

BIER
Internet-Draft
Intended status: Standards Track
Expires: 25 August 2022

D. Trossen
Huawei Technologies
21 February 2022

Realizing Forward Requests Return Multicast Semantic with BIER
draft-trossen-bier-frm-00

Abstract

This document introduces a semantic for multicast that is based on initiating forward requests, resulting in dynamic return multicast to the set of initiating clients. The key dynamic nature here is represented in the return multicast being possibly different for every transmission.

We introduce this semantic more formally, present exemplifying use cases and then focus on realizing this semantic using BIER technology.

Although this document formally introduces the FRRM semantic as a new communication semantic, it does not intend to show the realization of it through any other means besides BIER. This is left for separate documents, if desired.

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). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on 25 August 2022.

Copyright Notice

Copyright (c) 2022 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

FRRM

February 2022

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

Table of Contents

1.	Introduction	2
2.	Abbreviations	3
3.	Definition of FRRM Semantic	4
4.	Use Cases	5
4.1.	HTTP-based Streaming	5
4.2.	Intra-CDN Content Distribution	6
4.3.	Distributed Reasoning	6
5.	Requirements	6
6.	BIER Architecture	6
6.1.	Description of a BIER Multicast Overlay	8
6.2.	BIER Multicast Overlay Components	9
7.	Protocol Interactions	9
7.1.	BIER Multicast Overlay Operations	9
7.2.	Achieving Multicast Responses	11
7.3.	BIER Multicast Overlay Traffic Management	12
8.	Upper Layer Considerations	13
8.1.	Application (Protocol) Considerations	13
8.2.	Transport Protocol Considerations	13
9.	Conclusions	13
10.	Security Considerations	13
11.	Privacy Considerations	13
12.	IANA Considerations	13
13.	Acknowledgements	14
14.	Informative References	14
	Author's Address	15

[1.](#) Introduction

Multicast communication semantics complements unicast communication with the ability to define the delivery of packets to more than one destination. For instance, [[RFC4607](#)] defines source-specific multicast (SSM) as

A datagram sent with source IP address S and destination IP address G in the SSM range is delivered to each host socket that has specifically requested delivery of datagrams sent by S to G, and only to those sockets.

The nature of the 'specific request' for delivery is reflected in an explicit group management protocol, e.g., [[RFC4604](#)] through which a host can request that delivery, becoming a member of the group (address) G as a consequence.

In this document, we introduce a different multicast semantic where the nature of the 'specifically requested delivery' is that of a forward request to the server S, which in response either adds the response to that request to an existing multicast group or forms a new one on-demand. The nature of the multicast group, equivalent to G in [[RFC4607](#)], is ephemeral, limited by the time it takes to send a response to the (dynamically created) group of respondents.

Multicast semantics of the aforementioned nature have been exploited and realized in previous work, such as [[ICC2016](#)] [[POINT2016](#)], focussing on HTTP-based forward requests with multicast-delivered HTTP responses.

These works were transferred onto a BIER-based systems in a previous draft [[I-D.ietf-bier-multicast-http-response](#)]. Similarly, this draft focussed entirely on the delivery of HTTP responses under such multicast semantic, progressing to WG last call in 2021. Due to organizational reasons on the side of (some of) the authors of this draft, comments from the BIER and application area community were not be possible to address with the draft finally abandoned in 2022.

This document aims to take this initial work in [[I-D.ietf-bier-multicast-http-response](#)] further by (a) formally defining the underlying communication semantic for use across a number of use cases, (b) outline use case examples beyond 'just HTTP', (c) formulate requirements for its realization and (d) outline the realization in BIER.

