# Congestion Avoidance Through Deterministic, Adaptive and Scalable Resource Reservation

Pier Luca Montessoro, Riccardo Bernardini, Franco Blanchini, Daniele Casagrande, Mirko Loghi, Stefan Wieser
*Members, IEEE*
Dep. of Electrical, Managerial and Mechanical Engineering University of Udine - Udine, Italy
{ montessoro, bernardini, blanchini, casagrande, loghi }@uniud.it, swieser@edu.aau.at

***Abstract*—A new paradigm based on per-flow state information stored in network components is proposed. It is based on a set of new distributed algorithms recently published that overcome performance and scalability issues.**

## I. INTRODUCTION

The concept of increasing cooperation between intermediate systems and end nodes is gaining popularity, e.g., in the IETF Congestion Exposure (conex) working group projects [1]. As discussed in [2] and [3], "congestion control cannot be realized as a pure end-to-end function only. Congestion is an inherent network phenomenon and can only be resolved efficiently by some cooperation of end-systems and the network."

Recently, the congestion control problem has been addressed by some research works in the direction of *congestion avoidance* through a deterministic resource booking schema where the routers handle the requests coming from the end nodes by storing per-flow state information to keep trace of the accepted reservations. This approach has traditionally been considered non-scalable, but, provided the increased size of the memory available for control information, the algorithms presented in [4] (REBOOK) and in [5] (Distributed Linked Data Structure, DLDS) show how it is possible to access each flow state at constant cost. This way, an end node adjusts its transmission rate according to the confirmed reservation, and waits for resources to be released if the request has been rejected. [6] presents our Integrated framework for Multimedia Flow Management (IMFM), which provides resource reservation and virtual circuit–like performance on a conventional packet switching network, without affecting the underlying routing protocols. IMFM is the starting point of this proposal. Although IMFM has been designed primarily for video streaming, large data file transfer can benefit of this technique as well. It is likely that guaranteed delivery time or minimum transfer rate will be required even for non-multimedia data in the future, to provide effective cloud services.

To the authors' knowledge, these algorithms represent one of the few, if not the only, proposals that can actually change the scenario for multimedia delivery and large data transfer on the Internet. They provide a new paradigm, compatible with existing protocols, and a feasible way for actual deployment. The increasing interest in cloud infrastructures and content-centric networks represent an ideal opportunity to develop and deploy the techniques here proposed.

## II. RELATED WORK

Congestion control is a key topic for the future Internet. Many protocols (approved or just proposed), algorithms and research works address congestion control; a discussion related to REBOOK can be found in [4] (Section 2) whereas [7] provide a comprehensive description of the known techniques.

RFC 6077 [3] presents a wide analysis of the state of the art of the algorithms and protocols related to congestion control, together with the challenges that future research must face.

We propose IMFM as the enabling technology for a new paradigm where per-flow information in network components is no longer a taboo. As described below, IMFM can provide solutions and support to new developments in several of the challenges discussed in RFC 6077:

*Challenge 1: Network Support*

Par. 3.1.2: "… different amounts of state can be kept in routers […]. The additional router processing is a challenge for Internet scalability and could also increase end-to-end latencies." The DLDS algorithm addresses exactly this problem, providing O(1) complexity for accessing flow state kept in routers.

*Challenge 4: Flow Startup*

Par. 3.4: "when a connection to a new destination is established, the end-systems have hardly any information about the characteristics of the path in between and the available bandwidth.". REBOOK provides deterministic resource reservation, so that not only the end systems know the available bandwidth available along the path, but it is guaranteed that the confirmed amount is reserved too.

*Challenge 7: Misbehaving Senders and Receivers*

Par. 3.7: "… self-restraint could disappear from the Internet. So, it may no longer be sufficient to rely on developers/users voluntarily submitting themselves to congestion control. As a consequence, mechanisms to enforce fairness […] need to have more emphasis within the research agenda." With IMFM, each data packet carries a pointer to an entry in a table that contains information referring the flow it belongs to. This makes the access cost to this information constant, independent from the number of active flows. As a result, it becomes possible to keep track of flow behaviour. Packets can be dropped and/or the sender's reservation can be reduced if it exceeds the granted

bandwidth.

## III. THE PROPOSED NEW PARADIGM

IMFM is based on a distributed linked data structure (DLDS) in the sense that it makes use of pointers (addresses of memory locations or indexes) to table entries containing information - stored in the routers - that are very likely to be accessed in the future. Unlike conventional linked data structures, pointers are not stored within the router they address, but in the end nodes, in the packets, and in adjacent routers.

