type of arrangement it could be argued that, if the maximum number of sessions that are handled by a single RTP-terminating network function instance is low enough, then the failure of one instance and the consequent loss of all the media sessions that it is currently handling represents a relatively minor impact to the service as a whole.

Some network operators may take the view that this approach meets their criteria for an acceptable quality of service. However it should be pointed out that, with a reasonably efficient implementation of the RTP-terminating function, a minimally-sized instance occupying just a single virtual CPU could be handling several hundred concurrent sessions. For most network operators, the loss of several hundred concurrent media sessions arising from the failure of an unprotected network element would be unacceptable.

It is also worth pointing out that deploying large numbers of small instances of a network function may restrict the size of the fault zone as it relates to failure of small-scale resources such as virtual machines, hypervisors or compute nodes, but it does not restrict the size of the fault zone as it relates to failure of large-scale resources such as an availability zone, an entire cloud instance or an entire site. Protection is still required in the event of these resources failing.

4.3. Re-Routing at the IP Layer Using BGP

It is possible to cause IP packets to be delivered to a different host system by means of appropriate interaction with the routing protocols of the IP network control plane. This capability can be exploited to support a highly available RTP-terminating network function.

In an IP network that employs Internal Border Gateway Protocol (BGP) [RFC4271], one way to accomplish this is to add a BGP speaker function to the RTP-terminating network function. The RTP-
terminating network function uses BGP to advertise a route to the RTP service address via its own host address. The IP infrastructure to which the RTP-terminating network function instance is connected effectively treats the host address of this instance as the next hop towards the RTP service address, and routes IP packets addressed to the RTP service address towards that RTP-terminating network function instance.

In the event of the failure of such an RTP-terminating network function instance, another RTP-terminating network function instance that is providing protection for the failed instance issues a BGP message that withdraws the original RTP service route via the host address of the failed instance, and advertises a new route via its own host address. The IP infrastructure will now route all IP packets addressed to the RTP service address towards the protecting RTP-terminating network function instance.

This approach places a number of demands on the IP routing infrastructure to which the active and standby RTP-terminating network function instances are connected which it may be difficult to meet in practice. In particular, the routing infrastructure must be able to respond to the withdrawal of a route and the advertisement of a new route to the RTP service address sufficiently rapidly to meet the requirement described in Section 3.1 on the upper limit for media interruption time.

It also requires that the routing policy prevailing in the infrastructure allows for individual host routes (e.g. IPv4 /32 or IPv6 /128 routes) to be installed in routing tables.

In many cases it may not be practicable or even possible to meet these demands.

4.4. Re-routing at the IP Layer Using Link-State Protocols

In IP networks that employ Interior Gateway Protocols other than IBGP, for example OSPF [RFC2328] or IS-IS [RFC1142], it may be possible to re-route RTP media at the IP layer using methods conceptually similar to that described in section 4.2. However, link-state protocols rely on the detection of a link failure to initiate re-routing of IP traffic, and it isn’t likely that the failure of an RTP-terminating network function instance could always be detected as a link failure by neighboring routers sufficiently quickly to meet the requirement on the upper limit for media interruption time described in section 3.1.
4.5. Anycast

Anycast [RFC4786] is a routing scheme whereby multiple host systems share a single address, and IP packets destined for that address are routed to the host that is "nearest" the sender.

Anycast techniques can be employed to implement a scheme that is conceptually similar to that described in Section 4.2 above, but which relies on the active and standby members of an RTP-terminating network function pair to advertise different route weights such that IP traffic is routed to the active member. Failover requires that the advertised route weights are adjusted to ensure that IP traffic is routed to the standby member.

Anycast techniques can also be employed to support a form of load-balancing. If multiple RTP-terminating network function instances are advertised to be reachable at the same address and with equal distance, the IP routing infrastructure can distribute load across the instances using Equal Cost Multi Path (ECMP) routing. Furthermore, if some means is provided for the detection of the failure of any given RTP-terminating network function instance and subsequent transmission of a BGP message withdrawing the route to that instance, then ECMP should act to re-distribute the load across the remaining instances.

This use of Anycast appears to address the N+k active-active use case very effectively, although it should be noted that, in the case of an RTP-terminating network function that is acting as a media relay, for example as a component of a session border controller, it is not generally possible to ensure that the two streams that make up a bi-directional RTP session are handled by the same media relay function instance. This may well add considerably to the complexity of the design of the media relay function.

A more serious problem with using Anycast in this way is that, in a virtualized environment, it becomes extremely challenging to manage the placement of the RTP-terminating network function instances. These challenges arise because, at each router supporting ECMP that sees multiple available routes to the Anycast address with the same distance, the router splits the traffic evenly between all these routes. If there is more than one router between the source of the traffic and the set of RTP-terminating network function instances that are the destination of the traffic, these instances must be arranged so as to create a symmetrical routing tree in order to ensure that each instance receives a similar share of the overall traffic load.
To illustrate this, consider the following scenario, described in the diagram below. All RTP media traffic from a given set of RTP endpoints transits via Router A (which might be, for example, an end-of-rack L3 switch), and then via either Router B or Router C (which might be, for example, top-of-rack L3 switches) to RTP-terminating network function instances M1 through M5. The routes to instances M1 and M2 are via Router B, while the routes to instances M3, M4 and M5 are via Router C. All RTP-terminating network function instances are advertising the same RTP service address.