[2.](#) Abbreviations

The following abbreviations are used throughout the remainder of this

draft, with reference to [[RFC8279](#)] and [[I-D.ietf-bier-te-arch](#)] for the definition (and technical explanation) of those terms:

BFER : Bit-Forwarding Egress Routers

BFIR : Bit-Forwarding Ingress Routers

BFR : Bit-Forwarding Routers

PCE : Path Computation Element

SH : Service Handler

Trossen

Expires 25 August 2022

[Page 3]

Internet-Draft

FRRM

February 2022

NAP : Network Access Point

[3.](#) Definition of FRRM Semantic

As the name FRRM (forward requests return multicast) indicates, multicast communication under this semantic is initiated through one or more forward request communication, i.e., from one or more of the eventual receivers of the multicast response(s).

More formally, we define the FRRM semantic as

A datagram with source address *S* towards destinations *D1*, ..., *Dn* is formed as one or more responses to adequate requests from *D1*, ..., *Dn* to *S*, where the ephemeral group address *R* is defined through an identifying characteristic across all received requests from *D1*, ..., *Dn*.

Where

'identifying characteristic' is an implementation-specific parameter used to distinguish among different requests (e.g., identifiers such as URIs) from any of the *D1*, ..., *Dn* to *S*.

The nature of FRRM multicast is inherently dynamic, i.e., the multicast responses are formed in response to incoming requests. One or more responses may be created for the ephemeral group that is being formed, thereby supporting request-response patterns as much as initiated streaming patterns (i.e. a single request may lead to a stream of responses).

The ephemeral groups are not unique but several may exist with different receiver groups each. This may happen when a set of forward requests arrives before time t_1 , upon which the server S decides to form the ephemeral group R towards the senders of those requests, while another group R may be formed for those incoming forward requests arriving after time t_1 and before another checkpoint time t_2 .

The decision when to form an ephemeral group R as a result of incoming forward requests is entirely left to the implementation considerations at the server S and may depend on the specific use case (see [Section 4](#)).

[4.](#) Use Cases

This section expands on the original HTTP-focussed use case in [\[I-D.ietf-bier-multicast-http-response\]](#) (still listed as an example in [Section 4.1](#)) through utilizing the FRRM semantic in, e.g., CDN optimization, distributed AI and more.

[4.1.](#) HTTP-based Streaming

Referring to the BIER Use Cases [\[I-D.ietf-bier-use-cases\]](#), multicast is used to scale HTTP Live Streaming (HLS) to a large number of receivers that use HTTP. Multicast can speed up both live and high-demand VoD streaming. Adaptive Bit Rate IPMC [\[TR_IPMC_ABR\]](#) describes the use of IP Multicast towards the CMTS or a box beside it, where the content is converted to HTTP/TCP to stream to the receivers (e.g., homes). A server hosting the HLS content is shown as "NAP Server". The gateways acting as receivers for the multicast from the server are shown as "Client-NAP" (CNAP). Each CNAP can serve multiple clients.

Dynamic Adaptive Streaming (DASH) [\[ISO_DASH\]](#) over HTTP is another HTTP-based streaming approach. In DASH, each media is described by a

Media Presentation Description (MPD) file, through which a DASH client (e.g. a media player) is instructed how to download, interpret and play the media. The media content is encoded into fragments or chunks at different bit rates. Both the MPD and media fragments are stored at a server. The DASH client first needs to retrieve the MPD file from the server; then it can start to retrieve media fragments encoded at different bits rates from the server. DASH players may use rate adaptation, i.e., switching the retrieval from one rate chunks to another rate. Usually this rate adaptation is utilizing delay measurements, resulting in TCP like behavior in terms of backoff in case of increasing delay. DASH has been designed to reuse most of existing Internet infrastructure and protocols and can run over different underlying transports including HTTP. For example, two major media service providers Netflix and Youtube use DASH over HTTP as their streaming technology.

HTTP request and response used in media streaming services like HLS and DASH over HTTP, use HTTP responses for delivery of content, i.e., each chunk is returned as an HTTP response to the requesting client. In such scenarios, where semi-synchronous access to the same resource occurs (such as watching prominent videos over Netflix or similar platforms or live TV over HTTP), traffic grows linearly with the number of viewers since the HTTP-based server will provide an HTTP response to each individual viewer. This poses a significant burden on operators in terms of costs and on users in terms of likely degradation of quality.

