

Reliable multicast with B.I.E.R.

Yoann Desmouceaux, Thomas Heide Clausen, Juan Antonio Cordero, W Mark Townsley

▶ **To cite this version:**

Yoann Desmouceaux, Thomas Heide Clausen, Juan Antonio Cordero, W Mark Townsley. Reliable multicast with B.I.E.R.. Journal of Communications and Networks, 2018, 20 (2), pp.182-197. 10.1109/JCN.2018.000025 . hal-02263362

HAL Id: hal-02263362
https://polytechnique.hal.science/hal-02263362

Submitted on 4 Aug 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Reliable Multicast with B.I.E.R.

Yoann Desmouceaux, Thomas Heide Clausen, Juan Antonio Cordero Fuertes, W. Mark Townsley

Abstract: **Inter-network multicast protocols, which build and maintain multicast trees, incur both explicit protocol signalling, and maintenance of state in intermediate routers in the network. B.I.E.R. (Bit-Indexed Explicit Replication) is a technique which can provide a multicast service yet removes such complexities: intermediate routers are unencumbered by group management, and no per-group state is to be maintained.**

This paper explores the use of B.I.E.R. as a basis for developing an efficient and reliable multicast mechanism, where redundant traffic is avoided, essential traffic is forwarded along shortest paths, and no per-flow state is required in intermediate routers. Evaluated by way of both an analytical model and network simulation both in generic and in real network topologies with varying background traffic loads, the proposed B.I.E.R.-based reliable multicast mechanism exhibits attractive performance attributes: it attains delivery success rates as high as any other reliable multicast service, but with significantly better link utilisation and no per-flow or per-group state in intermediate routers of the network.

Index Terms: **multicast, reliable multicast, Bit Indexed Explicit Replication (BIER), scalability, performance evaluation**

I. INTRODUCTION

Developed alongside their unicast counterparts, multicast protocols were never offered as universally available network services in the Internet [1] – in part, as the operational complexity of multicast was perceived as exceeding the potential benefits from efficient one-to-many distribution of content. "Native" multicast was therefore, when available, confined to within single networks (or even, to within single links) – and multicast between sites ("inter-network multicast") was established by way of overlays (*e.g.,* MBONE [2]). The complexity of multicast protocols is, in part, due to their group-based nature: schematically, when a client wishes to receive messages sent to a multicast group, it will explicitly and periodically send *join* messages to its "local multicast router" (using IGMP). This router will forward these *join* message upwards in the multicast tree (using *e.g.,* PIM [3]), until reaching the multicast source. Intermediate routers are expected to build, and maintain, flow state (*a minima*, a multicast tree) for as long as *join* messages are regularly received, in order to provide connectivity to all members of the multicast group.

Bit-Indexed Explicit Replication (B.I.E.R.) [4] was designed to eliminate this complexity, and to enable lightweight inter-network multicast – with the ambition being that intermediate routers maintain no flow state, other than that of an existing unicast routing table, and that intermediate routers are not involved in group management. The key idea in B.I.E.R., which is detailed in the below, is that the source of a multicast data packet encodes the set of destinations (*i.e.,* the members of the group) as a bit-string, and includes this bit-string in the header of each multicast data packet. Intermediate routers only need to be able to interpret that bit-string – leaving group management (if any) a matter for only the clients and the source. A proposal to increase the resiliency of multicast distribution with B.I.E.R. or B.I.E.R.-TE[1] with fast re-route mechanisms is introduced in [5].

The authors are with École Polytechnique, 91128 Palaiseau, France, emails: {Yoann.Desmouceaux, Thomas.Clausen, Juan-Antonio.Cordero-Fuertes, Mark.Townsley}@polytechnique.edu. Y. Desmouceaux and W. M. Townsley are also with Cisco Systems Paris Innovation and Research Laboratory (PIRL), 92782 Issy-les-Moulineaux, France, emails: {ydesmouc,townsley}@cisco.com.

A. Bit-Indexed Explicit Replication (B.I.E.R.)

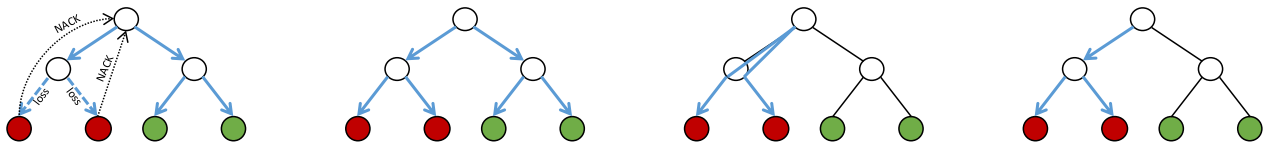
B.I.E.R. [4] is a multicast mechanism wherein the source of a multicast data packet explicitly identifies and indicates the set of destinations by way of inserting a destination bit-string into each multicast data packet. Each bit in the destination-bit-string corresponds to a destination in the network towards which – if the bit is set – the multicast data packet will be forwarded. Upon receipt of a multicast data packet, a B.I.E.R. router consults its unicast routing table in order to identify those interfaces over which copies of the received multicast data packet are to be sent, so as to follow the shortest path to each destination. When sending the multicast data packet over an interface, the B.I.E.R. router will clear all bits in the destination bit-string, except for those destinations for which the shortest path is via this interface.

B.I.E.R. thus offers multicast as a network service, by way of using (but, not constructing) a shortest-path source-tree, rooted in the source of each multicast data packet. A B.I.E.R. router requires neither per-group nor per-flow multicast forwarding state, thus there is also no multicast group maintenance signalling: every group exists a priori, by way of the destination bit-string and the unicast routing table. The only requirements for a router participating in B.I.E.R. multicast that it runs an (any) unicast routing protocol, and is able to perform the bit#→IP mapping.

The destination bit-string is a substitution (or, dictionary) coder: the position of a set bit maps to a destination IP address. The precise way in which the dictionary is constructed is of no algorithmic significance, although it is interesting to note that if coupled with destination address assignment, it is possible to make this a simple mapping function: a simple example is to use the bit number as the value of a well defined octet in an IPv6 address `<head-bits>:bit#:<rest-of-address>`.

The way in which the destination bit-string is carried in each multicast data packet is also without algorithmic significance, though several options exist. If address assignment and number of destinations permit, the destination bit-string can be encoded as part of the destination IPv6 address, *e.g.,* `<head-bits>:bit-string:<rest-of-address>` – in which case the destination bit-string incurs no extra over-the-wire overhead. Alternatively, the destination bit-string can be

[1] B.I.E.R.-Traffic Engineering, encoding not only destinations but also links to be traversed (thus, no need for an underlying routing protocol).



(a) Initial transmission (b) Multicast retransmission (c) Unicast retransmission (d) B.I.E.R. retransmission

Fig. 1. Comparison of different reliable multicast mechanisms. In this example, two clients do not receive a packet and send a NACK to the source (a). Multicast retransmissions (b) will vainly incur traffic towards the clients that had successfully received the packet. With unicast retransmissions (c), identical packets will be transmitted on the same link. B.I.E.R. retransmissions (d) ensure that the traffic footprint is minimal.

carried as an IPv6 extension header – which does incur extra octets in each multicast data packet, but at the same time allows an unlimited number of destinations, and is independent from any address assignment constraints or policies.

B. Statement of Purpose

The original B.I.E.R. specification [4] emphasises the flexibility provided by B.I.E.R. at the flow level, allowing for adding to and removing from the set of destinations of a flow without the need of building multicast trees and maintaining flow state in the routers. It is possible to take this one step further, to modifying the set of destinations on a per-packet basis, and to do so entirely without incurring any additional overhead.

This flexibility can be used to develop efficient reliable multicast: if the source of a multicast data packet is informed as to the set of destinations, to which a retransmission is required, it can use B.I.E.R. for minimising the traffic footprint of this retransmission: set the bit-string so as to contain only the destinations affected by a multicast data packet loss, thus the retransmission will be forwarded only along the shortest path tree covering the source and these destinations.

This paper studies this use of B.I.E.R. for reliable multicast – and, compares this "*reliable B.I.E.R.*" to two reliable multicast references: (i) the mechanisms known from *e.g.,* NORM [6], in which the source – when informed about a destination in the multicast group being affected by a multicast data packet loss – will retransmit the multicast data packet to all destinations, and (ii) retransmission by way of unicast(s) to those destinations affected by a multicast data packet loss: this is for instance the case in RMTP [7], if the number of affected destinations is below a given threshold.

Figure 1 illustrates the intuition that when faced with a multicast data packet loss (figure 1a), a NORM-style retransmission from the source and to the entire multicast group (naturally) will impose a load on all links in the multicast tree, regardless of if they lead to destinations affected by a multicast data packet loss (figure 1b). Unicast retransmissions (figure 1c), while traversing only the shortest-path tree between the source and the destinations affected by a multicast data packet loss, may cause the same multicast data packet to be retransmitted across the same link multiple times – whereas B.I.E.R. utilises only the shortest path tree between the source and the destinations affected by a multicast data packet loss, with each packet only retransmitted across the same link once (figure 1d). This paper formalises a simple, reliable, multicast mechanism using B.I.E.R., and examines if the suggested intuition holds – and in which conditions. To that purpose, network simulations are conducted, and an analytical model is developed.

C. (Semi-)Reliable Multicast

While the term "reliable" is used throughout this paper, and is generally used in literature, it is perhaps more realistic to describe the attained multicast network services as semi-reliable. For example, maintaining a retransmission buffer (regardless if centralised at the source or distributed / peer-based) indefinitely is hardly feasible – nor will excessive retransmissions necessarily increase the overall success rate across heavily congested paths. Therefore, this paper will not consider mechanisms whereby a source adapts its sending rate to the worst destination; rather, it will assume that the source sends a stream at a fixed rate (*e.g.,* a live broadcast media stream), and that destination applications might decide to give up on certain packets – if they are behind heavily congested links, and/or if they no longer would need the packet after retransmission.

D. Related Work

While never widely deployed as an inter-networking service, several reliable multicast protocols have been developed [8]. "Log-Based Receiver-reliable Multicast" (LBRM) [9] uses a *log server* for caching packets sent by the source, and which also reacts to requests for retransmissions. A hierarchical architecture is suggested: a destination that has not received a packet will first solicit retransmission from a local *log server* – and, only if that fails, solicit a primary *log server*. In the "Reliable Multicast Transport Protocol" (RMTP), [7] proposes a hierarchical architecture, wherein intermediate routers in the multicast tree, will contribute retransmissions in case of isolated losses, but with global recovery handled by the source. [10] introduces "Scalable Reliable Multicast" (SRM), which uses receiver-based reliability (receivers detect losses and request retransmissions) combined with low-rate multicast by every member to report the highest sequence number received. The "Tree-based Multicast Transport Protocol" (TMTP) [11] is another instance of a reliable multicast protocol, wherein destinations are grouped in a tree of different domains, within which local recovery can be performed.

The IETF[2] standardised "Negative-acknowledgment Oriented Reliable Multicast" (NORM) [12], using NACKs and source-based retransmissions to attain reliability. NORM also proposes redundancy and recovery by way of Forward Error

[2] `http://www.ietf.org/`

Coding (FEC) – either proactively, or in response to negative acknowledgements. A TCP-friendly congestion-control mechanism has been proposed in [13]. The IETF has also examined the "Pragmatic General Multicast" (PGM) protocol [14], which uses negative acknowledgements and local repairs.

In the context of data-centers, [15] proposes the end-host based protocol "Reliable Data Center Multicast" (RDCM): through a central controller, RDCM explicitly builds a multicast tree, and a multicast-tree-aware backup overlay, for data dissemination. Retransmissions are performed on a peer-to-peer (unicast) basis: every receiver is responsible for providing, if needed, retransmissions for up to two of its peers.

The performance of (reliable, and otherwise) multicast protocols has been studied both analytically and through network simulations. For example, [16] develops an analytical model and carries simulations to study the performance of a generic reliable block-based multicast protocol using stop-and-wait, positive acknowledgements, and selective retransmissions. This model quantifies the number of transmission attempts until full reception, assuming independent losses in different links. [17] investigates the optimal placement of FEC in reliable multicast trees, by way of studying generic models of such trees (*i.e.,* a single path common to all receivers, a set of completely separate paths to each receiver) and a refinement of the model developed in [16]. The number of successful receptions for different types of trees is studied by way of analysis and simulation in [18], which also derives a generic approximation for the expected number of transmissions for reliable delivery.

