Assuring QoS Guarantees for Heterogeneous Services in RINA Networks with ΔQ

7 authors, including:

Some of the authors of this publication are also working on these related projects:

BeefSteak-O-Matic View project

JNT Coloured Books Network Protocols View project

Assuring QoS Guarantees for Heterogeneous Services in RINA Networks with ΔQ

Sergio Leon Gaixas, Jordi
Perelló, and Davide Careglio
Universitat Politècnica de Catalunya
Barcelona, Spain
email: slgaixas@ac.upc.edu

Eduard Grasa, Miquel Tarzan
Fundació Privada i2CAT
Barcelona, Spain

Neil Davies and Peter Thompson
Predictable Network Solutions
Glastonbury, United Kingdom

Abstract— **With the increasing usage of cloud computing and dependence on a diverse set of distributed applications, users are reliant on consistent outcomes from a shared infrastructure. This drives the need for improved QoS guarantees for heterogeneous communication requirements over shared networks. The Recursive Inter-Network Architecture (RINA) is a fundamental programmable network architecture that provides a consistent model for supporting QoS across multiple layers. In this work we evaluate the performance outcomes provided by such polyservice RINA networks in conjunction with per-layer ΔQ-based resource allocation policies. ΔQ provides a resource allocation model able to enforce strict statistical limits on the maximum experienced losses and delays through the smart utilization of traffic policing and shaping strategies, together with an analytical pre-dimensioning of buffer thresholds. Our target scenario is a backbone network that prioritizes communications among geographically distributed datacentres using resources shared with best-effort background traffic. Results obtained with the RINASim simulation software show that a ΔQ-enabled RINA network can yield the desired absolute QoS guarantees to the assured traffic classes without negatively impacting the rest, unlike current MPLS-based VPN solutions.**

Keywords— RINA; QoS; deltaQ; inter-datacentre

I. Introduction

Current IP networks lack the ability to respond to the increasing variety of communication requirements of heterogeneous distributed applications. Even worse, such networks do not provide standard means for applications to even express their communication quality requirements in terms of maximum delay, data loss, etc. It is of course possible to perform some QoS differentiation at the IP layer. For example, solutions like Multi-Protocol Label Switching (MPLS)-based Virtual Private Networks (VPNs) guarantee a minimum bandwidth with internal QoS differentiation, but usually at the expense of degrading the remaining traffic to a best-effort treatment using the available resources.

In this context, the Recursive InterNetwork Architecture (RINA) [1,2] allows applications to request communication services with statistically bounded metrics such as loss or delay. RINA is a recursive, multi-layer architecture that models computer networking as distributed Inter-Process Communications (IPC). All layers in RINA perform the same set of functions, but configured with specific policies that best

deliver the requested outcomes. In RINA, each layer knows the expectations of the applications using it, and the level of service that it can expect from the layers below. Given this recursive, programmable architecture and the knowledge of requirements, RINA becomes an interesting Internet model capable of supporting a myriad of distributed applications with heterogeneous requirements.
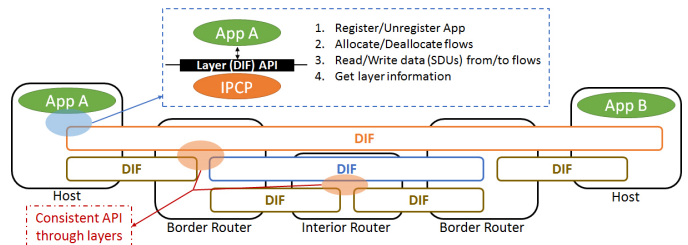

Fig. 1   Structure of the Recursive InterNetwork Architecture (RINA)

In this work we focus on evaluating the service outcomes provided by such polyservice RINA networks exploiting the concept of ΔQ [3,4,5], which leads to a resource allocation framework able to guarantee predictable outcomes to data flows with heterogeneous quality requirements. The rest of the paper is organized as follows. Section II introduces RINA and some of its benefits compared to the current Internet model. Section III introduces the concept of ΔQ and how can it be incorporated in RINA networks. Section IV details our use case scenario and the specific assumptions made in our analysis. Section V provides illustrative numerical results. Finally, section VI concludes the paper.