The use of HTTP-based streaming of video content is not limited to traditional TV broadcasting. Consider a virtual reality use case where several users are joining a VR session at the same time, e.g., centered around a joint event. Hence, due to the temporal correlation of the VR sessions, we can assume that multiple requests are sent for the same content at any point, particularly when viewing angles of VR clients are similar or the same. Due to availability of virtual functions and cloud technology, the actual end point from where content is delivered may change.

HTTP streaming is not limited to video content, however. Software updates for, e.g., mobile devices or vehicles, become increasingly important, introducing point-to-multipoint traffic from a software server to devices. Using V2X as an example, the software server could be a part of telecom operators or maintained by car

manufacturers. In either case, the software server keeps vehicle software or firmware images, which will be transmitted to many vehicles across the global Internet, based on a pull or push model. HTTP is commonly used for those software updates to provide an E2E transport between the software server and each vehicle requesting software updates. As a result, the traffic from the software server to vehicles increases linearly with the number of connected vehicles since each vehicle will establish a HTTP connection with the software server.

4.2. Intra-CDN Content Distribution

More to come here based on the work outlined in [[fCDN](#)]

4.3. Distributed Reasoning

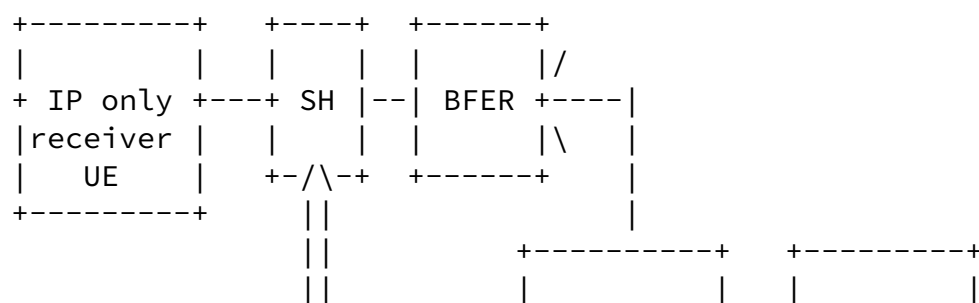
TOD0: solicit a co-author with more background to cover this

5. Requirements

TBD: capture formal requirements here

6. BIER Architecture

Figure 1 shows the architecture for FRRM over a BIER overlay.



forward requests and the responses, possibly delivered via BIER multicast capabilities.

The SH on the server side is responsible for aggregating the relevant incoming forward requests and sending one or more BIER-based multicast response to multiple clients who requested the same content, following the FRRM semantic in [Section 3](#).

[6.1](#). Description of a BIER Multicast Overlay

The Service Handler (as in Figure 1) in BIER Multicast Overlay, process the service request. At the service level, e.g., for an HTTP service, the fixed relationship among consumer and providers may be abstracted using "Service Names", and the changing relationship at the Service execution endpoints can be managed at the "multicast overlay" level, handing out the exact locations where service requests or responses needs to be sent to BIER layer.

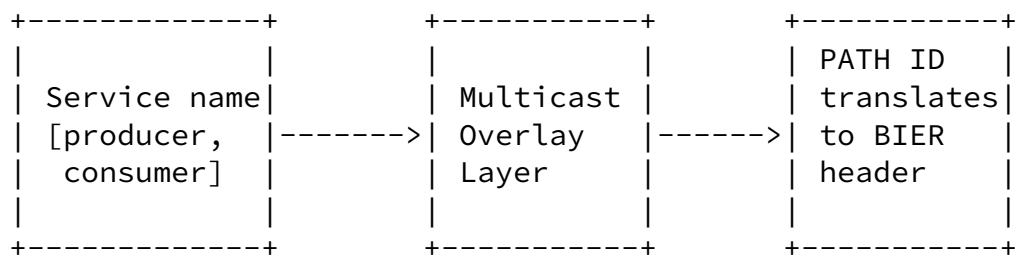


Figure 2: Service Name to Path ID Translation

It should be noted, a number of identifiers can be used as service name, such as a URI or an IP address. In the example illustration, other layers such as TCP, IP have been terminated at the egress point. Outside BIER domain we terminate TCP/IP session to extract the service name to be processed by the "multicast overlay" layer to generate the PATH IDENTIFIER, which is used as BIER header.