Finally, models relying on TCP overlays for multicast have also been developed: [19] introduces the *One-to-Many TCP Overlay* for reliable multicast services, as an application-level multicast alternative to IP reliable multicast; [20] studies the performance of TCP-based reliable multicast trees, built as a set of reliable point-to-point links, in data-centers.

E. Paper Outline

The remainder of this paper is organised as follows: section II details the use of B.I.E.R. for light-weight, reliable multicast. Section III and section IV evaluate the performance of this reliable multicast mechanism (denoted *reliable B.I.E.R.*) in different topologies and with different losses, by way of network simulations – and compare the performance with other reliable multicast mechanisms. Generalising the observations from the network simulations, section V provides an analytical study of *reliable B.I.E.R.* performance. Section VI concludes this paper.

II. RELIABLE B.I.E.R. – SPECIFICATION

To provide reliable multicast as a network service, transparent to applications, *reliable B.I.E.R.* is designed to operate as a shim-layer above the network layer, as depicted in figure 2. This *reliable B.I.E.R. shim layer* is, of course, upper-layer agnostic, and as such supports both transport layers (*e.g.,* UDP) as well as IP-in-IP.

The *reliable B.I.E.R. shim layer* assumes a unique flow ID from the upper layer, and maintains for each flow ID a sequence number, monotonically increased for each new packet being handed by the upper layer. The tuple (flow ID, sequence number) allows uniquely identifying each original multicast data packet in the network, identifying when a multicast data packet is received out of order, etc. For many transport layers, the flow ID would be a hash of the tuple (protocol number, source IP address, source port), whereas for more specific transport layers a transport-layer-specific identifier could be used (*e.g.,* the session ID for QUIC [21]).

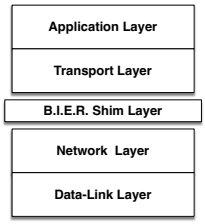


Fig. 2. B.I.E.R. shim layer

For each outgoing multicast data packet, the *reliable B.I.E.R. shim layer* takes a destination set, rather than a single destination IP address, from the upper layer – and compresses this into the destination-bit-string. The destination-bit-string and the sequence number comprise the *reliable B.I.E.R. header*, to be included in each multicast data packet. The precise form of the *reliable B.I.E.R. header* has no algorithmic significance – but in an IPv6 context can, for instance, take the form of a destination options extension header.

For each incoming multicast data packet, the *reliable B.I.E.R. shim layer* at each destination will inspect the sequence number to detect losses, and signal losses by way of sending negative-acknowledgements (NACKs) from destinations towards the source. Having this signalling only involve the end-points (source and destinations) serves, in part to run the protocol over any "standard" set of B.I.E.R. routers doing best-effort forwarding[3], and in part to facilitate deployment (only the end-points need to agree on parameters, for example). As in other reliable multicast protocols, *e.g.,* [6], [14], NACKs are used in order to avoid an "ACK storm". Of course, for very large error rates or errors affecting a wide range of destinations, this may lead to a "NACK storm". Such a storm of control traffic could be avoided *e.g.,* by allowing intermediate routers to aggregate NACKs before forwarding them upwards, without changing the end-to-end behaviour specified in this paper.

For the purpose of this paper, a slightly modified socket API is used – specifically, allowing the sender to provide the set of destinations (rather than a single multicast group address) to which a multicast data packet is to be forwarded.

A. Source Operation

A *reliable B.I.E.R.* source operates as detailed in algorithm 1. On sending a packet through a socket, the *reliable B.I.E.R. shim layer* caches a copy of the packet, with which it associates a

[3]It is to be noted, however, that if intermediate routers provides caching capabilities [22], they could be extended to intercept NACKs and perform retransmissions in place of the source. This would not change the end-to-end behaviour as specified in this paper.

Algorithm 1 *Reliable B.I.E.R.* Source Operation

$\Delta t_{agg} \leftarrow$ NACK aggregation delay
$w \leftarrow$ packet cache window size
$F \leftarrow$ unique flow identifier
$B \leftarrow$ destination bitstring
$S \leftarrow 0$ ▷ *sequence number*
$C \leftarrow \{\}$ ▷ *packet cache*
$R \leftarrow \{\}$ ▷ *retransmit bitstrings*
$T \leftarrow \{\}$ ▷ *retransmit timers*
for each outgoing packet p **do**
  insert *reliable B.I.E.R.* header with flow F, seq S
  insert B.I.E.R. header with bitstring B
  transmit p
  $C[S] \leftarrow p, R[S] \leftarrow 0, T[S] \leftarrow \infty$
  delete $C[S-w], R[S-w], T[S-w]$ ▷ *garbage collection*
  $S \leftarrow S + 1$
  for $s \in C$ **with** $T[s] \leq T_{now}$ **do** ▷ *perform retransmits*
    $p \leftarrow C[s]$ ▷ *retrieve cached packet*
    insert *reliable B.I.E.R.* header with flow F, seq s
    insert B.I.E.R. header with bistring $R[s]$
    transmit p
    $R[s] \leftarrow 0, T[s] \leftarrow \infty$
  end for
end for
for each received NACK packet with flow F, seq s, bit b **do**
  if $s \in C$ **then**
    $R[s] = R[s]$ OR 2^b ▷ *add b to the retransmit bitstring*
    $T[s] \leftarrow \min\{T[s], T_{now} + \Delta t_{agg}\}$ ▷ *schedule retransmit*
  end if
end for

Algorithm 2 *Reliable B.I.E.R.* Destination Operation

$\Delta t_{retry} \leftarrow$ NACK retransmission delay
$l \leftarrow$ NACK retransmission limit
$F \leftarrow$ unique flow identifier
$L \leftarrow \{\}$ ▷ *seqnum of lost packets*
$T \leftarrow \{\}$ ▷ *NACK transmit timers for lost packets*
$N \leftarrow \{\}$ ▷ *number of times a NACK has been sent*
$C \leftarrow \{\}$ ▷ *recovered packets pending for app*
$n \leftarrow 0$ ▷ *next expected seqnum*
for each incoming packet p with flow F, seq S **do**
  if $S = n$ **then** ▷ *received in-order packet*
    transmit p to application
    $n \leftarrow n + 1$
  else if $S > n$ **then** ▷ *received out-of-order packet*
    if $S \notin C$ **then** ▷ *cache packet and un-schedule NACK*
      $C[S] \leftarrow p$
      $L \leftarrow L \setminus \{S\}$, delete $T[S], N[S]$
    end if
    ▷ *schedule NACK for packets between n and S*
    for seq **from** n **to** S **with** $seq \notin L \cup C$ **do**
      $L \leftarrow L \cup \{seq\}, T[seq] \leftarrow T_{now}, N[seq] \leftarrow 0$
    end for
  end if
  ▷ *send appropriate NACKs*
  for $seq \in L$ **with** $T[seq] \leq T_{now}$ **do**
    if $N[seq] < l$ **then**
      send NACK with flow F, seq seq
      $T[seq] \leftarrow T_{now} + \Delta t_{retry}, N[seq] \leftarrow N[seq] + 1$
    else ▷ *abort trying to recover this packet*
      $L \leftarrow L \setminus \{seq\}$, delete $T[seq], N[seq]$
      $C[seq] \leftarrow \{\}$ ▷ *put a dummy packet in the cache*
    end if
  end for
  ▷ *send pending recovered packets to application*
  while $n \in C$ **do**
    $p \leftarrow C[n]$, delete $C[n]$
    transmit p to application
    $n \leftarrow n + 1$
  end while
end for

B.I.E.R. retransmit bit-string with all bits cleared, and a timer of duration Δt_{agg}, where Δt_{agg} represents a window of time between the first NACK is received and a retransmission is made. Within this window, for each NACK received, the corresponding bit in the *B.I.E.R. retransmit bit-string* is set; at the end of this window, the cached multicast data packet is retransmitted with the destination-bit-string set to the associated *B.I.E.R. retransmit bit-string*. This permits aggregation of retransmissions to multiple destinations in a single B.I.E.R. packet, thus potentially reducing the number of transmissions of this packet. If after retransmission, a subsequent NACK for the same packet is received, a new Δt_{agg} window is opened and the aggregation mechanism is restarted.

B. Destination Operation

A *reliable B.I.E.R.* destination operates as described in algorithm 2. In short, a destination will send a NACK to the source when it detects that a packet was lost – a packet being deemed lost when one of its successors is received[4]. If necessary, NACKs for a lost packet are then retransmitted regularly by means of a timer, until a retransmission is received.

More precisely, for each incoming multicast data packet, the *reliable B.I.E.R. shim layer* parses the *reliable B.I.E.R. header* and either hands it off to the upper layer, or (if received out-of-order) records it in a buffer, C. For each multicast data packet that (1) is received out-of-order, and (2) creates a "hole" (*i.e.,* a set of missing packets between two consecutively received packets) in C, the *reliable B.I.E.R. shim layer* adds the element(s) corresponding to this "hole" in the list of lost packets, L. Each element in L is identified by the sequence number, s, of the corresponding lost packet, and is associated with a timer, $T[s]$, and a NACK count, $N[s]$.

For each element of L, a NACK is sent towards the source. The NACK contains a *reliable B.I.E.R. header* wherein the included bit-string indicates the bit of the client sending the NACK[5], and the sequence number corresponding to the lost packet. Then, the NACK count is incremented, and a new timer for this packet is set to expire after Δt_{retry} (a configurable retry delay). Upon timer expiration, if no retransmission has been received, and if the NACK count is below a configurable limit l, another NACK is sent and the process is restarted. When the retransmission count reaches l, it is assumed that the source is not able to offer timely retransmission (for instance, due to congestion on the path), and the destination gives up trying to request. This achieves a *"poor-man's congestion control"*, by limiting the number of possible retransmission of a multicast data packet.

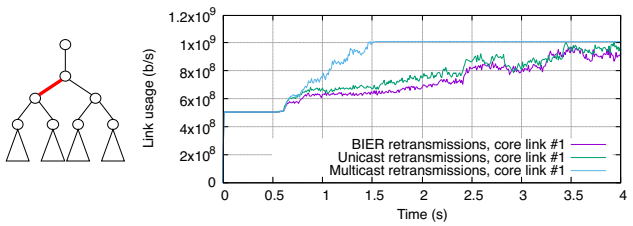
C. Parameter Discussion

The value of Δt_{agg} directly influences the global behaviour of *reliable B.I.E.R.*:

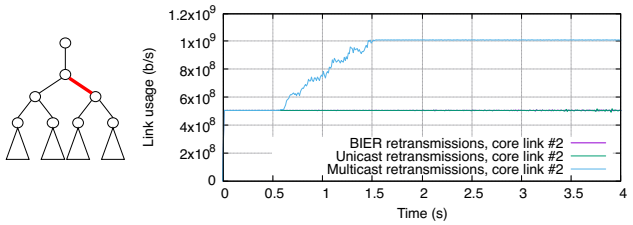
- when $\Delta t_{agg} = 0$, no aggregation is performed, and the protocol degenerates to individual, unicast-based, retransmissions;
- when $\Delta t_{agg} > 0$, aggregation is enabled: the greater Δt_{agg}, the greater the probability of aggregating retransmissions, but at

[4]The successor of the last data packet is a special end-of-connection packet. To ease readability, Algorithms 1 and 2 assume an infinite stream.

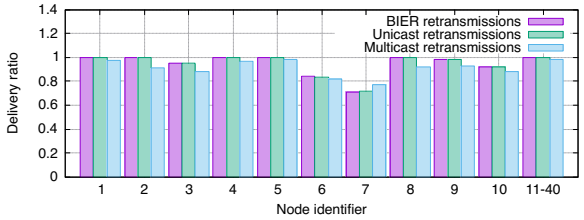
[5]One reason for including a bit-string is, that this allows the originator to create the and *B.I.E.R. retransmit bit-string* by a simple OR operation of received NACKs.