II. Recursive InterNetwork Architecture

RINA is an architecture for computer networking based on the idea that networking is distributed IPC and only IPC [6]. As illustrated in Fig. 1, RINA presents a single type of layer, a distributed application called a Distributed IPC Facility (DIF), that repeats as many times as needed by the network design. While in the Internet architecture each layer contains a different set of functionalities and offers different Application Programming Interfaces (APIs) to upper layers, RINA defines a single programmable layer that contains all the functions that are needed to provide IPC services to applications or higher level DIFs. RINA specifies the different functions and components contained in the DIF, and a set of variable

behaviours (policies) for each function. This allows the network administrator to properly select policies for each DIF depending on its scope, operating environment, and level of service provided by its lower level DIFs.

Even though all DIFs provide the same type of service, the characteristics of the offered service will vary between DIFs. Each DIF defines a set of supported QoS Cubes, i.e., QoS classes for flows providing statistical bounds on metrics like data rate, latency, losses, etc. Given these QoS Cubes, applications and upper DIFs can request flows with specific requirements, and the DIF then provides a flow with the QoS Cube that matches these requirements, using the policies (scheduling, routing, etc.) best suited for its purpose.

The tasks of creating and managing DIFs are accomplished by the Network Management System (NMS), a distributed application composed of one or more Manager(s), and a Management Agent (MA) residing in each device. The NMS is responsible for creating RINA nodes, triggering their enrolment into DIFs, monitoring them, updating their configuration, etc. As each DIF knows the requirements of all its upper flows, RINA can make more informed and optimal decisions within each layer, including the requests to lower layers, e.g., to allocate new flows with certain QoS levels.

Given the layering structure of the DIFs and the control of the NMS, changes in the connectivity between nodes in one layer can be completely hidden from upper layers. This, together with its complete naming and addressing schemes and configurable policies, makes RINA a smart substitute for the current IP model. Despite RINA's clear differences from the current IP model, it remains compatible with it, allowing a progressive migration at little expense, either by replacing lower layers with RINA while keeping IP networks and services on top, or by using any network protocol (e.g. Ethernet, WDN, IP, UDP, etc.) as a bearer for RINA.

III. CONCEPT OF QUALITY DEGRADATION

In a computer network, quality is something that is never gained, but can only be lost. An ideal network would copy information with zero data loss and zero latency. However, real networks introduce latency and can lose data. This reduction in quality is called attenuation or degradation, and how this is shared between different flows competing for shared resources is the core of any QoS assurance strategy. In particular, three properties are important in terms of quality: bandwidth, latency and losses.

We use 'ΔQ' to designate the quality attenuation between two points along a network path. The performance of IPC depends on different details of the packet delivery depending on the protocol/application; hence, average measures do not contain sufficient information to guarantee the performance of every IPC. Thus, the measure of ΔQ is based on distributions instead of averages or moments. For example, saying that in traversing a network segment "at most 0.01% of packets will experience more than 5ms of latency" provides more useful bounds that "the average latency is 3ms".

With respect to latency and loss, there are two distinct causes of degradation in any network. Firstly, any bearer introduces some degradation, given by metrics like packet service time and distance, which is independent of the current usage of it. Secondly, there is variable degradation due to contention for shared resources such as buffers. This variable degradation involves three parameters: offered load, latency and loss, connected by a relation with two degrees of freedom. Specifically:

- For a fixed load, reducing latency means increasing losses.
- For a fixed latency, as the offered load increases, so does the probability of dropping packets.
- For a fixed loss probability, increasing load increases latency.

Within these constraints, it is possible, for a given load, to adjust the sharing of latency and loss degradation to correlate with the needs of the flows.

As some applications are more sensitive to losses than others, and the same can be said for latency, we can say that some flows are more cherished (require lower losses) or more urgent (require less latency). Hence, their requirements can be mapped into a Cherish/Urgency (C/U) matrix, that is, an NxM matrix with relative latency and losses at each edge (a 3x2 C/U matrix is shown in Table I). This has a straightforward implementation called a Cherish/Urgency multiplexor (C/U mux) [3]. A C/U mux provides differential loss probability using a shared buffer with higher thresholds for packets of more cherished flows, and differential urgency by giving higher precedence service for packets of more urgent flows. Just as total delay is conserved under scheduling [7], ΔQ is conserved in C/U multiplexing.