Path Identifier or PATH ID, is used in path-based approach, which utilizes path information provided by the source of the packet for forwarding said packet in the network. This is similar to segment routing albeit differing in the type of information provided for such source-based forwarding.

Once the BIER header is determined and added at the BFIR, the rest of the transport layers is assumed to be any underlay technology as supported by BIER. We assume TCP friendly transport, which can assure reliable delivery.

[6.2.](#) BIER Multicast Overlay Components

With reference to Figure 1, the following components are part of BIER Multicast Overlay Layer.

1. Service Handler (SH): The Service handler terminates transport level protocols, such as TCP, and extracts the service name. It processes the service name in order to determine the PATH ID by contacting the PCE for a suitable path resolution, which in turn is used to send the service request.
2. Optional PCE : Path Computation Element keeps track of all service execution end points through a registration process. SH interacts with the PCE to obtain PATH information by resolving the service name at the ingress SH to a suitable PATH ID.
3. Interface functions to BFIR where the PATH ID is mapped to BIER header. An Interface to the BFER is likely not required because the BFER will only receive the traffic that they need and should be able to derive from the BIER payload which subset of its receivers need to get an encapsulated version of a particular reply.

[7.](#) Protocol Interactions

[7.1.](#) BIER Multicast Overlay Operations

As shown in Figure 3, the "Multicast overlay function" includes a function called PCE (Path Computation Element function), which is responsible for selecting the correct multicast end point and possibly realizing path policy enforcement. The result of the selection is a BIER path identifier, which is delivered to the SH upon initial path computation request (or provided to the ingress router BFIR to be added as BIER header) (i.e., when sending a request to or response for a specific URL for the first time). The path identifier is utilized for any future request for a given URL-based request.

All service end points indicate availability to the PCE through a registration procedure, the PCE will instruct all SHs to invalidate previous path identifiers to the specific URL that might exist. This may result in an a renewed path computation request at the next service request forwarding. Through this, the newly registered service endpoint might be utilized if the policy-governed path computation selects said service instance. Otherwise, a previously resolved PATH ID for the URI determined at the ingress SH is being

used instead, removing any resolution latency to an SH-local lookup of the PATH ID.

Internet-Draft

FRRM

February 2022

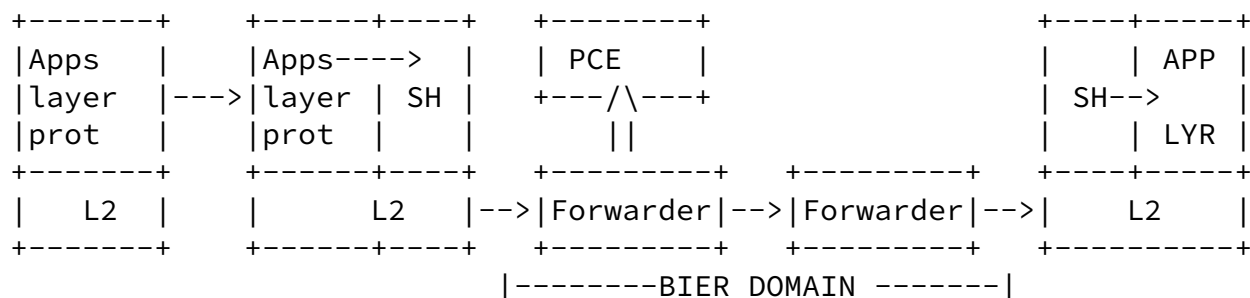


Figure 3: Protocol Stack for Multicast Overlay Layer

In the diagram shown above, a service request is sent by an IP-based device towards the service name of the server defined in the service request.