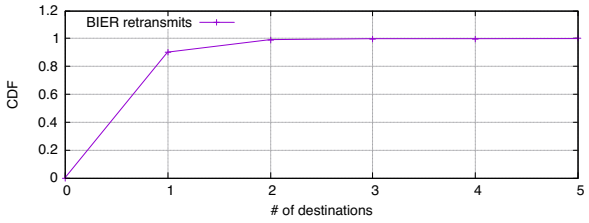
(a) Link usage, between core router and first aggregation router.



(b) Link usage, between core router and second aggregation router.



(c) Delivery ratio, for the multicast flow



(d) Cumulative Distribution Function (CDF) for the number of clients in B.I.E.R. retransmit bitstrings.

Fig. 3. Uncorrelated localised losses experiment. B.I.E.R. retransmits vs unicast and multicast retransmits.

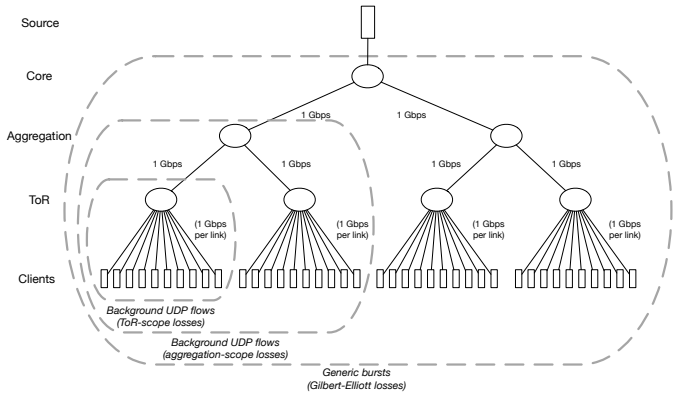


Fig. 4. Data-Center Simulation Topology

the cost of a greater delay for recovery;

- when $\Delta t_{agg} = RTT_{max} - RTT_{min}$, maximum aggregation is enabled; higher values of Δt_{agg} will not provide further benefit.

Determination of the value of Δt_{agg} thus requires taking into account RTT variations and error probability along the different paths, as well as the sensitivity of the application to delay: it thus corresponds to a policy decision. Furthermore, to make sure that each destination sends at most one NACK for each multicast data packet (re)transmission failure, Δt_{retry} should be greater than $\Delta t_{agg} + RTT_{max}$.

III. DATA-CENTER SIMULATIONS

Regardless of the underlying network topology, content delivery with B.I.E.R. from a given source will follow (but not construct) a shortest-path tree. Thus for this first set of simulations, *reliable B.I.E.R.* is tested on a simple tree-topology, modelling a data-center, depicted in figure 4: a core router, connected to two aggregation routers – each of which is connected to two Top-of-Rack switches (ToR), and with each rack hosting 10 machines.

The purpose of the set of tests in this section is to examine if the intuition, introduced in section I-B and depicted in figure 1, holds: that using B.I.E.R. (rather than multicast or unicast) for retransmissions can yield a measurable and significant diminution of the traffic footprint.

To this end, three different scenarios are constructed around the same physical topology depicted in figure 4. These scenarios serve to explore how *reliable B.I.E.R.* performs both when losses are spatially located and when they are not, specifically:

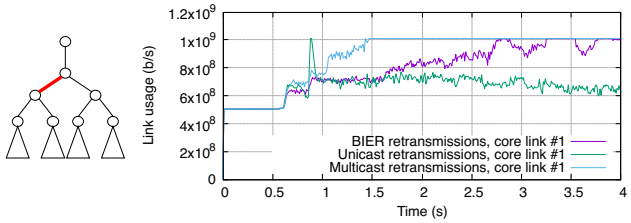
Uncorrelated localised losses, where background traffic is present inside the leftmost rack, *i.e.,* where both the source and destination of the background traffic are members of the leftmost rack, saturating individual links between the ToR switch and the machines in the rack, and thus affecting these machines individually – but with the rest of the data-center unaffected. This scenario is studied in section III-B.

Correlated localised losses, where background traffic is present inside the two leftmost racks, *i.e.,* where source and destination are members of the two leftmost racks. This saturates the incoming links to the two leftmost ToR switches, and thus will affect destinations on all machines within a rack, together – again, with the rest of the data-center unaffected. This scenario is studied in section III-C.

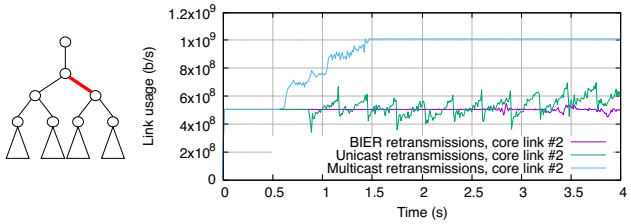
Bursty, non-localised losses, where losses are not related to localised background traffic, but are produced by a loss model in each individual link. This scenario is studied in section III-D.

A. Simulation Parameters and Setup

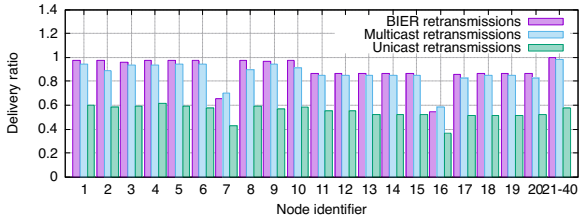
The B.I.E.R. shim layer, described in section II, has been implemented in NS-3 [23], with the bit#→IP-address mapping assumed *a priori* available in all routers, as discussed in section I-A. UDP is used as transport protocol, and B.I.E.R. and *reliable B.I.E.R.* headers are implemented as IPv6 extension headers. The *reliable B.I.E.R.* parameters from section II are chosen as follows: $\Delta t_{agg} = 7ms$ (NACK aggregation delay), $\Delta t_{retry} = 15ms$ (NACK retransmission timer), and $l = 3$ (retransmission limit). All links are homogeneous, point-to-point, with 1 Gbps capacity, with an MTU of 1500 octets and a prop-



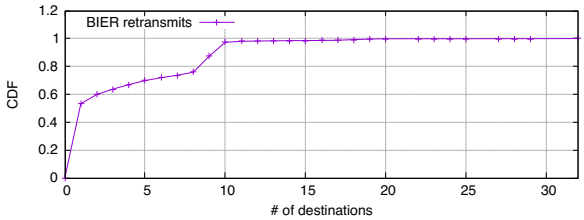
(a) Link usage, between core router and first aggregation router.



(b) Link usage, between core router and second aggregation router.



(c) Delivery ratio for the multicast flow.



(d) Cumulative Distribution Function (CDF) for the number of clients in B.I.E.R. retransmit bitstrings.

Fig. 5. Correlated localised losses experiment. B.I.E.R. retransmits vs unicast and multicast retransmits.

agation delay of 1 μs. Network interfaces all have tail-drop queues of size 512 packets. The links themselves are lossless – except for the simulations in section III-D, which considers link-losses according to a Gilbert-Elliot loss model [24]. For these simulations, the multicast data packet source is attached to the core router, which generates a constant bit-rate *reliable B.I.E.R.* flow of 500 Mbps. All 40 machines are destinations for this flow.

This scenario can be considered to represent *e.g.,* broadcasting of a live media, where a constant transmission bitrate has to be sustained, and where a retransmitted multicast data packet is of no value if received "too late". Thus, some packets may not be received by all destinations, and the ratio of packets successfully received after retransmissions (the *delivery ratio*) will be a metric of interest – as will the network load of the different links in the network, as well as the sum of traffic over all links in the network (the *traffic footprint*).

When unicast background traffic is introduced in the network (for the simulations in section III-B and III-C), it takes the form of 19 UDP flows of a constant bit-rate of 500 Mbps. Each flow has a randomly selected source and destination. These flows are injected into the network in a staggered fashion, starting every 200 ms, and each lasting until the end of the simulation. The simulations in section III-B and III-C differ in the domain from which the (source, destination) pairs are randomly chosen.

As a reference, *reliable B.I.E.R.* (*i.e.,* using B.I.E.R. for retransmissions, as per this paper) is compared with multicast (as in *e.g.,* [6]) and unicast (as in *e.g.,* [7]) retransmissions of NACKed multicast data packets.

B. Uncorrelated, Localised Losses

For this set of simulations, UDP background flows are introduced with sources and destinations both within the leftmost rack (figure 4) as described in section III-A. This will saturate some of the links between the ToR router and the individual machines, leading to packet losses in the "downwards" interfaces of the leftmost ToR router. B.I.E.R. aggregation will thus only happen when, by chance, two or more clients detect a packet loss (and thus generates NACKs) at the same time.

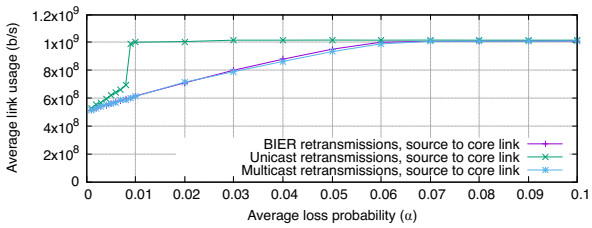
Figure 3 depicts the results of a 4-second simulation run, specifically the usage of the two core links, the delivery ratio, and the distribution of the number of clients in B.I.E.R. retransmissions. A first observation from figure 3b is that, with multicast retransmissions, the rightmost aggregation link carries unnecessary traffic, unlike unicast and B.I.E.R. retransmissions. It can also be observed that multicast retransmissions saturate the core links faster than the two other mechanisms: this is because excess retransmissions produce additional congestion, leading to additional losses of original transmissions, in turn leading to additional retransmissions.

Comparing with unicast retransmissions, the use of B.I.E.R. retransmissions further minimises the traffic footprint: even with uncorrelated losses, when using B.I.E.R., retransmissions are aggregated when several clients do not receive the same packet. This is illustrated in figure 3d, which depicts the Cumulative Distribution Function (CDF) of the number of simultaneous clients to which a B.I.E.R. retransmission is performed: $\sim 10\%$ of B.I.E.R. retransmissions are destined for multiple (≥ 2) destinations, and thus benefit from aggregation. This allows a further reduction of link usage, as depicted in figure 3b.

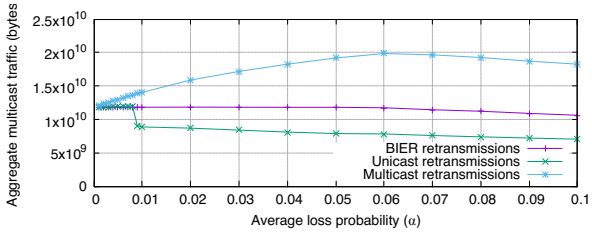
A conclusion to draw from these simulations is that, in case of localised losses, B.I.E.R. and unicast retransmissions are preferable to multicast retransmissions – due to the latter incurring unnecessary traffic on links in paths unaffected by losses. Another conclusion is, that when multiple destinations do not receive a given multicast data packet, B.I.E.R. retransmissions allow aggregation – an advantage over unicast retransmissions.

C. Correlated, Localised Losses

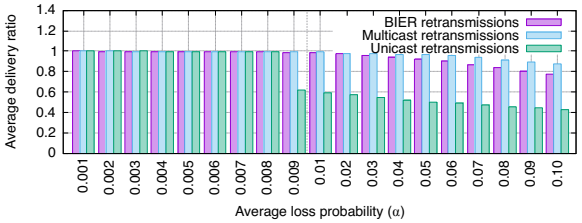
For this set of simulations, UDP background flows are introduced with sources and destinations within the **two** leftmost racks (figure 4) as described in section III-A. This will, again, saturate some of the links – this time, in addition to between the individual machines and the ToR routers, also between the ToR



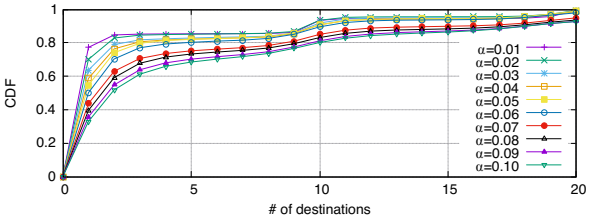
(a) Link usage, between the source and the core router (averaged over all the duration of a simulation).