TABLE I EXAMPLE OF A 3X2 CHERISH/URGENCY MATRIX

C/U	+ Cherished	….	- Cherished
+ Urgent	A1	A2	A3
- Urgent	B1	B2	B3

In addition to the C/U multiplexor, a Policer/Shaper (P/S) can be used to re-shape, limit the flow rate or even change the C/U class of flows, to better adapt to the different QoS requirements and provide protection versus breaches on contracts or the maturing of overbooking hazards. While not always required, this P/S allows a better assurance of QoS and a way to relate the distinct services in different congestion scenarios, and gives considerable flexibility in controlling the ΔQ experienced by assured flows. The C/U mux by itself implements a strict partial order between the ΔQ experienced by the different classes; combining this with bounds on the offered load delivers absolute statistical guarantees.

ΔQ can be composed and decomposed along a path [5], corresponding to multiplexing points at DIF boundaries. ΔQ at an N-level DIF can also be related to the ΔQ provided by its underlying (N-1)-level DIF(s) and its own operation. Thus, the compositional properties of ΔQ and the DIF structure of RINA are complementary. Furthermore, flows sharing a C/U class experience similar ΔQ, allowing for complete flexibility in aggregating and disaggregating flows, enabling a scalable approach to ensuring quality.

IV. SCENARIO DESCRIPTION

Given the importance of QoS-aware networks and cloud computing for future network scenarios like 5G, remote applications for smart devices, sensor networks, etc., providing and assuring differentiated and bounded levels of
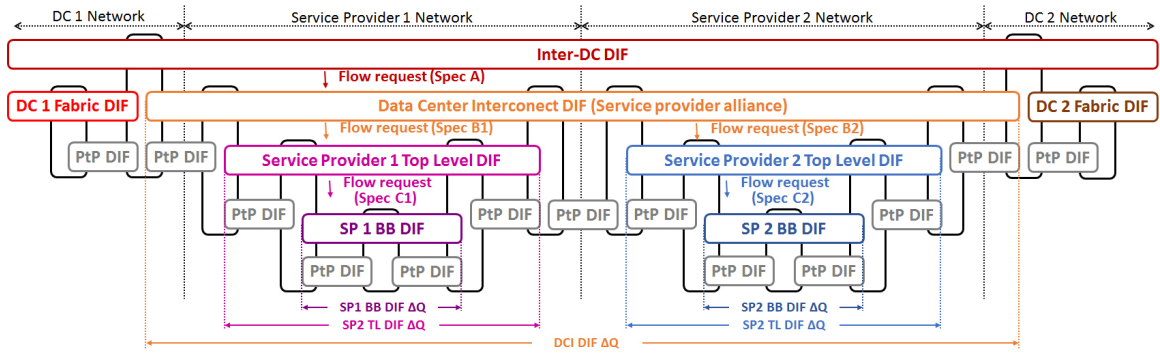
Fig. 2   DIF configuration in the assumed scenario

QoS on backbone networks is a must, so we took these networks as an interesting study case.

In the scenario under consideration, a network provider offers different types of IPC services. Some are targeted to providing connectivity between distributed datacentres, as illustrated by Fig. 2, while others are for general best-effort traffic. The provider network consists of two main layers: the Top Level DIF, which is exposed to customers and allows them to access the provider's IPC services, and the backbone (BB) DIF, which is an internal layer where most of the routing and resource allocation policies are enforced. We will analyse the backbone DIF, in charge of providing adequate differential traffic treatment for the different traffic classes.