From the point of view of Router A, there are two possible routes to the RTP service address, via Router B and Router C respectively. It therefore sends half of the RTP flows to Router B, and half to Router C. Router B will distribute half of the RTP flows that it receives from Router A to each of M1 and M2, while Router C will distribute one third of the flows it receives from Router A to each of M3, M4 and M5. It can be seen that the load is not evenly balanced over the population of RTP-terminating network function instances.

In the general case, placing the instances of RTP-terminating network functions so as to form a symmetrical routing tree presents an extremely difficult problem for the workload scheduling algorithm in a virtualized environment, particularly if the intention is to spread the load between RTP-terminating network function instances.
on two or more separate sites. Topology-aware scheduling is not a capability offered by current generations of cloud orchestration software, and even if it were, dynamically scaling the population of RTP-terminating network function instances while maintaining a symmetric routing tree would be cumbersome and inflexible.

4.6. RTP Proxy / Load Balancer

It is possible to imagine a solution based on an RTP proxy or load balancer which sits between RTP-terminating network functions and a population of RTP endpoints that are sending RTP media towards those RTP-terminating network functions. The RTP proxy or load balancer presents a single IP address towards the population of SIP UAs. In the event that an instance of an RTP-terminating network function fails, the RTP proxy or load balancer can detect the failure of the instance, and re-direct incoming RTP media to a different instance of an RTP-terminating network function which has been configured so as to receive and correctly process the incoming RTP media streams that were previously being sent to the failed instance.

The problem with this approach is that the RTP proxy or load balancer itself represents a single point of failure that must be protected by some means in order to provide a high availability service. All that is achieved in deploying an RTP proxy or load balancer is that the RTP failover problem is moved from the RTP-terminating network functions to an RTP-proxying function. The fundamental problem remains the same: a population of RTP endpoints expects to be able to transmit RTP media streams to the IP address and port number that was negotiated when the session was set up, and this address must be preserved across a failover of the RTP proxy or load balancer in order to ensure session continuity.

4.7. Multipath RTP

Multipath RTP [I-D.ietf-avtcore-mprtp] (MPRTP) is a proposed extension to RTP which splits a single RTP stream into multiple subflows that are transmitted over different network paths. It is primarily intended to leverage pooling of the resource capacity of multiple network paths to improve user experience by enabling higher bit-rate and higher quality codecs to be used.

It is possible to imagine using MPRTP to support failover of individual RTP streams, by defining two MPRTP sub-flows at session establishment time and then sending all media over one of the sub-flows. If an RTP-terminating network function involved in such an MPRTP session were to fail, media could then be transmitted and received via the other sub-flow.
There are a number of concerns about the use of MPRTP to support the simple case of failover. MPRTP is primarily concerned with the support of multiple simultaneous sub-flows that must be merged by the receiver. This needs additional RTP header information which would require extensive enhancements to the RTP stack in each endpoint. This additional RTP header information would not be required for the simple failover case. Furthermore, MPRTP mandates that endpoints keep alive sub-flows on which no media is being sent. This would result in the unnecessary consumption of resources in RTP-terminating network functions. Finally, MPRTP does not support any mechanism for signaling to a transmitting RTP endpoint that it should stop sending media on one sub-flow and start sending it on another. Thus any solution for RTP failover based on the use of MPRTP would require further protocol extensions to address this requirement.

5. Proposed New Approach to RTP Media Failover

This document has argued that currently available solutions for RTP media failover are inadequate because they are inefficient from a hardware resources standpoint and not well suited to the evolving environment of network functions virtualization. It has also pointed out that many of the challenges faced by RTP media failover solutions arise from the need to preserve the destination IP address of the RTP-terminating network function across a failover event.

The need for robust and flexible high availability solutions for SIP User Agents is addressed by existing standards by permitting SIP UAs to establish multiple flows over which SIP signaling messages can be sent and received [RFC5626].

This document proposes that an analogous scheme be defined for RTP endpoints. The details of such a proposed scheme will be described in another Internet Draft.

6. References

6.1. Normative References


6.2. Informative References

7. Change Log


Corrected missing section header "Re-Routing at the IP Layer Using BGP"

Added new section 4.7 on MPRTP

Authors’ Addresses

Martin Taylor
Metaswitch Networks
100 Church St
Enfield EN2 6BQ
UK

Email: martin.taylor@metaswitch.com

Nic Larkin
Metaswitch Networks
100 Church St
Enfield EN2 6BQ
UK

Email: nic.larkin@metaswitch.com