(b) Aggregate multicast traffic (over all links and for all the duration of a simulation).

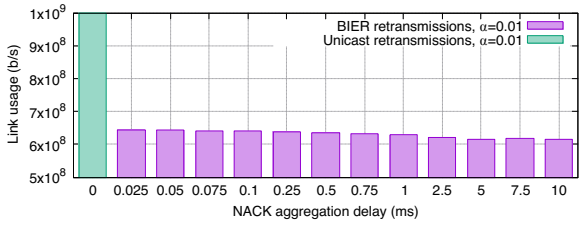


(c) Average delivery ratio for the multicast flow.

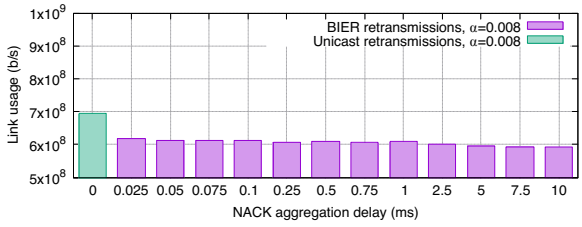


(d) Cumulative Distribution Function (CDF) for the number of clients in B.I.E.R. retransmit bitstrings.

Fig. 6. Unlocalised bursty losses experiments. 19 simulations with different loss probabilities.



(a) $\alpha = 1\%$



(b) $\alpha = 0.8\%$

Fig. 7. Influence of Δt_{agg}: link usage between the source and the core router.

routers and aggregation routers. A loss on one of these links will affect all the destinations within a rack.

Figure 5 depicts the results of a 4-second simulation. For the same reasons as in section III-B, multicast retransmissions cause unnecessary traffic on the right "half" of the data-center (see figure 4) – and causes earlier saturation on the left core link.

A single lost multicast data packet will, in this scenario, typically fail to be received by several destinations, therefore unicast retransmissions will generate a larger traffic footprint as compared to B.I.E.R. retransmissions. With a link capacity of 1 Gbps and a multicast flow of 0.5 Gbps, the link between the source and the core router can sustain the unicast retransmission load only when each multicast data packet is, on average, retransmitted no more than once. Beyond that, the link becomes saturated with retransmissions, thus preventing "legitimate" original transmissions to succeed. This explains why unicast recovery incurs a lower delivery ratio, as depicted in figure 5c. This is also why the link usage on the first core link is lower for unicast retransmissions than for B.I.E.R. retransmissions, as depicted in figure 5a.

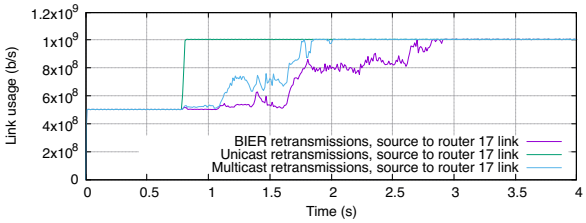
The distribution of the number of simultaneous clients to which a multicast data packet is retransmitted is depicted in figure 5d: $\sim 46\%$ of B.I.E.R. retransmissions are destined for multiple destinations, and thus benefit from aggregation.

D. Unlocalised, Bursty Losses

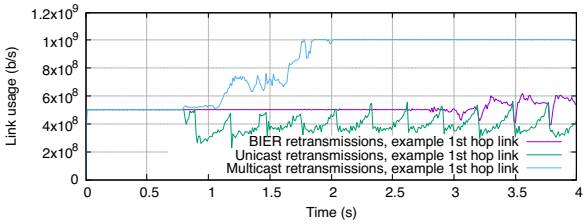
The simulations on sections III-B and III-C illustrate the benefits of B.I.E.R. retransmissions when losses are spatially localised. Instead of creating background UDP flows, this section assumes uncontrolled, exogenous congestion in the data center – modelled by a Gilbert-Elliott loss model [24] on all links. This model is used to model bursty transmissions, and [26] shows that it accurately describes packet losses in the Internet. In sum, the Gilbert-Elliott loss model prescribes that a link can be in either a *good* or *bad* state. In *good* state, the link is ideal (no losses), whereas in *bad* state, the probability of a transmission to be successful (*i.e.,* to not be lost) is h. For each packet to be transmitted over a link, the state of the link may change: from *bad* to *good* with a probability of r, and from *good* to *bad* with a probability p.

For the purpose of the simulations in this section, the success probability in the *bad* state is set to $h = 0.5$, and the transition probability from *bad* to *good* to $r = 0.01$ (*i.e.,* the expected loss burst duration is 100 packets). The transition probability from *good* to *bad*, p, is set so that the average packet loss ratio is α, according to equation (4) in [26]: $p = \frac{r\alpha}{1-h-\alpha} = \frac{0.01\alpha}{0.5-\alpha}$. In order to quantify the sensitivity of the system to different congestion levels, 19 simulations are run, for $\alpha \in \{0.001, 0.002, \ldots, 0.01, 0.02, \ldots, 0.1\}$.

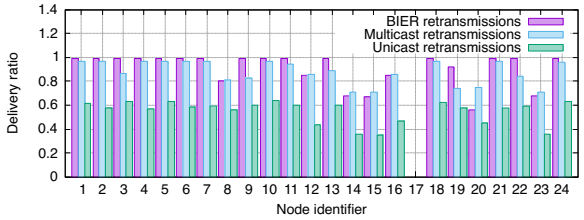
Figure 6 depicts the simulation results. A stability limit ($\alpha = 6\%$ for B.I.E.R. and multicast, $\alpha = 0.9\%$ for unicast) can be observed in figure 6a. Above this limit, retransmissions



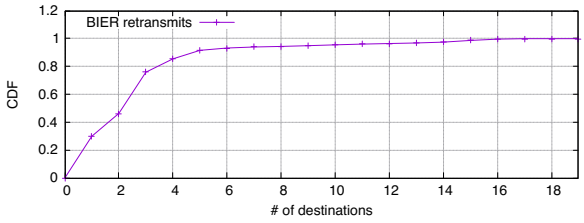
(a) Link usage, between the source and router 17 (London).



(b) Link usage, between router 17 (London) and router 5.



(c) Delivery ratio for the multicast flow. Node 17 is the source.



(d) Cumulative Distribution Function (CDF) for the number of clients in B.I.E.R. retransmit bitstrings.

Fig. 8. ISP topology experiment. B.I.E.R. retransmits vs unicast and multicast retransmits.

compete with original transmissions over the link between the source and the core router, causing some of these original transmissions to be lost, requiring additional retransmissions. This, then, causes the delivery ratio to deteriorate, as depicted in figure 6c. Thus, for this stability metric, *reliable B.I.E.R.* shows superior results as compared to unicast retransmissions, while behaving similarly as compared to multicast retransmissions. Figure 6b shows the aggregate traffic (the sum of traffic induced by transmissions and retransmissions over all links) for each of the values of α. While unicast retransmissions (in the stability zone) behave slightly worse than B.I.E.R. retransmissions, multicast retransmissions incur a substantial traffic footprint (of approximately $1.7\times$ that of B.I.E.R. retransmissions, for $\alpha = 6\%$).

A conclusion to draw from these simulations is that, when losses are unlocalised and bursty, B.I.E.R. retransmissions are vastly preferable to multicast retransmissions, in terms of global traffic footprint and also vastly preferable to unicast retransmissions, in terms of avoiding saturation of individual links.

Finally, the CDF of the number of simultaneous clients to which a multicast data packet is retransmitted is depicted in figure 6d, which shows that when $\alpha \geq 6\%$, more than 50% of B.I.E.R. retransmissions are destined for multiple destinations, and thus benefit from aggregation.

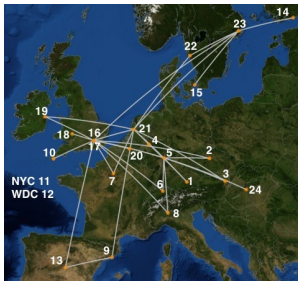


Fig. 9. Network topology (picture from [25]).

E. Influence of the Aggregation Timer

As described in section II-C, increasing the aggregation timer Δt_{agg} (*i.e.,* the amount of time during which the source collects NACKs for a given packet before retransmitting) directly reduces the induced network traffic (since aggregating more NACKs means that the corresponding retransmission is sent to more clients), at the cost of packets being delivered later to the application. In order to quantify this phenomenon, an experiment using the scenario of section III-D is conducted. For two target loss probabilities $\alpha = 1\%$ and $\alpha = 0.8\%$ (slightly above and below the stability limit observed in figure 6a, respectively), different values for the parameter Δt_{agg} are used, ranging from 25 μs to 10 ms.

Figure 7a depicts the usage of the link between the source and the core router, averaged above the duration of the simulation, with $\alpha = 1\%$. It is interesting to observe that enabling NACK aggregation (*i.e.,* making $\Delta t_{agg} \neq 0$) causes a substantial improvement in performance. In particular, the use of any non-zero NACK aggregation delay leads to a significant benefit in terms of link usage: the smallest non-zero explored value ($\Delta t_{agg} = 25\ \mu s$) makes the link usage drop from 1 Gbps (with $\Delta t_{agg} = 0$) to 645 Mbps. Further increases of the aggregation delay lead to minor reductions of link usage (*e.g.,* from 645 Mbps for $\Delta t_{agg} = 25\ \mu s$, to 615 Mbps for $\Delta t_{agg} = 10\ ms$), at the cost of linearly increasing the delay of retransmitted packets by Δt_{agg}. Figure 7b depicts the results for $\alpha = 0.8\%$: a similar pattern can be observed, with a lower amount of traffic for unicast retransmissions – due to α being below the stability limit.

IV. ISP TOPOLOGY SIMULATIONS

Sections III-B, III-C, and III-D illustrated the benefits of *reliable B.I.E.R.* for reducing the traffic footprint in strict tree topologies such as those from data centers – begging the question of if these benefits are dependent on these topologies. In order to answer that question, this section presents simulations of a real (not just realistic) topology, specifically that of BT Eu-

rope (Aug. 2010) from [25] (see figure 9). Note that while the topology used comes from a real deployment, this simulation does not claim to reproduce realistic Internet traffic: the goal is to explore the behaviour of *reliable B.I.E.R.*.

This topology consists of 24 routers, connected by 1 Gbps links. It is assumed that the unicast routing protocol has converged and each router has perfect shortest paths to all other routers. For the purpose of this simulation study, the source node is attached to router 17 (in London), and a destination is attached to each of the other 23 routers. The simulation parameters and multicast traffic flow parameters are as per section III-A. Background traffic flows are also as per section III-A, noting that router 17 is never chosen as source or destination for a background flow.

Figure 8a depicts the link usage between the source and router 17 – revealing that unicast retransmissions rapidly saturate the link, and that multicast retransmissions saturate the link faster than B.I.E.R. retransmissions. Figure 8b depicts the link usage of the link between router 17 (to which the source is attached) and router 5 (one of its directly adjacent peer routers), noting that it is up to twice as high for multicast retransmissions, as it is for both B.I.E.R. retransmissions and for unicast retransmissions. The link load for unicast retransmissions on this link is lower than for *reliable B.I.E.R.*. The reason for this is, that as unicast retransmissions already saturated the link between the source and router 17 (figure 8a), fewer unicast retransmissions make it onto the link between router 17 and router 5 (figure 8b), causing an overall lower data delivery ratio when using unicast retransmissions. This is depicted in figure 8c, which also indicates a higher data delivery ratio for B.I.E.R. retransmissions than for multicast retransmissions.

The distribution of the number of simultaneous clients to which a multicast data packet is retransmitted is depicted in figure 8d: $\sim 71\%$ of B.I.E.R. retransmissions are destined for multiple destinations, and thus benefit from aggregation.

The conclusion to draw from these simulations is, that the results obtained with a data-center topology (section III) also can be valid for other topologies.

V. RELIABLE B.I.E.R. PERFORMANCE ANALYSIS

The simulations in sections III and IV illustrate the performance benefits of B.I.E.R.-retransmissions for reliable multicast, both when faced with (i) rare, isolated losses, and with (ii) correlated, frequent losses in traffic-intensive environments. Specifically, *reliable B.I.E.R.* was observed to result in a substantially lower traffic footprint in the simulated scenarios, with equivalent or better multicast data packet delivery ratios than when using multicast and unicast retransmissions.