To analyse the benefits of a RINA backbone DIF with ΔQ-based policies, we consider a DIF with the same internal structure as the 10-node IP/MPLS layer of the Internet-2 backbone network [8] as shown in Fig. 3, where link latencies are derived from the real physical distances between node locations. Circles represent nodes in the DIF (called IPC Processes or IPCPs) and solid lines flows provided by the multiple N-1 level point-to-point DIFs shown in Fig. 2 (called N-1 flows). Moreover, we assume that all N-1 flows have a capacity of 10 Gbps and impose a Maximum Transmission Unit (MTU) of 5KB. Note, however, that the achieved results would also be representative of scenarios with higher N-1 flow capacities (40 or 100 Gbps), provided that the offered loads are scaled accordingly.
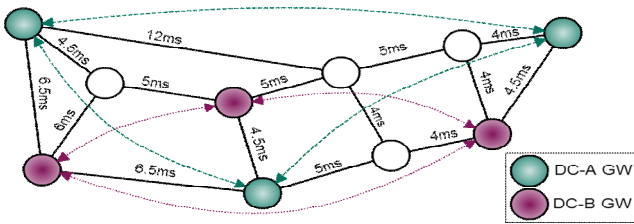

Fig. 3   ISP Backbone DIF describing a network of 10 nodes and 15 bidirectional links used for the experiments, where the locations of the datacentres are highlighted. Dashed and dotted arrows illustrate inter-datacentre communications.

In this scenario, we assume two distributed datacentres (DCs), each one situated in three different geographical locations. These exchange two different classes of inter-DC traffic directly mapped down to the ISP Backbone DIF, whose characteristics are detailed below:

- **Gold/Urgent traffic (GU)**: Assured bandwidth, minimum latency and jitter. Essentially lossless.
- **Silver/Non-Urgent traffic (SN)**: Assured bandwidth, low latency and jitter. Nearly lossless (weaker requirement compared to GU traffic).

A 1:4 GU:SN traffic ratio has been assumed when generating the inter-DC flows. Moreover, each pair of data centre locations exchanges up to 2 Gbps and 1 Gbps in DC-A and DC-B, respectively. This inter-DC traffic mix has to share the available lower-level N-1 flows (hereafter referred as "links" for simplicity) capacities with background traffic flows of two different classes:

- **Sensitive Best-effort (sBE)**: Minimum latency and jitter. Allows losses but preferable to limit consecutive packet losses on flows to less than 3 (it accepts losses for voice and video streaming, sensor updates, etc.).
- **Best-effort (BE)**: Traffic class with the lowest requirements in terms of losses, delay and jitter.

Each pair of RINA IPC processes (referred to as "nodes" from now on) exchange sBE and BE flows (representing a full-mesh background Internet traffic) with a 3:7 sBE:BE traffic ratio, so that all DIF links are filled with a similar load level. Our motivation behind this is to assume a worst-case scenario (in terms of congestion) in which all links are heavily loaded. Furthermore, we use a simple Equal Cost Multi-Path (ECMP) forwarding with memory per flow [9,10] in order to perform load balancing without introducing jitter or the need for reordering caused by disjoint path usage by the same flow.

A. Configuring RINA ΔQ-POLICIES

Traffic classes in the scenario give some approximation on QoS requirements derived from user expectations (assured bandwidth, low latency/jitter, etc.). However, more specific information is needed for the distinct types of flows in order to complete the DIF configuration.

We can observe that, while we have requirements for low latency, the stationary latency caused by the distance between nodes may be higher than any delay introduced by nodes due to medium or moderately high congestion. This has two important implications when configuring the DIF. Firstly, it is clear that following the shortest paths (in terms of hops) may not provide optimal latency for delay-sensitive flows. Instead, we can use the total stationary latency as a metric for routing the more urgent flows, resulting in an overall improvement of

their experienced performance. Secondly, although the stationary factors may dominate the average latency, the congestion in network nodes is the source of jitter. In order to control this jitter, we are going to put strict requirements on the degradation of latency in each hop, in terms of packet service times (PST).

Apart from considering latency and jitter degradation, our focus is the sharing of loss degradation amongst competing flows. Primarily, given the importance of inter-DC traffic, we are interested in making this almost lossless, even in overloaded links. After assuring that, we next guarantee that sBE and BE flows do not suffer excessive losses upon high congestion. Particularly, we aim to ensure that sBE flows do not suffer consecutive packet losses, as this can negatively affect real-time and streaming applications' performance.