At the client facing SH, the service request is terminated at the TCP level. The server side SH at the egress terminates any transport protocol on the outgoing (server) side. These terminating functions are assumed to be part of the client/ server SH. As a consequence, the SH obtains the destination "Service Name" from the received service request.

If no local BIER forwarding information exists at the client side SH, the path computation entity (PCE) is consulted, which calculates a unicast path from the BFIR to which the client SH is connected to the BFER to which the server SH is connected. The PCE provides the forwarding information (Path ID) to the client SH, which in turn caches the result. The Client SH may forward the Path ID to BFIR, which creates the BIER header.

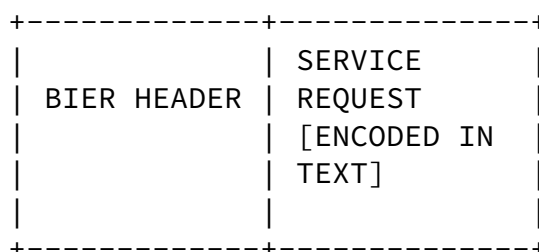


Figure 4: Encapsulation of Service Request

Ultimately, the "Service Request" encapsulated by BIER header, as shown in above diagram, is forwarded by the client SH towards the server- facing SH via the local BFIR. We assume a (TCP-friendly) transport protocol being used for the transmission between client and server SH. The possibility of sending one service response to

several SHs makes this a reliable multicast transport protocol. The exact nature of this transport protocol is left for further studies. A suitable transport or Layer 2 encapsulation, as supported by BIER layer, is added to the above payload.

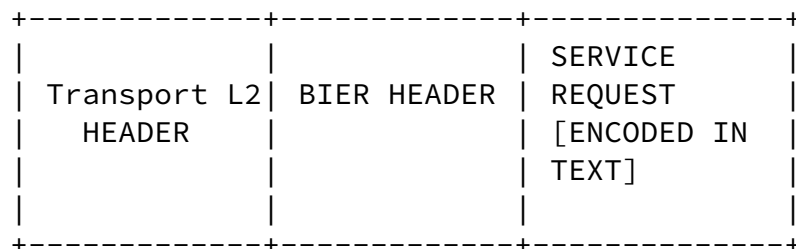


Figure 5: Transport Encapsulation of BIER payload

Upon arrival of a service request at the server SH, it forwards the service request as a well-formed service request locally to the server, awaiting an service response for the reverse direction.

If no BIER forwarding information exists for the reverse direction towards the requesting client SH, this information is requested from the PCE, similar to the operation in forward direction.

[7.2.](#) Achieving Multicast Responses

Upon arrival of any further client SH request at the server SH to an service request whose response is still outstanding, the client SR is added to an internal request table. Optionally, the request is suppressed from being sent to the server.

Upon arrival of a service response at the server SH, the server SH consults its internal request table for any outstanding service requests to the same request. The server SH retrieves the stored BIER forwarding information for the reverse direction for all outstanding service requests and determines the path information to all client SHs through a binary OR over all BIER forwarding identifiers with the same SI field. This newly formed joint BIER multicast response identifier is used to send the service response across the network.

BIER makes the solution scalable. Instead of IP Multicast with IGMP/PIM, BIER is being used between Server SH and client SHs, the server-facing SH simply coalesces the forwarded service requests from the client-facing SH, and determines for every requested block the set of SHs requesting it. A set of SHs corresponds to a set of bits in the BIER-bitstring, one bit per SH. The SH then sends the block into BIER with the appropriate bitstring set.

This completely eliminates any dynamic multicast signaling between client-facing SHs and server-facing SH. It also avoids sending of any unnecessary data block.