This section aims at generalising these observations, by way of formulating an analytical model of arbitrary tree topologies – and using this model to, analytically, quantify the number of successful and failed transmissions[6] necessary for a reliable multicast operation to succeed (*i.e.,* for all destinations to have received a copy of a multicast data packet). Section V-B derives an exact expression of this as $M^{\mathrm{B}}_{[i]}$, for *reliable B.I.E.R.* – and for comparison, $M^{\mathrm{m}}_{[i]}$ and $M^{\mathrm{u}}_{[i]}$ for when using multicast and unicast retransmissions, respectively.

These exact expressions, however, become mathematically intractable for large trees, thus section V-C develops a first-order approximation of the average traffic footprints of reliable multicast using B.I.E.R., multicast, and unicast retransmissions, respectively.

A. Model, Assumptions and Definitions

Network links have an associated packet loss probability[7] $\alpha \in [0, 1]$. As illustrated in section 1, multicast transmissions and retransmissions (regardless of if B.I.E.R., multicast, or unicast retransmissions) span a tree, rooted in the source. For describing these trees, the following notation is introduced: routers and destinations are indiscriminately termed *node*, and each node is uniquely labeled, with the path from the root of the tree to it[8]; $[[i], j]$ denotes the j-th child of node $[i]$, and the term *"the subtree* $[i]$*"* refers to the subtree, which is rooted in node $[i]$. Finally, the set of children of $[i]$ is denoted $c([i])$: $c([i]) = \{[[i], 1], [[i], 2], \dots\}$

This analysis assumes retransmissions by the source until all destinations have received a copy of the multicast data packet, and that the source collects all generated NACKs before retransmitting a packet (*i.e.,* $l = \infty$, no NACKs are lost, $\Delta t_{agg} \geq RTT_{max} - RTT_{min}$, $\Delta t_{retry} \geq \Delta t_{agg} + RTT_{max}$). It quantifies, under these assumptions, (1) the number of retransmissions of a multicast data packet that are made by the source, and (2) the total number of transmissions in the network, until all clients have received (at least) one copy of the multicast data packet.

Definitions: Given a node $[i]$ and its child $[[i], j]$, and with reference to figure 11:

- $T_{[[i],j]}$: is the number of *attempts* from node $[[i], j]$, *i.e.,* number of times that $[[i], j]$ must transmit copies of a multicast data packet to its children, to ensure that all destinations in its subtree receive the multicast data packet. If $[[i], j]$ is a leaf, then by convention $T_{[[i],j]} = 1$.
- $X_{[i]\to j}$: is the number of *transmissions* made by $[i]$ over the link $([i], [[i], j])$, needed to ensure that node $[[i], j]$ receives the $T_{[[i],j]}$ copies of the multicast data packet.
- $M^{\star}_{[i]}$: Number of *packets* transmitted inside a subtree $[i]$ to ensure that all destinations receive a copy, where $\star$ indicates the considered variant (B for B.I.E.R., m for multicast, u for unicast). If $[i]$ is a leaf, then by convention $M^{\star}_{[i]} = 0$.

The number of attempts by $[i]$ is the worst of the number of transmissions on all links $([i], [[i], j])$:

$$T_{[i]} = \max_{[[i],j] \in c([i])} X_{[i]\to j} \tag{1}$$

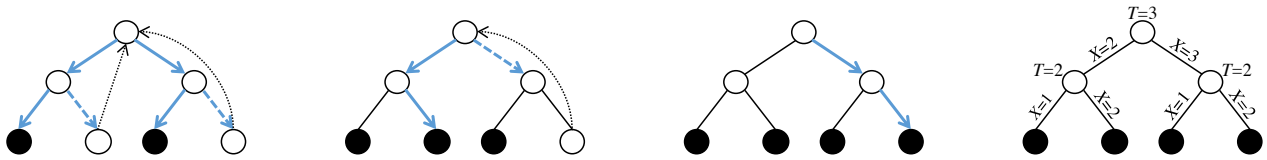
B. Computation of $T_{[i]}$, $X_{[i]\to j}$, and $M^{\star}_{[i]}$

Each node $[[i], j]$ needs to receive $T_{[[i],j]}$ copies of the multicast data packet from its parent, $[i]$. For each of these, the number of transmissions over the link $([i], [[i], j])$ until the copy is

[6] Colloquially speaking, to *count the blue arrows* in figure 1.

[7] For lossless links, operating below capacity and with finite buffers, packet losses are due to buffer overflow – thus while a link may be lossless, an interface may still experience packet losses.

[8] The root is labelled $[1]$; the first child of the root is labelled $[[1], 1]$, its second child $[[1], 2]$; the first child of $[[1], 1]$ is $[[[1], 1], 1]$; etc.



(a) First attempt (b) Second attempt (c) Third attempt (d) Total number of attempts and transmissions

Fig. 10. *Reliable B.I.E.R.* examplified – solid blue arrows represent successful transmissions, dashed blue arrows represent unsuccessful transmissions, and dashed black arrows represent NACKs. A NACK is sent by two nodes (a) upon receipt of a subsequent packet in the stream. After retransmission (b), the last node still has not received a copy of the packet, and sends a second NACK upon timeout of Δt_{retry} (Algorithm 2). Figure (d) shows the total number of *attempts* $T_{[i]}$ at each node $[i]$, and total *transmissions* $X_{[i]\to j}$ at each link $[i] \to [[i], j]$. The total number of packets sent in the tree (the sum of all $X_{[i]\to j}$) is $M^{\mathrm{B}}_{[i]} = 11$.

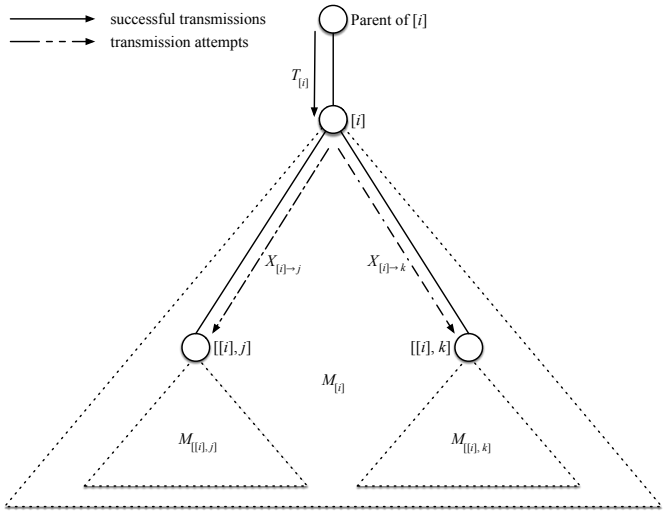


Fig. 11. Notation.

successfully received at $[[i], j]$ is geometrically distributed with success probability $(1-\alpha)$, which leads to the following proposition:

Proposition 1 *The total number of transmissions over the link* $([i], [[i], j])$ *follows a negative binomial distribution with (random) parameter* $T_{[[i],j]}$*. For* $x \geq k \geq 1$*:*

$$Pr(X_{[i]\to j} = x | T_{[[i],j]} = k) = \binom{x-1}{k-1} \alpha^{x-k}(1-\alpha)^k \quad (2)$$

and the number of attempts from node $[i]$*,* $T_{[i]}$ *is:*

$$Pr(T_{[i]} = k) = \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} \leq k) - \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} \leq k-1) \quad (3)$$

Equations (2) and (3) allow computing the probability density function (PDF) for $X_{[i]\to j}$ and $T_{[i]}$ recursively, from the leaves towards the root, using the convention that for a leaf, $T_{[i]} = 1$.

B.1 *B.I.E.R. retransmissions:* $M^B_{[i]}$

When using B.I.E.R. retransmissions, $M^{\mathrm{B}}_{[i]}$ is the sum of the transmissions on each link $([i], [[i], j])$ and the packets transmitted in each subtree $[[i], j]$, *i.e.,* :

$$M^{\mathrm{B}}_{[i]} = \sum_{[[i],j]\in c([i])} (X_{[i]\to j} + M^{\mathrm{B}}_{[[i],j]}) \quad (4)$$

Figure 10 provides a detailed example of a B.I.E.R. reliable transmission, with the corresponding values for $T_{[i]}$, $X_{[i]\to j}$ and $M^{\mathrm{B}}_{[i]}$ displayed in Fig. 10d.

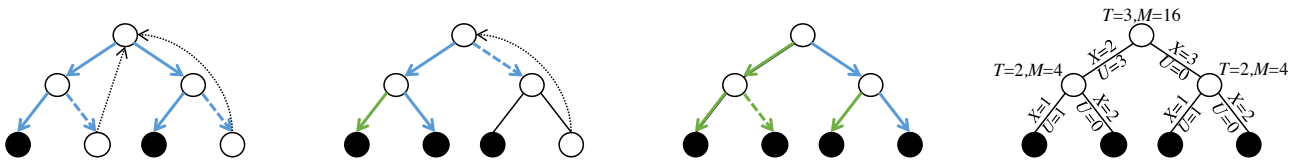
B.2 *Multicast retransmissions:* $M^m_{[i]}$

The number of multicast data packets sent over a network with multicast retransmissions can be obtained by adapting the previously presented model (section V-B.1). Consider the transmission from a node $[i]$ to a node $[[i], j]$: transmitted packets can be classified into two categories: (i) packets sent until the subtree $[[i], j]$ is covered, and (ii) packets flooded by $[i]$ inside the subtree $[[i], j]$ after it has been covered. The latter packets come from retransmissions from the source $[i]$ that are due to other subtrees $[[i], k]$ having not yet been covered. Let $U_{[i]\to[[i],j]}$ be the number of packets that fall into the second category. The number of floods is $T_{[i]} - X_{[i]\to j}$ (*i.e.,* the number of times $[i]$ transmits after $[[i], j]$ has already received enough packets): index these floods with $f \in [1, T_{[i]} - X_{[i]\to j}]$. For each of these floods, let $Y^f_{[i]\to j}$ be a Bernoulli variable of parameter $(1-\alpha)$ representing the success of transmission on the link $[i] \to [[i], j]$, and $F^f_{[[i],j]}$ be a variable representing the number of packets flooded in the subtree $[[i], j]$. The number of unnecessary packets is then, for each of these floods, one packet (from $[i]$ to $[[i], j]$) plus, if the transmission succeeded (*i.e.,* if $Y^f_{[i]\to j} = 1$), the number of packets $F^f_{[[i],j]}$ resulting from a multicast flood sourced at $[[i], j]$:

$$U_{[i]\to[[i],j]} = \sum_{f=1}^{T_{[i]}-X_{[i]\to j}} (1 + Y^f_{[i]\to j} F^f_{[[i],j]}) \quad (5)$$

where the mean number of packets sent in a multicast flood from $[i]$, $\mathbb{E}(F_{[j]})$, can be recursively computed as follows (with $\mathbb{E}(F_{[j]}) = 0$ for every leaf $[j]$):

$$\mathbb{E}(F_{[i]}) = \sum_{[[i],j]\in c([i])} (1 + (1-\alpha)\mathbb{E}(F_{[[i],j]})) \quad (6)$$



(a) First attempt (b) Second attempt (c) Third attempt (d) Total number of attempts and transmissions

Fig. 12. Example of multicast reliable transmission. In addition to the conventions of figure 10, green arrows represent unnecessary retransmissions. In the second attempt (b), node $[[1],1]$ floods one unnecessary packet. In the third attempt (c), the root floods three unnecessary packets, and node $[[1],2]$ floods one unnecessary packet. Figure (d) shows the total number of *attempts* $T_{[i]}$ at each node $[i]$, total *transmissions* $X_{[i]\to j}$ at each link $[i] \to [[i],j]$, and the total number of *unnecessary packets* $U_{[i]\to[[i],j]}$ flooded by each node $[i]$ in the subtree $[[i],j]$. The total number of packets sent in the tree (the sum of all $X_{[i]\to j}$ and $U_{[i]\to[[i],j]}$) is $M^{\mathrm{m}}_{[1]} = 16$.