To reify these goals, we have set the following end-to-end requirements (w/ high probability) in our scheduling policies:

1. GU traffic must experience zero losses up to at least 150% aggregated load (i.e., relative to the total link capacity).
2. SN traffic should be supported without losses up to at least 120% aggregated load.
3. Losses of sBE and BE traffic should be below 0.05% up to at least 95% aggregated load.
4. GU and sBE flows should not exceed 50 PST of variable latency up to at least 120% aggregated load.
5. SN and BE flows should not exceed 1000 PST of variable latency up to at least 120% aggregated load.
6. sBE flows should not lose more than 3 consecutive packets up to at least 110% aggregated load.

B. QoS to Cherish/Urgency classes

We can map the requirements of the GU, SN, sBE and BE traffic classes, into the 4x2 Cherish/Urgency matrix shown below in Table II. In order to simplify the configuration, we consider the same worst-case scenario at each node, and consider ΔQ in a way that requirements are met in the longest paths. To achieve this, we set some requirements per hop for the different levels of cherish and urgency depending on the current aggregated load on the link, subsequently illustrated in Tables III and IV.

TABLE II QoS TO CHERISH/URGENCY CLASSES

C/U	+ Cherished	….		- Cherished
+ Urgent	GU		sBE	
- Urgent		SN		BE

TABLE III MAXIMUM AVG. LOSS (%) PER LOAD PER HOP

Load/QoS	0.90	0.95	1.00	1.20	1.50
GU	~0%				0.001
SN		~0%		0.001	0.01
sBE	0.001	0.04	0.2	-	
BE	0.002	0.05	0.3	-	

TABLE IV MAXIMUM PST REQ. PER LOAD PER HOP

Load/QoS	0.90	0.95	1.00	1.20	1.50
GU, sBE	<10				<15
SN, BE	<75	<125	<200	<225	<250

Being in a RINA scenario where the load of the flows can be controlled at source and destination IPCPs in a trusted way, the requirement for flow policing is minimized, so we omitted it from the analysis. Instead, we used a modified version of the C/U mux based on: i) a first heuristic threshold; ii) a second absolute threshold of total buffer occupancy. When the total occupancy reaches the heuristic threshold, i.e., congestion starts to occur, a proportion of the incoming packets will be independently randomly dropped. With this simple heuristic threshold, we can largely reduce the probability of consecutive losses when reaching the absolute threshold while, at the same time, allowing the different cherish classes to better share losses.

C. ΔQ analysis

As stated before, the triad Bandwidth-Latency-Loss has two degrees of freedom. We assume a worst-case scenario where the input rate is fixed at the maximum load, and have to decide how to share ΔQ between delay and losses. In the common case, having largely elastic traffic would cause the average load to not exceed 100% by much, however transient traffic loads can be higher, which is why we want to ensure QoS guarantees even under considerable overloads. With the fixed load assumption, we can know the ratios between the offered load of the different QoS cubes. We use these ratios under the considered link load levels (90, 95, 100, 120 and 150%) to configure the resources in the different nodes in a way that accomplishes our requirements when possible.

This configuration should be computed per node. However, we simplify it here to a single configuration that ensures the requirements in worst-case scenarios, thus obtaining a compliant configuration for any node (although most probably not the optimal one). To this end, we require our configurations to ensure per hop requirements for all QoS classes in scenarios where the ratio of load between GU/SN and sBE/BE is 1:9, 2:8 and 3:7 (1:4 between GU and SN and 3:7 between sBE and BE). Given that the offered load is fixed, with the C/U-mux configuration above, the only remaining point to configure is the different buffer thresholds. In order to configure such thresholds, we consider the analytical approach described in [5]. To include the use of the first threshold in the analysis, we take the most pessimistic view of introducing PDU drop, namely: for each queue we consider that, for queue lengths beyond the first threshold, all PDUs were discarded, while for the other queues PDUs were not discarded until their second threshold. To reduce the possible set of configurations, we fixed the probabilistic threshold for each QoS class at 10 buffers below the absolute one, except for GU for which we only consider an absolute one. In addition, we consider the same absolute thresholds for GU and SN (A) and for sBE and BE (B). As for the drop probability applied when the heuristic threshold is exceeded, we set it to 0.1 for SN and sBE and 0.2 for BE.

TABLE VI PARAMETRISED BUFFER PER QoS

QoS	Heuristic Threshold	Drop Probability	Absolute Threshold
GU	-	-	120
SN	110	0.1	120
sBE	90	0.1	100
BE	90	0.2	100

After analysing the possible configuration ranges, we settled on the one in Table VI, which fulfils the imposed requirements while allowing for a small margin of error in case of having non-Poisson arrival patterns.