Furthermore, using the approach with BIER, the server-facing SH can also easily control how long to delay sending of blocks. For example, it may wait for some percentage of the time of a block (e.g, 50% = 1 second), therefore ensuring that it is coalescing as many requests into one BIER multicast answer as possible. The realization of such controlled sending of a single response, however, needs further study as to the possible interaction with upper-layer methods, particularly congestion control.

[7.3.](#) BIER Multicast Overlay Traffic Management

BIER-TE (BIER Traffic Engineering [[I-D.ietf-bier-te-arch](#)]) forwards and replicates packets like BIER based on a BitString in the packet

header. Where BIER forwards and replicates its packets on shortest paths towards BFER, BIER-TE allows (and requires) to also use bits in the bitstring to indicate the paths in the BIER domain across which the BIER-TE packets are to be sent. This is done to support Traffic Engineering for BIER packets via explicit hop-by-hop and/or loose hop forwarding of BIER-TE packets. A BIER-TE controller calculates explicit paths for this packet forwarding.

The Multicast Flow Overlay operates as in BIER. Instead of interacting with the BIER layer, it interacts with the BIER-TE Controller.

In this draft, "Name-based" service forwarding over BIER, is described to handle changes in service execution end points and manage adhoc relationship in a multicast group. BIER-TE is another way of doing this, while integrated with BIER architecture. The PCE function described earlier in the BIER Multicast Overlay, may become part of BIER-TE Controller. The server- and client-facing SH function communicates with the BIER TE controller. SH sends the service name to the controller, which processes the request using the PCE function and returns the "bitstring" to be used as BIER header for delivery of the service response to multiple clients.

[8.](#) Upper Layer Considerations

[8.1.](#) Application (Protocol) Considerations

TBD: add something on DASH here, app considerations for using SH capabilities, ...

[8.2.](#) Transport Protocol Considerations

TBD: move transport discussions in [Section 7](#) into this place and pose the challenges that need addressing beyond the FRRM aspect at the BIER level

[9.](#) Conclusions

This draft generalizes the work in [\[I-D.ietf-bier-multicast-http-response\]](#) beyond HTTP being the application protocol. Instead, this draft proposes a general

multicast semantic termed 'forward requests return multicast' (FRRM), which can be utilized by any application layer protocol atop a BIER transport network.

10. Security Considerations

The operations in [Section 7](#) consider the forwarding of service request packets between ingress and egress points based on information derived from the service request. The support for transport-level security, e.g., TLS, is foreseen to ensure suitable encryption capability of such exchanges. For this to happen, we expect certificate sharing agreements to exist between the content provider and the BIER overlay provider, ensuring the extraction of the suitable request parameters while allowing for the re-encryption of the content for an encrypted delivery over the BIER network. Since we liken the relationship between content and BIER overlay provider to that between content and CDN provider, the existence of certificate sharing agreements is similar to the practice used for CDNs.

11. Privacy Considerations

TBD: Anything here on exposing request IDs?

12. IANA Considerations

This draft does not request any IANA action.

13. Acknowledgements

This draft originated from the narrower use case described in [\[I-D.ietf-bier-multicast-http-response\]](#). We acknowledge the work that has gone into the development of this draft and the contributions through discussions on the mailing list. We specifically thank Toerless Eckert, Sebastian Robitzsch, Akbar Rahman and Chonggang Wang for their contributions to the original draft, some of which have been transferred to this draft, where appropriate.

14. Informative References

[DVB_REF_ARCH]

DVB, "Adaptive media streaming over IP Multicast", Report DVB Document A176, 2018, <https://www.dvb.org/resources/public/standards/a176_adaptive_media_streaming_over_ip_multicast_2018-02-16_draft_bluebook.pdf>.

[fCDN]

Al-Naday, M., Reed, M., Riihijarvi, J., Trossen, D., Thomos, N., and M. Al-Khalidi, "fCDN: A Flexible and Efficient CDN Infrastructure without DNS Redirection or Content Reflection", Paper Arxiv, 2018, <<https://arxiv.org/abs/1803.00876>>.