From this, the total number of packets sent in a subtree $[i]$ until all of its destinations receive a copy, $M^{\mathrm{m}}_{[i]}$, can be computed recursively. It corresponds, for each child $[[i],j]$, to the number of transmissions over the link $[i] \to [[i],j]$ required by $[[i],j]$, plus the number of packets sent inside $[[i],j]$ so as to cover all destinations, plus the unnecessary multicast packets originating from $[i]$:

$$M^{\mathrm{m}}_{[i]} = \sum_{[[i],j]\in c([i])} \left[X_{[i]\to j} + M^{\mathrm{m}}_{[[i],j]} + U_{[i]\to[[i],j]} \right] \quad (7)$$

Proposition 2 indicates a simple way to compute the average traffic footprint for multicast retransmissions, using $T_{[i]}$ and $F_{[i]}$.

Proposition 2 *Let $[i]$ be a node in the tree. With multicast retransmissions, the mean number of packets sent until all destinations in $[i]$ obtain a copy can be computed as such:*

$$\mathbb{E}(M^{m}_{[i]}) = \mathbb{E}(T_{[i]})\mathbb{E}(F_{[i]}) \quad (8)$$

Proof: See appendix -A. □

Figure 12 provides a detailed example of multicast reliable transmission, with corresponding values for $T_{[i]}$, $X_{[i]\to j}$, $U_{[i]\to[[i],j]}$, and $M^{\mathrm{m}}_{[i]}$ variables.

B.3 *Unicast retransmissions:* $M^{u}_{[i]}$

With unicast retransmissions, losses experienced by each destination are treated individually by the source. Given the loss of a multicast data packet (sent by the source $[1]$) at a destination $[c]$, connected to the source in $d([c])$ hops, the number of retransmissions from the source before successful delivery of the packet to $[c]$ is a random variable, $R_{[1]\to[c]}$, whose mean is described in proposition 3.

Proposition 3 *The mean value of $R_{[1]\to[c]}$ is:*

$$\mathbb{E}(R_{[1]\to[c]}) = \frac{1-(1-\alpha)^{d([c])}}{\alpha(1-\alpha)^{d([c])}} \quad (9)$$

Proof: See appendix -B. □

The previous proposition allows to compute the total number of multicast data packets sent from the source with the unicast reliability mechanism, $M^{\mathrm{u}}_{[1]}$. This variable corresponds to the multicast data packets sent in the first multicast flood, plus, for each destination that did not receive a copy of the multicast data packet, the number of unicast retransmissions needed until the copy is successfully received. Proposition 4 expresses $M^{\mathrm{u}}_{[1]}$ and provides a closed expression for its mean, $\mathbb{E}(M^{\mathrm{u}}_{[1]})$.

Proposition 4 *Let $\mathcal{C}$ be the set of destinations, and let $\mathcal{F}_{[1]}$ be the (random) set of destinations that have successfully received a copy after the first multicast flood by $[1]$. Then, the number of multicast data packets sent from the source, under unicast retransmissions, until each destination has received a copy is:*

$$M^{u}_{[1]} = F_{[1]} + \sum_{[c]\in\mathcal{C}} \mathbf{1}_{\{[c]\notin\mathcal{F}_{[1]}\}} R_{[1]\to[c]} \quad (10)$$

and its mean is:

$$\mathbb{E}(M^{u}_{[1]}) = \mathbb{E}(F_{[1]}) + \sum_{[c]\in\mathcal{C}} \frac{(1-(1-\alpha)^{d([c])})^2}{\alpha(1-\alpha)^{d([c])}} \quad (11)$$

Proof: From the definition of $\mathcal{F}_{[1]}$, equation (10) holds. The mean number of multicast data packets sent in the network is therefore:

$$\mathbb{E}(M^{\mathrm{u}}_{[1]}) = \mathbb{E}(F_{[1]}) + \sum_{[c]\in\mathcal{C}} Pr([c]\notin\mathcal{F}_{[1]})\mathbb{E}(R_{[1]\to[c]})$$

And since $Pr([c]\notin\mathcal{F}_{[1]}) = 1-(1-\alpha)^{d([c])}$, the result in (11) is obtained by using equation (9).

□

B.4 *Computational Example*

For trees of small height, it is possible to recursively derive the average number of transmissions needed in the network in order to deliver a multicast data packet to all destinations, in an exact manner, for B.I.E.R. reliability (proposition 1 and equation (4)), multicast reliability (proposition 2) or unicast reliability (proposition 4). As an example, figure 13 reports the results of this computation, for a binary tree of height 2 (as in figure 1): as expected, reliable B.I.E.R. incurs the lowest overhead.

C. *Total Traffic Approximation*

Directly computing the traffic footprint is intractable when the depth of the tree is important. Therefore, this section provides a

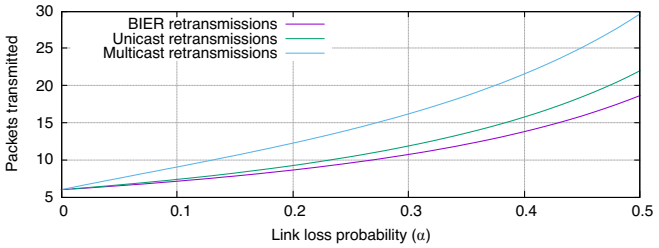


Fig. 13. Number of packets $M^{\mathrm{B}}_{[1]}$ (B.I.E.R. retransmissions), $M^{\mathrm{m}}_{[1]}$ (multicast retransmissions) and $M^{\mathrm{u}}_{[1]}$ (unicast retransmissions) transmitted in the binary tree of figure 1 until all destinations receive a copy.

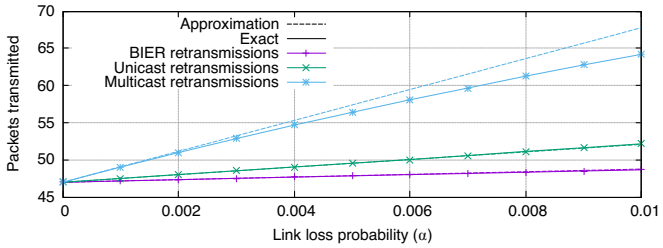


Fig. 14. Number of packets $M^{\mathrm{B}}_{[1]}$ (B.I.E.R. retransmissions), $M^{\mathrm{m}}_{[1]}$ (multicast retransmissions) and $M^{\mathrm{u}}_{[1]}$ (unicast retransmissions) transmitted in the tree of figure 4 until all destinations receive a copy. Solid lines represent exact values (obtained by simulation), dotted lines represent the low-loss approximation given in theorem 1.

first-order approximation of the number of packets transmitted in the network before all destinations receive a copy, under the assumption that losses are rare (*i.e.,* $\alpha \to 0$, as in [18]), for arbitrary trees and for each retransmission mechanism (reliable B.I.E.R., unicast retransmissions, multicast retransmissions).

For a link l of the tree, $d(l)$ is the depth of l ($d(l) = 1$ for a link rooted at the source); L is the total number of links in the tree, and D is the average depth of a links in the tree: $D = \frac{1}{L}\sum_l d(l)$. C is the number of destinations, and Δ is the average squared depth of a destination: $\Delta = \frac{1}{C}\sum_c d(c)^2$. Given these parameters, Theorem 1 describes first-order (for α) approximations of the number of transmissions in the network:

Theorem 1 *The average number of multicast data packets that need to be sent in the tree until all destinations have received a copy are, for reliable B.I.E.R. ($M^{B}_{[1]}$), for multicast retransmissions ($M^{m}_{[1]}$) and for unicast retransmissions ($M^{u}_{[1]}$), given by the following approximation when $\alpha \to 0$:*

$$\begin{cases} \mathbb{E}(M^{B}_{[1]}) & = L + LD\alpha + \mathcal{O}(\alpha^2) \\ \mathbb{E}(M^{m}_{[1]}) & = L + [L^2 - L(D-1)]\alpha + \mathcal{O}(\alpha^2) \\ \mathbb{E}(M^{u}_{[1]}) & = L + [C\Delta - L(D-1)]\alpha + \mathcal{O}(\alpha^2) \end{cases} \quad (12)$$

Proof: See appendix -C. □

When there are no losses ($\alpha = 0$), the transmission of one multicast data packet yields L packets in the tree (one per link), whichever retransmission mechanism (B.I.E.R., multicast or unicast) is used. In addition to these L packets, with B.I.E.R. retransmissions, the traffic is approximately $LD\alpha$ packets, as compared to approximately $L^2\alpha$ packets for multicast retransmissions and approximately $C\Delta\alpha$ for unicast retransmissions (Theorem 1). The traffic due to unicast or multicast retransmissions can thus be orders of magnitudes bigger than the corresponding B.I.E.R. traffic, if the number of links and/or the depth of destinations is important.

D. Discussion

The accuracy and relevance of approximations from theorem 1 can be assessed against simulations in realistic tree topologies. Two examples are examined in this section: (1) reliable multicast over a datacenter-like topology as depicted in figure 4, and (2) multicast flows over fat-tree-like topologies [27].

The datacenter-like topology of figure 4 yields the parameters $L = 47$, $C = 40$, $D = \frac{177}{47}$, and $\Delta = 16$. When $\alpha \to 0$, retransmissions will incur a footprint of 177α packets for B.I.E.R., 510α packets for unicast, and 2079α packets for multicast. In order to quantify the quality of the approximation for this example, the means for $M^{\mathrm{B}}_{[1]}$, $M^{\mathrm{m}}_{[1]}$ and $M^{\mathrm{u}}_{[1]}$ have been computed over 10^6 random samples, for different values of α with $0 \leq \alpha \leq 1\%$. Figure 14 depicts the results of these simulations, as well as the linear approximation from theorem 1. For B.I.E.R. and unicast retransmissions, the approximation accurately fits the computed mean. For multicast retransmissions, the approximation is within a 6% error margin of the computed value.

k	L	$L \times D$	C	B.I.E.R.	unicast	multicast
4	28	68	16	68α	104α	744α
6	78	204	54	204α	360α	5958α
8	168	456	128	456α	864α	27936α
10	310	860	250	860α	1700α	95550α
12	516	1452	432	1452α	2952α	265320α
14	798	2268	686	2268α	4704α	635334α
16	1168	3344	1024	3344α	7040α	1362048α

Table 1. Average number of retransmissions per multicast data packet for $(k, k/2, k/2)$ tree topologies (approximation as per theorem 1)

For reliable multicast flows over fat-tree-like topologies, the root has k children, each having $k/2$ children, each also having $k/2$ children. For these trees, the parameters become: $L = k + \frac{k^2}{2} + \frac{k^3}{4}$, $L \times D = k + k^2 + \frac{3k^3}{4}$, $C = \frac{k^3}{4}$, $\Delta = 9$, allowing calculating the approximation of theorem 1. Table 1 depicts the approximate retransmission footprint for B.I.E.R., multicast and unicast retransmissions. It can be observed that unicast retransmissions exhibit a footprint approximately twice as high as B.I.E.R. retransmissions. The footprint for multicast retransmissions is at least one order of magnitude higher, and clearly does not scale with the number of clients.

VI. CONCLUSION

This paper has proposed a scalable network service offering efficient and reliable multicast. NACK-based, this network service uses B.I.E.R. (Bit-Indexed Explicit Replication) for ensuring that traffic (original transmissions and retransmissions, both) are forwarded over a minimal shortest path tree, requiring maintenance of neither per-flow nor per-group state by intermediate routers: the source will encode, for each (re)transmission of a

multicast data packet, the precise destination set – be that every member of a given group, or those members having issued a NACK to request retransmission.

The performance of this network service is compared with "classic" reliable multicast mechanisms, where retransmissions are either unicast (to all destinations having sent a NACK, only) or multicast to all destinations in a given group (when a NACK for a multicast data packet was received from any destination).

Simulation studies in both data-center-like and in Internet-like topologies, and when faced with different loss models, show that the proposed B.I.E.R.-based reliable multicast network service is able to achieve reliability, while overcoming the two main shortcomings of these reference mechanisms: (i) contrary to multicast reliability, links not concerned by losses are not affected by retransmissions and (ii) contrary to unicast reliability, links concerned by losses do not unnecessarily carry multiple copies of the same packets.