V. NUMERICAL RESULTS

A. ΔQ configuration validation

In order to validate the configuration for our scenario, we perform simulations with the RINASim simulation software developed in the FP7 PRISTINE Project [11,12]. As we have configured our DIF in a way that ΔQ requirements are ensured per hop in a worst-case scenario, our first tests aim to check whether the expected behaviour computed via analysis of loads and requirements is delivered in a simple single-hop scenario. This scenario is simply composed of a sequence of three nodes inter-connected by two 10Gbps links. For each offered load value and ratio between inter-datacenter flows and best-effort ones (1:9, 2:8 and 3:7), a 10s simulation is performed. In each run, 1000 flows are established, served following an exponential distribution and with average data rate distributed between 7 and 13 Mbps (At 100% load) and spread between the distinct QoS given the stated rates between QoS. To add more randomness, packet sizes were uniformly distributed between 2KB and 4KB.
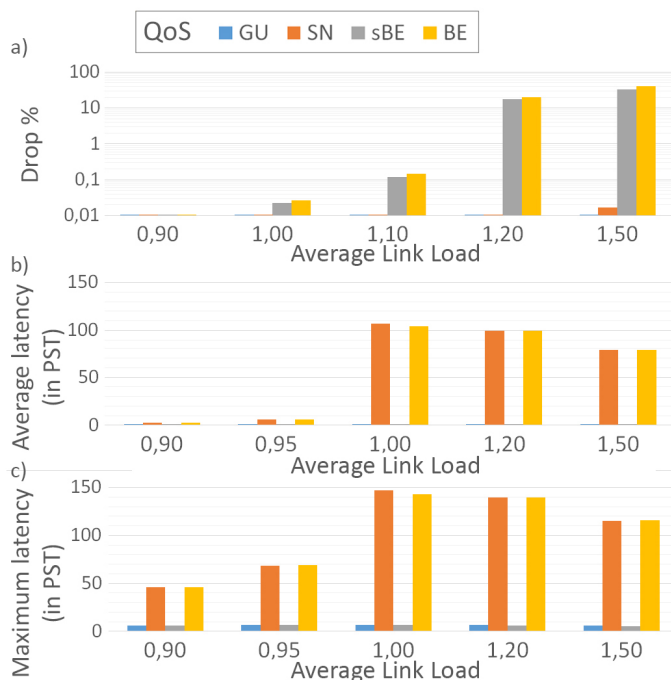

Fig. 6. For single hop flows at distinct loads: a) Average drop (%); b) Average delay; c) Max delay.

The results from these first experiments confirm those from the analysis. As shown in Fig. 6-a, inter-DC flows (GU and SN) remain practically lossless in all the experiments, successfully fulfilling their loss requirements. In addition, the losses experienced by sBE and BE flows are within the allowed limits and their losses match pretty much the over-load in the network. In terms of latency, as shown in Fig. 6-c, urgent flows encountered less than 10 preceding packets, having to wait on average for 0 or 1 packets to be served, as

shown in Fig. 6-b. Less urgent flows also remained within acceptable delays, correctly shared between both classes. As a note, is interesting to see how the increment in losses for sBE and BE flows resulted in smaller latencies for non-urgent flows given the changes on accepted loads for each QoS.

B. Simulating the full backbone DIF

Having validated the configuration of isolated nodes, we proceed to perform backbone DIF-wide experiments. For these experiments, we offer a load to the DIF so that all links are equally loaded, as mentioned in the previous section. We consider the previously stated distribution of bandwidth for a full load, allocating the same type of flows (7 to 13Mbps at 100% load, 2-4KB PDUs, etc.) between the distinct pairs of nodes, and then scaling the flow data rate to the desired DIF load. For inter-DC traffic we set up 40 GU and 160 SN flows (Avg. 2 Gbps at 100%) between DC-A nodes and 20 GU and 80 SN flows between DC-B nodes (Avg. 1 Gbps at 100%). In addition, as we are interested in the degradation of sBE and BE flows, we set 30 sBE and 70 BE flows between those same pairs of nodes, enough to get comparable data with respect to the other two classes. Finally, we set multiple point-to-point sBE and BE flows, in a 3:7 ratio, between all pairs of nodes in order to reach the targeted 1000 flows per link.