When a packet is received by a router, the pointer to the table entry to be used for its processing is found in the packet itself and lookup procedures are avoided. The location of the pointer within the packet is identified with an offset-field in the packet. IMFM-aware routers read the pointer, retrieve the information from memory, and update the offset in the packet. This adaptation allows IMFM to function in a heterogeneous environment, in which routers use varying amounts of bits to address memory.

A specific integrity check is adopted to keep the pointers consistent in a dynamic environment, where route changes may send packets to routers different than the ones the pointers refer to. It makes IMFM completely autonomous in identifying faults, errors and route changes and does not require any change or interaction with the underlying routing protocols. In contrast to MPLS, ATM and similar techniques, IMFM does not require any agreement across the entire network and between the Autonomous Systems. Moreover, gradual deployment is possible, which is critical for widespread adoption.

The overhead introduced by the integrity check is a critical issue. Thanks to careful design of the distributed linked data structure, experimental data (reported in Section IV) demonstrate that this overhead is overcome by the speedup provided by the lookup-free access to table entries.

A complete description of the algorithms would not fit into the limited length of this paper. In the following, the fundamental elements will be described, referring to [4] (REBOOK), [5] (DLDS) and [6] (IMFM) for the details. We also provide an Internet Draft [10] to aid with REBOOK implementation.

IMFM offers a synergic approach to resource reservation and fast packet forwarding. Resource reservation, provided by the REBOOK algorithm, is soft state and makes use of keepalive messages. A low priority process for table cleanup is used, which removes obsolete entries produced by faults, errors or route changes. Fig. 1 shows an example of resource reservation in a partial deployed scenario, with two IMFM aware routers. The reservation request is 2 Mb/s, but router R4 has only 1 Mb/s available, and this is reported to the sender. Subsequent keepalive messages will confirm the reservation and let R1 release the unused 1 Mb/s. Note that only the non-contiguous routers R1 and R3 are IMFM-aware; the non-IMFM-aware router R2 is seen as a physical link. It is possible that the route chosen by the underlying protocols changes during the lifetime of a flow. In this case, at least one consistency check fails and the list of pointers is invalidated. The absence of keeaplive acknowledges will signal the sender that the reservation has been dropped and a new reservation will be required along the new path. Along the old path the table cleanup procedure will remove the obsolete entries and release the resources assigned to that flow. We present a detailed performance evaluation in [6].

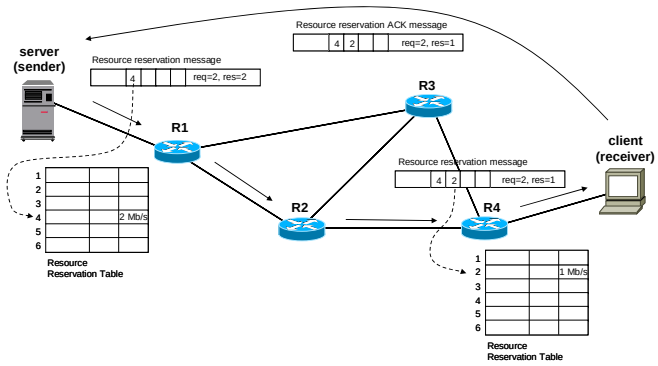


**Figure 1.** Resource reservation and pointers collection

The IMFM-unaware router R2 can be a bottleneck if its capacity is lower than the reservation for this flow. Even in this case, R1 and R4 benefit from IMFM in that the receiver can monitor the data flow bit rate and generate a partial reservation release request if the current reservation is higher than the actual availability along the path. Reservations in excess will be released when this request is received by the server and the first keepalive message, containing the new reservation value, is sent.

Thanks to this ability of reducing the reservation and, consequently, the sender message rate, it is possible to adjust the amount of resources reserved to a flow between its minimum request, below which the multimedia stream cannot be decoded, and its "optimal" request amount, depending on the availability in each router along the path. When a router close to congestion is traversed by many flows, the decision on which flows must be reduced and the reduction amount is critical for stability, fairness and optimality. For this reason, IMFM is not bounded to a specific control function. On the contrary, it provides a framework do develop, implement and test distributed control functions based on per-flow state information stored in routers. In [8] a possible control function is presented.

The second feature of IMFM is the support to lookup-free procedures for fast packets forwarding. Using the same pointers of the resource reservation (or just a similar schema), it is possible to insert in each data packet the pointer to the forwarding table entry to be used in the router being traversed. To keep to the minimum the packet size overhead, a "pointer swapping" technique can be adopted, although for short routes and large packets the overhead of the complete list of pointers could become negligible. Even for this feature, a highly efficient consistency check is used. As for resource reservation, route changes are autonomously handled: the consistency check fails, pointers are invalidated and data packet forwarding falls back to the forwarding table lookup mode. Packets are never dropped nor misrouted by IMFM. Figure 2 shows the behavior of IMFM in case of routing instability (a) in a network traversed by UDP flows controlled by IMFM. After the consistency checks fail, the pointers are invalidated and then restored (b). R2 and R3 saturate, being traversed by all the flows; however, congestion is prevented because the senders' transmission rate is reduced (c). The distributed control function we propose in [8] provides op-

timality and fairness.