Generalising from the simulation studies, an analytical model is presented, which quantifies the retransmission footprint incurred by the three mechanisms in *any* topology – and which shows that the B.I.E.R.-based reliable multicast network service incurs a consistently lower overhead.

ACKNOWLEDGEMENTS

The authors are grateful towards Mohammed Hawari for reviewing this paper, towards Pierre Pfister for insightful discussions and comments, and towards the reviewers whose comments greatly helped improve the quality of this manuscript.

REFERENCES

[1] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the ip multicast service and architecture," *IEEE network*, vol. 14, no. 1, pp. 78–88, 2000.

[2] H. Eriksson, "Mbone: The multicast backbone," *Communications of the ACM*, vol. 37, no. 8, pp. 54–61, 1994.

[3] J. Nicholas, A. Adams, and W. Siadak, "Protocol Independent Multicast - Dense Mode (PIM-DM): Protocol Specification (Revised)," RFC 3973, 2005. [Online]. Available: https://rfc-editor.org/rfc/rfc3973.txt

[4] I. Wijnands, E. C. Rosen, S. Aldrin, T. Przygienda, and A. Dolganow, "Multicast using Bit Index Explicit Replication," Internet Engineering Task Force, Internet-Draft draft-ietf-bier-architecture-05, work in Progress. [Online]. Available: https://tools.ietf.org/html/draft-ietf-bier-architecture-05

[5] W. Braun, M. Albert, T. Eckert, and M. Menth, "Performance comparison of resilience mechanisms for stateless multicast using bier," in *IFIP/IEEE Symposium on Integrated Management (IM)*. IEEE, 2017.

[6] B. Adamson and J. P. Macker, "Reliable messaging for tactical group communication," in *Military Communications Conference, 2010-MILCOM 2010*. IEEE, 2010, pp. 1899–1904.

[7] S. Paul, K. K. Sabnani, J. C.-H. Lin, and S. Bhattacharyya, "Reliable multicast protocol (rmtp)," *IEEE Journal on Selected Areas of Communications*, vol. 15, no. 3, pp. 407–421, 1997.

[8] A. Popescu, D. Constantinescu, D. Erman, and D. Ilie, "A survey of reliable multicast communication," in *Next Generation Internet Networks, 3rd EuroNGI Conference on*. IEEE, 2007, pp. 111–118.

[9] H. W. Holbrook, S. K. Singhal, and D. R. Cheriton, "Log based receiver reliable multicast for distributed interactive simulation," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 328–341, October 1995.

[10] S. Floyd, V. Jacobson, S. McCanne, C.-G. Liu, and L. Zhang, "A reliable multicast framework for light-weight sessions and application level framing," *ACM SIGCOMM Computer Communication Review*, vol. 25, no. 4, pp. 342–356, 1995.

[11] R. Yavatkar, J. Griffoen, and M. Sudan, "A reliable dissemination protocol for interactive collaborative applications," in *Proceedings of the third ACM international conference on Multimedia*. ACM, 1995, pp. 333–344.

[12] C. Bormann, M. J. Handley, and B. Adamson, "NACK-Oriented Reliable Multicast (NORM) Transport Protocol," RFC 5740, 2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5740.txt

[13] J. P. Macker and R. B. Adamson, "A tcp friendly, rate-based mechanism for nack-oriented reliable multicast congestion control," in *Global Telecommunications Conference, 2001. GLOBECOM'01. IEEE*, vol. 3. IEEE, 2001, pp. 1620–1625.

[14] J. Gemmell, T. Montgomery, T. Speakman, and J. Crowcroft, "The pgm reliable multicast protocol," *IEEE network*, vol. 17, no. 1, pp. 16–22, 2003.

[15] D. Li, M. Xu, Y. Liu, X. Xie, Y. Cui, J. W. Wang, and G. Chen, "Reliable multicast in data center networks," *IEEE Transactions on Computers*, vol. 63, no. 8, Aug. 2014.

[16] P. Bhagwat, P. P. Mishra, and S. K. Tripathi, "Effect of topology in performance of reliable multicast communication," in *Proc. INFOCOM'94*, 1994.

[17] J. Nonnenmacher and E. W. Biersack, "Reliable multicast: Where to use fec," in *Protocols for High-Speed Networks V*, W. Dabbous and C. Diot, Eds. Springer US, 1997, ch. 4, pp. 134–148.

[18] ——, "Performance modelling of reliable multicast transmission," in *Proc. INFOCOM'97*, 1997.

[19] F. Baccelli, A. Chaintreau, Z. Liu, and A. Riabov, "The one-to-manv tcp overlay: A scalable and reliable multicast architecture," in *INFOCOM'2005*. IEEE, 2005.

[20] D. Basin, K. Birman, I. Keidar, and Y. Vigfusson, "Source of instability in data center multicast," in *LADIR'10*. ACM, 2010.

[21] R. Hamilton, J. Iyengar, I. Swett, and A. Wilk, "QUIC: A UDP-Based Secure and Reliable Transport for HTTP/2," Internet Engineering Task Force, Internet-Draft draft-hamilton-early-deployment-quic-00, 2016, work in Progress. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-hamilton-early-deployment-quic-00

[22] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, P. Crowley, C. Papadopoulos, L. Wang, B. Zhang *et al.*, "Named data networking," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 3, pp. 66–73, 2014.

[23] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *Modeling and tools for network simulation*, pp. 15–34, 2010.

[24] E. O. Elliott, "Estimates of error rates for codes on burst-noise channels," *The Bell System Technical Journal*, vol. 42, no. 5, pp. 1977–1997, 1963.

[25] "Internet Topology Zoo," Feb. 2017. [Online]. Available: http://www.topology-zoo.org

[26] G. Hasslinger and O. Hohlfeld, "The gilbert-elliott model for packet loss in real time services on the internet," in *Proc. 14th GI/ITG Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems (MMB 2008)*, Mar. 2008.

[27] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 63–74.

Appendix

A. Comparison with Multicast Retransmissions

Proof: [Proof of proposition 2] by induction. If $[i]$ is a leaf, $M^{\mathrm{m}}_{[i]} = 0$ and $F_{[i]} = 0$, hence the result holds. Otherwise, let $[i]$ be a node that is not a leaf, and assume that the result holds for all children of $[i]$. Then, using equation (7), and Wald's equation to expand $\mathbb{E}(U_{[i]\to[[i],j]})$ from equation (5), it follows that:

$$\begin{aligned}
\mathbb{E}(M^{\mathrm{m}}_{[i]}) &= \sum_{[[i],j]\in c([i])} \Big[\mathbb{E}(X_{[i]\to j}) + \mathbb{E}(M^{\mathrm{m}}_{[[i],j]}) \\
&\qquad + \mathbb{E}(T_{[i]} - X_{[i]\to j})(1 + (1-\alpha)\mathbb{E}(F_{[[i],j]}))\Big] \\
&= \sum_{[[i],j]\in c([i])} \Big[\mathbb{E}(X_{[i]\to j}) + \mathbb{E}(T_{[[i],j]})\mathbb{E}(F_{[[i],j]}) \\
&\qquad + \mathbb{E}(T_{[i]} - X_{[i]\to j})(1 + (1-\alpha)\mathbb{E}(F_{[[i],j]}))\Big] \\
&= \sum_{[[i],j]\in c([i])} \Big[\mathbb{E}(X_{[i]\to j}) + (1-\alpha)\mathbb{E}(X_{[i]\to j})\mathbb{E}(F_{[[i],j]}) \\
&\qquad + \mathbb{E}(T_{[i]} - X_{[i]\to j})(1 + (1-\alpha)\mathbb{E}(F_{[[i],j]}))\Big]
\end{aligned}$$

$$= \sum_{[[i],j]\in c([i])} \mathbb{E}(T_{[i]})(1+(1-\alpha)\mathbb{E}(F_{[[i],j]}))$$
$$= \mathbb{E}(T_{[i]})\mathbb{E}(F_{[i]})$$

where the equality $\mathbb{E}(X_{[i]\to j}) = \frac{\mathbb{E}(T_{[[i],j]})}{1-\alpha}$ and equation (6) were used. □

B. Comparison with Unicast Retransmissions

Proof: [Proof of proposition 3] Let $[c]$ be a destination at depth $d([c])$; for simplicity, write $d = d([c])$. The unicast retransmission will succeed if the packet successfully traverses d successive links: the probability of a unicast success from the source to $[c]$ is therefore $(1-\alpha)^d$. Let $Z_{[1]\to[c]}$ represent the number of trials before (and not counting) the unicast success. $Z_{[1]\to[c]}$ is geometrically distributed with parameter $(1-\alpha)^d$:

$$Pr(Z_{[1]\to[c]} = k) = (1-\alpha)^d[1-(1-\alpha)^d]^k, \forall k \geq 0$$

$$\mathbb{E}(Z_{[1]\to[c]}) = \frac{1-(1-\alpha)^d}{(1-\alpha)^d}$$

For each of these first $Z_{[1]\to[c]}$ (unsuccessful) attempts, $N_{[1]\to[c]}$ unicast packets will be sent through the chain of links from $[1]$ to $[c]$, where $N_{[1]\to[c]}$ is distributed as:

$$Pr(N_{[1]\to[c]} = k) = \frac{(1-\alpha)^{k-1}\alpha}{1-(1-\alpha)^d}, \forall 1 \leq k \leq d$$

$$\mathbb{E}(N_{[1]\to[c]}) = \frac{1-\alpha d(1-\alpha)^d-(1-\alpha)^d}{\alpha(1-(1-\alpha)^d)}$$

The last (successful) unicast attempt will generate d packets (one per link). Hence, the total number of unicast packets sent until the destination $[c]$ receives a copy, $R_{[1]\to[c]}$, is:

$$\mathbb{E}(R_{[1]\to[c]}) = \mathbb{E}(Z_{[1]\to[c]})\mathbb{E}(N_{[1]\to[c]}) + d = \frac{1-(1-\alpha)^d}{\alpha(1-\alpha)^d}$$

□

C. Low-loss Limit

This appendix provides a proof of theorem 1; for this, three lemmas will be needed. Lemma 1 first gives an approximation at order 1 in α of the probabilities of having one or two transmissions. Then, lemma 2 gives a bound on the corresponding probability distributions, which will be used in lemma 3 to show that terms corresponding to three or more transmissions do not contribute to the terms of order 1 in α.

For an arbitrary node $[i]$ in the tree, let $l([i])$ be the number of links in the subtree rooted at $[i]$, with $l([i]) = 0$ if $[i]$ is a leaf.

Lemma 1 *Let $[i]$ be a node in the tree, and (if $[i]$ is not a leaf) $[[i],j]$ an arbitrary child of $[i]$. The following approximations hold when $\alpha \to 0$:*

$$\begin{aligned} Pr(X_{[i]\to j} = 1) &= 1-(1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2) \\ Pr(X_{[i]\to j} = 2) &= (1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2) \\ Pr(T_{[i]} = 1) &= 1-l([i])\alpha + \mathcal{O}(\alpha^2) \\ Pr(T_{[i]} = 2) &= l([i])\alpha + \mathcal{O}(\alpha^2) \end{aligned}$$

Proof: by induction over the structure of the tree. If $[i]$ is a leaf, then $T_{[i]} = 1$ by definition (a client needs one copy of the packet). Otherwise, let $[i]$ be a node that is not a leaf, and assume that the result holds for all children of $[i]$. Let $[[i],j]$ be an arbitrary child of $[i]$. Equation (2) yields:

$$\begin{aligned} Pr(X_{[i]\to j} = 1) &= Pr(T_{[[i],j]} = 1)(1-\alpha) \\ &= (1-l([[i],j])\alpha + \mathcal{O}(\alpha^2))(1-\alpha) \\ &= 1-(1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2) \\ Pr(X_{[i]\to j} = 2) &= Pr(T_{[[i],j]} = 1)\alpha(1-\alpha) \\ &\quad + Pr(T_{[[i],j]} = 2)(1-\alpha)^2 \\ &= (1-l([[i],j])\alpha + \mathcal{O}(\alpha^2))\alpha(1-\alpha) \\ &\quad + (l([[i],j])\alpha + \mathcal{O}(\alpha^2))(1-\alpha)^2 \\ &= (1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2) \end{aligned}$$

Then, the definition of $T_{[i]}$ gives:

$$\begin{aligned} Pr(T_{[i]} = 1) &= \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} = 1) \\ &= \prod_{[[i],j]\in c([i])} \left[1-(1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2)\right] \\ &= 1 - \sum_{[[i],j]\in c([i])} \left[1+l([[i],j])\right]\alpha + \mathcal{O}(\alpha^2) \\ &= 1-l([i])\alpha + \mathcal{O}(\alpha^2) \end{aligned}$$

$$\begin{aligned} Pr(T_{[i]} = 2) &= \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} \leq 2) \\ &\quad - \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} \leq 1) \\ &= \prod_{[[i],j]\in c([i])} (1+\mathcal{O}(\alpha^2)) \\ &\quad + \prod_{[[i],j]\in c([i])} \left[1-(1+l([[i],j]))\alpha + \mathcal{O}(\alpha^2)\right] \\ &= \sum_{[[i],j]\in c([i])} \left[1+l([[i],j])\right]\alpha + \mathcal{O}(\alpha^2) \\ &= l([i])\alpha + \mathcal{O}(\alpha^2) \end{aligned}$$

□

The following lemma provides a geometric bound on the distribution of $X_{[i]\to j}$ and $T_{[i]}$ variables, and will be useful to then bound their expectations.