Regarding the considered scenario, there are two points to note. Firstly, we are considering a worst-case scenario where no congestion control is in use, whereas RINA allows (and promotes) a multi-layer congestion control, with fast detection and reaction, which would reduce the rates of sBE and BE flows once network congestion starts to happen. Nonetheless, as we disable the congestion control in these tests, we have a scenario where only scheduling actions are taken to respond to congestion problems. Secondly, statistics are only computed for flows with multiple hops whose ΔQ increases along their paths, whereas 'dumb' flows used to fill links are point-to-point. This means that their ΔQ does not affect the behaviour at other nodes, that is, the losses at one congested node do not reduce the incoming rate of PDUs at downstream nodes, which would happen if the network had been filled with multi-hop flows.

In this work, we do not limit ourselves to only establishing that end-to-end requirements are met using the RINA + ΔQ-based policies. Instead, we also compare these results with other solutions currently in use, in particular a baseline entirely Best-Effort scenario (referred as BBE) and an MPLS-based VPN [13,14]. In both cases, we used the same routing for fair comparison purposes, where urgent flows use latency as the metric to optimize and the rest use number of hops. For the Best-Effort scenario, we use a simple FIFO queue at each node with the same 120 packets absolute threshold as the one used for the GU class. For the MPLS-based case, we configure a Weighted Fair Queuing (WFQ)-based scheduling where 260 buffers are distributed as follows: 80 for GU and SN and 50 for each best-effort class. In this last case, in order to ensure the loss levels for GU and SN flows, 40% of the available bandwidth is reserved for GU flows and 30% for SN flows, while the remaining bandwidth is shared between sBE and BE flows following a 2:1 ratio.

From these experiments, we find some interesting results in favour of RINA + ΔQ-based policies. In RINA, we find that

GU and SN flows are lossless, while losses in sBE and BE flows are well distributed and satisfy the requirements previously stated, as can be seen in Fig. 7-a. We can also compare these results against the WFQ-based scheduling policy configured to satisfy the requirements of GU and SN services, while allowing some differentiation between sensitive and non-sensitive best-effort flows. As can be observed in the same figure, BE flows are the only ones experiencing dramatic losses. These losses also happen earlier, given the division of buffering space and low priority. Regarding the BBE baseline case we find that, while being the last one experiencing losses, as all packets are accepted until reaching the 120 packets threshold, all classes share losses uniformly, failing to ensure GU and SN loss requirements.

In terms of latency, as variable latency is not of great importance in this scenario, we focused instead on the maximum jitter in PST experienced by each traffic class. In the RINA + ΔQ scenario, we find that the requirements per hop are provided, thus meeting the constraint of ensuring minimum jitter for urgent flows, while also limiting it for the less urgent ones, as can be seen in Fig. 7-b. In contrast, both WFQ and BBE encounter problems. As for BBE, equally sharing the available resources among all flows increases the jitter for urgent flows to unacceptable levels. On the other hand, WFQ provides good service to both GU and SN traffic (even better than required). However, this is achieved at expenses of increasing sBE and BE losses and jitter.
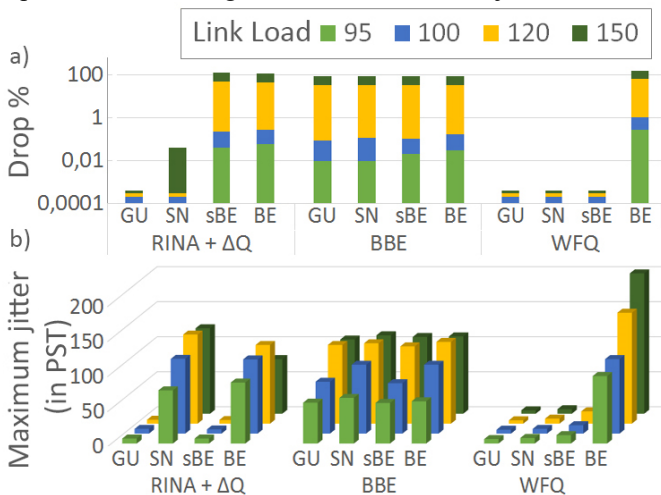