[I-D.ietf-bier-multicast-http-response]

Trossen, D., Rahman, A., Wang, C., and T. Eckert, "Applicability of BIER Multicast Overlay for Adaptive Streaming Services", Work in Progress, Internet-Draft, [draft-ietf-bier-multicast-http-response-06](https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06), 10 July 2021, <<https://www.ietf.org/archive/id/draft-ietf-bier-multicast-http-response-06.txt>>.

[I-D.ietf-bier-te-arch]

Eckert, T., Menth, M., and G. Cauchie, "Tree Engineering for Bit Index Explicit Replication (BIER-TE)", Work in Progress, Internet-Draft, [draft-ietf-bier-te-arch-12](https://www.ietf.org/archive/id/draft-ietf-bier-te-arch-12), 28 January 2022, <<https://www.ietf.org/archive/id/draft-ietf-bier-te-arch-12.txt>>.

[I-D.ietf-bier-use-cases]

Kumar, N., Asati, R., Chen, M., Xu, X., Dolganow, A., Przygienda, T., Gulko, A., Robinson, D., Arya, V., and C. Bestler, "BIER Use Cases", Work in Progress, Internet-Draft, [draft-ietf-bier-use-cases-12](https://www.ietf.org/archive/id/draft-ietf-bier-use-cases-12), 10 September 2020, <<https://www.ietf.org/archive/id/draft-ietf-bier-use-cases-12.txt>>.

Trossen

Expires 25 August 2022

[Page 14]

Internet-Draft

FRRM

February 2022

[ICC2016]

Reed, M., Al-Naday, M., Thomos, N., Trossen, D., Petropoulos, G., and S. Spirou, "Stateless multicast switching in software defined networks", Paper IEEE ICC 2016, 2016.

- [ISO_DASH] ISO, "Information technology -- Dynamic adaptive streaming over HTTP (DASH) -- Part 1: Media presentation description and segment formats", Report ISO/IEC 23009-1:2014, 2014, <<http://standards.iso.org/ittf/PubliclyAvailableStandards/index.html>>.
- [POINT2016] Kim, S.-Y., Robitzsch, S., Trossen, D., Reed, M., Al-Naday, M., and J. Riihijarvi, "Realizing IP-based Services over an Information-Centric Networking Transport Network", Paper Proceedings of the 3rd ACM Conference on Information-Centric Networking, Pages 215-216, 2016.
- [RFC4604] Holbrook, H., Cain, B., and B. Haberman, "Using Internet Group Management Protocol Version 3 (IGMPv3) and Multicast Listener Discovery Protocol Version 2 (MLDv2) for Source-Specific Multicast", [RFC 4604](#), DOI 10.17487/RFC4604, August 2006, <<https://www.rfc-editor.org/info/rfc4604>>.
- [RFC4607] Holbrook, H. and B. Cain, "Source-Specific Multicast for IP", [RFC 4607](#), DOI 10.17487/RFC4607, August 2006, <<https://www.rfc-editor.org/info/rfc4607>>.
- [RFC8279] Wijnands, IJ., Ed., Rosen, E., Ed., Dolganow, A., Przygienda, T., and S. Aldrin, "Multicast Using Bit Index Explicit Replication (BIER)", [RFC 8279](#), DOI 10.17487/RFC8279, November 2017, <<https://www.rfc-editor.org/info/rfc8279>>.
- [TR_IPMC_ABR] CableLabs, "IP Multicast Adaptive Bit Rate Architecture Technical Report", Report OC-TR-IP-MULTI-ARCH-V01-141112 C01, 2016, <<https://community.cablelabs.com/wiki/plugins/servlet/cablelabs/alfresco/download?id=51b3c11a-3ba4-40ab-b234-42700e0d4669;1.0>>.

Author's Address

Dirk Trossen
Huawei Technologies
Munich
Germany

Internet-Draft

FRRM

February 2022

Email: dirk.trossen@huawei.com