Lemma 2 *Let $[i]$ be a node in the tree, and (if $[i]$ is not a leaf) $[[i],j]$ an arbitrary child of $[i]$. There exist positive constants $A_{[[i],j]}, B_{[[i],j]}, C_{[i]}, D_{[i]}$ such that, for all $\alpha \in [0,1)$:*

$$\begin{aligned} Pr(X_{[i]\to j} = x) &\leq A_{[[i],j]}(B_{[[i],j]}\alpha)^{x-1}, \forall x \geq 1 \\ Pr(T_{[i]} = k) &\leq C_{[i]}(D_{[i]}\alpha)^{k-1}, \forall k \geq 1 \end{aligned}$$

Proof: by induction over the structure of the tree. If $[i]$ is a leaf, then $T_{[i]} = 1$ and the result holds with $C_{[i]} = 1, D_{[i]} = 1$. Otherwise, assume that $[i]$ is not a leaf, and that the result holds for all children of $[i]$. Let $[[i], j]$ be an arbitrary child of $[i]$, and $x \geq 1$. Using the induction hypothesis, and the fact that $\alpha \leq 1$:

$$\begin{aligned}
Pr(X_{[i]\to j} = x) &= \sum_{k=1}^{x} Pr(T_{[[i],j]} = k)\binom{x-1}{k-1}\alpha^{x-k}(1-\alpha)^k \\
&\leq \sum_{k=1}^{x} C_{[[i],j]}(D_{[[i],j]}\alpha)^{k-1}\binom{x-1}{k-1}\alpha^{x-k}(1-\alpha)^k \\
&= C_{[[i],j]}\alpha^{x-1}\sum_{k=1}^{x}(D_{[[i],j]})^{k-1}\binom{x-1}{k-1}(1-\alpha)^k \\
&= C_{[[i],j]}\alpha^{x-1}(1-\alpha)[1+D_{[[i],j]}(1-\alpha)]^{x-1} \\
&\leq C_{[[i],j]}\alpha^{x-1}[1+D_{[[i],j]}]^{x-1}
\end{aligned}$$

The result for $X_{[i]\to j}$ follows, with $A_{[[i],j]} = C_{[[i],j]}$ and $B_{[[i],j]} = 1 + D_{[[i],j]}$.

The result for $T_{[i]}$ remains to be proven. For $t \geq 1$:

$$\begin{aligned}
Pr(T_{[i]} = k) = &\prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} \leq k) \\
&- \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} < k) \\
= &\prod_{[[i],j]\in c([i])} [Pr(X_{[i]\to j} < k) + Pr(X_{[i]\to j} = k)] \\
&- \prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} < k)
\end{aligned}$$

When developing the first product, a term $\prod_{[[i],j]\in c([i])} Pr(X_{[i]\to j} < k)$ appears, which cancels out with the second product. Remaining terms in the first product are indexed with σ. These terms contain one or more factors of the form $Pr(X_{[i]\to j} = k)$ where $[[i], j]$ is a child of $[i]$, and other factors of the form $Pr(X_{[[i],j']} < k)$. Let $j(\sigma)$ be one of the j such that $Pr(X_{[[i],j(\sigma)]} = k)$ appears in the term. An upper-bound for the other factors is 1, effectively keeping only the contribution of $Pr(X_{[[i],j(\sigma)]} = k)$:

$$\begin{aligned}
Pr(T_{[i]} = t) &\leq \sum_{\sigma} Pr(X_{[[i],j(\sigma)]} = k) \times 1 \\
&\leq \sum_{\sigma} A_{[[i],j(\sigma)]}(B_{[[i],j(\sigma)]}\alpha)^{k-1}
\end{aligned}$$

The result for $T_{[i]}$ follows, with $C_{[i]} = \sum_{\sigma} A_{[[i],j(\sigma)]}$ and $D_{[i]} = \max_{\sigma} B_{[[i],j(\sigma)]}$. □

Lemma 3 provides an approximation of the expectations of $X_{[i]\to j}$ and $T_{[i]}$ variables, at order 1 in α, using lemmas 1 and 2.

Lemma 3 *Let $[i]$ be a node in the tree, and (if $[i]$ is not a leaf) $[[i], j]$ an arbitrary child of $[i]$. The following approximations hold when $\alpha \to 0$:*

$$\begin{aligned}
\mathbb{E}(X_{[i]\to j}) &= 1 + (1 + l([[i], j]))\alpha + \mathcal{O}(\alpha^2) \\
\mathbb{E}(T_{[i]}) &= 1 + l([i])\alpha + \mathcal{O}(\alpha^2)
\end{aligned}$$

Proof: First, it will be shown that $\sum_{x=3}^{+\infty} xPr(X_{[i]\to j} = x) = \mathcal{O}(\alpha^2)$. By summing the inequalities in lemma 2, and provided that α is small enough ($\alpha < 1/B_{[[i],j]}$), it is possible to write:

$$\begin{aligned}
\sum_{x=3}^{+\infty} xPr(X_{[i]\to j} = x) &\leq A_{[[i],j]}B_{[[i],j]}^2 \sum_{x=0}^{+\infty}(x+3)(B_{[[i],j]}\alpha)^x\alpha^2 \\
&= A_{[[i],j]}B_{[[i],j]}^2 \frac{3 - 2B_{[[i],j]}\alpha}{(1 - B_{[[i],j]}\alpha)^2}\alpha^2 \\
&= \mathcal{O}(\alpha^2)
\end{aligned}$$

Then, using lemma 1, $\mathbb{E}(X_{[i]\to j})$ can be approximated as:

$$\begin{aligned}
\mathbb{E}(X_{[i]\to j}) &= Pr(X_{[i]\to j} = 1) + 2Pr(X_{[i]\to j} = 2) \\
&\quad + \sum_{x=3}^{+\infty} xPr(X_{[i]\to j} = x) \\
&= 1 - (1 + l([[i], j]))\alpha + 2(1 + l([[i], j]))\alpha + \mathcal{O}(\alpha^2) \\
&= (1 + l([[i], j]))\alpha + \mathcal{O}(\alpha^2)
\end{aligned}$$

which concludes the proof for $\mathbb{E}(X_{[i]\to j})$. The proof for $\mathbb{E}(T_{[i]})$ is similar. □

This allows proving theorem 1 for B.I.E.R. reliability. In the following, $D([i])$ denotes the sum of depth of links in the tree rooted at $[i]$: $D([i]) = \sum_{l\in[i]} d(l)$.

Theorem (Traffic footprint for B.I.E.R.) *Let $[i]$ be a node in the tree. The following approximation holds when $\alpha \to 0$:*

$$\mathbb{E}(M_{[i]}^B) = l([i]) + D([i])\alpha + \mathcal{O}(\alpha^2)$$

Proof: by induction over the structure of the tree. The results holds for leaves, because $M_{[i]}^{\mathrm{B}} = l([i]) = D([i]) = 0$ by definition. Otherwise, let $[i]$ be a node that is not a leaf, and assume that the result holds for the children of $[i]$. Then:

$$\begin{aligned}
\mathbb{E}(M_{[i]}^{\mathrm{B}}) &= \sum_{[[i],j]\in c([i])} \left[\mathbb{E}(X_{[i]\to j}) + \mathbb{E}(M_{[[i],j]}^{\mathrm{B}})\right] \\
&= \sum_{[[i],j]\in c([i])} [1 + (1 + l([[i], j]))\alpha \\
&\qquad + l([[i], j]) + D([[i], j])\alpha + \mathcal{O}(\alpha^2)] \\
&= \left[\sum_{[[i],j]\in c([i])} 1 + l([[i], j])\right] \\
&\quad + \left[\sum_{[[i],j]\in c([i])} 1 + (D([[i], j]) + l([[i], j]))\right]\alpha + \mathcal{O}(\alpha^2)
\end{aligned}$$

$$= l([i]) + D([i])\alpha + \mathcal{O}(\alpha^2)$$

In the last sum, the first term corresponds to the link from $[i]$ to a child $[[i], j]$, and the second term corresponds to the sum of depths of all links in the subtree rooted at $[[i], j]$ incremented by 1, *i.e.,* the depth as counted from the root $[i]$. □

A proof of theorem 1 for multicast reliability can now be expressed.

Theorem (Traffic footprint for multicast) *The following approximation holds when $\alpha \to 0$:*

$$\mathbb{E}(M^m_{[1]}) = L + \left[L^2 - L(D-1)\right]\alpha + \mathcal{O}(\alpha^2)$$

Proof: Let $d(l)$ be the depth of a link l as seen by the root (a link between the root and one of its children having depth 1). Using equation (6), it is possible to write:

$$\begin{aligned}
\mathbb{E}(F_{[1]}) &= \sum_l (1-\alpha)^{d(l)-1} \\
&= \sum_l \left[1 - (d(l)-1)\alpha + \mathcal{O}(\alpha^2)\right] \\
&= L - L(D-1)\alpha + \mathcal{O}(\alpha^2)
\end{aligned}$$

Combining proposition 2 and lemma 3 yields:

$$\begin{aligned}
\mathbb{E}(M^{\mathrm{m}}_{[1]}) &= \mathbb{E}(F_{[1]})\mathbb{E}(T_{[1]}) \\
&= (L - L(D-1)\alpha + \mathcal{O}(\alpha^2))(1 + L\alpha + \mathcal{O}(\alpha^2)) \\
&= L + [L^2 - L(D-1)]\alpha + \mathcal{O}(\alpha^2)
\end{aligned}$$

which concludes the proof. □

Finally, the following proves theorem 1 for unicast reliability.

Theorem (Traffic footprint for unicast) *The following approximation holds when $\alpha \to 0$:*

$$\mathbb{E}(M^u_{[1]}) = L + \left[C\Delta - L(D-1)\right]\alpha + \mathcal{O}(\alpha^2)$$

Proof: As in the proof for multicast reliability, the first term in equation (11) can be approximated as: $\mathbb{E}(F_{[1]}) = L - L(D-1)\alpha + \mathcal{O}(\alpha^2)$. Hence, the whole expectation can be approximated as:

$$\begin{aligned}
\mathbb{E}(M^{\mathrm{u}}_{[1]}) &= \mathbb{E}(F_{[1]}) + \sum_{[c]\in\mathcal{C}} \frac{(1-(1-\alpha)^{d([c])})^2}{\alpha(1-\alpha)^{d([c])}} \\
&= \mathbb{E}(F_{[1]}) + \sum_{[c]\in\mathcal{C}} \frac{\alpha^2 d([c])^2 + \mathcal{O}(\alpha^3)}{\alpha} \\
&= L - L(D-1)\alpha + \mathcal{O}(\alpha^2) + \sum_{[c]\in\mathcal{C}} d([c])^2\alpha + \mathcal{O}(\alpha^2) \\
&= L + [C\Delta - L(D-1)] + \mathcal{O}(\alpha^2)
\end{aligned}$$

□