Fig. 7. Average drop (a) and maximum jitter in PST (b) for GU, SN, sBE and BE flows depending on the scheduling policy used in the network

Conversely, in addition to providing better assurance on delay and losses, the RINA + ΔQ-based policies also meet the requirements of avoiding multiple consecutive losses under light congestion. Particularly, we found that for a link load of 110%, even thought there are necessarily high losses in average, the soft requirement of having at most 3 consecutive losses in sBE flows was pretty much fulfilled. Under higher loads, it becomes nearly impossible to avoid consecutive losses given the multiple points of congestion. However, those are still limited to, for example, less than 1% situations of more than 3 consecutive packets for an offered load of 120%.

VI. CONCLUSIONS

Future networks must become demand-attentive in order to support the requirements of multiple distributed applications over the same infrastructure. Assuring absolute QoS guarantees is a must, so that each application using the network gets the quality it requires, no more and no less. In this paper, we have presented RINA, a multi-layer, recursive, programmable network model based on the distributed IPC paradigm that provides scalability and QoS assurance guarantees. In conjunction with the ΔQ resource allocation model for quality degradation, we have shown that it is possible not only to provide quality of service differentiation with SLA assurance, but doing it while also ensuring a bounded impact on less stringent services.

While the testing scenario was simple, with rather non-restrictive requirements, real scenarios are more complex and therefore require more complex configurations and more restrictive assurances. Future work in this area will be directed to providing dynamic/reactive re-configuration of nodes given changes on the network, working on QoS-aware routing policies, and providing a wide range of QoS classes to ensure any type of ΔQ requirement.

REFERENCES

[1] E. Trouva, E. Grasa, J. Day, I. Matta, L. T. Chitkushev, P. Phelan, M. P. d. Leon and S. Bunch, "IS THE INTERNET AN UNFINISHED DEMO? MEET RINA!", October 2010.

[2] J. Day, I. Matta, and K. Mattar. "Networking is IPC: A Guiding Principleto a Better Internet". In CoNEXT'08: Proceedings of the 2008 ACMCoNEXT Conference, pages 1-6, New York, NY, USA, 2008. ACM.

[3] Predictable network Solutions Limited, "A Study of Traffic Management Detection Methods & Tools". OfCom, United Kingdom, August 2015.

[4] N. Davies, "Delivering predictable quality in saturated networks," Technical Report, September 2003, available online at http://www.pnsol.com/public/TP-PNS-2003-09.pdf

[5] N. Davies, J. Holyer and P. Thompson, "An operational model to control loss and delay of traffic in a network switch," in third IFIP Workshop on Traffic Management and Design of ATM Networks, 1999.

[6] J. Day, Patterns in network architecture: a return to fundamentals, Prentice Hall, January 2008.

[7] L. Kleinrock "Queuing System, Vol 1: Theory" Wiley, 1975, p.117

[8] Internet2 website, available at http://www.internet2.edu

[9] Y. Ganjali and A. Keshavarzian "Load balancing in ad hoc networks: Single-path routing vs. multi-path routing", Proceedings of IEEE INFOCOM

[10] Srikanth Kandula, Dina Katabi, Shantanu Sinha, and Arthur Berger. 2007. Dynamic load balancing without packet reordering. SIGCOMM Comput. Commun. Rev. 37, 2, pp 51-62, March 2007.

[11] PRISTINE FP7-619305, Programmability in RINA for European supremacy of virtualized networks, http://ict-pristine.eu

[12] RINASim, available at https://github.com/kvetak/RINA

[13] Haeryong Lee; Jeongyeon Hwang: Byungryong Kang; Kyo "End-To-End QoS Architecture for VPNs: MPLS VPN Deployment in a Backbone Network". Parallel Processing, 2000. Proceedings 2000 lnternational Workshops, pp 479-483, 2000.

[14] V. Fineberg, "A practical architecture for implementing end-to-end QoS in an IP network", IEEE Comm. Mag., vol. 40, no. 1, pp.122 -130, January 2002