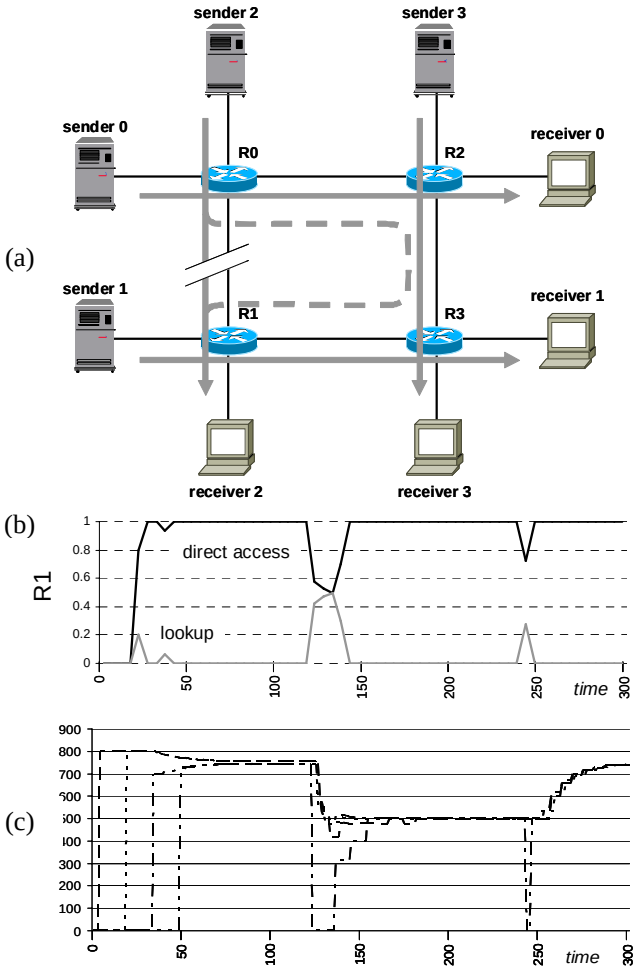


**Figure 2.** Routing instability and its effect on IMFM

IMFM has been fully implemented as a C/C++ library. Currently it is being ported into the Linux kernel of a software router. To mark the packets (data and keepalive) containing IMFM fields the IP Router Alert Option [9] is being used, although in a non-standard format. Specifically, the two octect values are filled with a non-zero value not (yet) compliant with the IANA registry in order to immediately identify IMF packets.

## IV. PERFORMANCE

In [4][5] and [6] several performance measurements are reported. End nodes and routers *emulators* running in independent threads, processes or computers have been implemented to overcome the limits of the simulations. Moreover, the presented results are conservative because all the modules have been run in user space. Tables have been populated with up to 10,000,000 random generated flows.

The measurements show that:

- the overhead for reservation setup, removal, and keepalive messages is negligible
- the cost of the consistency check is negligible if compared with the speedup of the lookup-free accesses, being less than 11 ns the total CPU time - on a 1.6 GHz Intel(R) Core 2 computer – to perform the consistency check and access the entry in the table
- the overhead in terms of additional packets (keepalive messages) and packets size is very limited.

## V. DEPLOYMENT AND SECURITY

The algorithms on which IMFM is based have been developed with issues arising from their deployment in real world networks in mind. Many protocols have been developed, standardized and almost never been used. We believe that our proposal has good chances to be accepted for several reasons.

- IMFM is an open framework, so that it can support already known or future control functions that address TCP friendliness, optimal handling of elastic flows, etc. In any case, the control functions can use at no cost the per flow state information storage managed by IMFM.
- IMFM is an add-on to existing routers and switches, and it can be gradually deployed.
- Pointers generated by a router are stored in the packets and in the end systems, but are used only by *that* router. This means that fast symmetric cryptography can be used to secure IMFM packets without the need of key exchange.

## VI. CONCLUSION

We propose a novel approach to scalable per-flow resource reservation, enabled by efficient algorithms and data structures part of the IMFM framework. By combining the guaranteed constant access cost of information within routers provided by DLDS, with distributed resource reservation provided by REBOOK, we offer solutions to several practical challenges. In addition, our proposal supports gradual deployment, which is critical for a widespread adoption in practice.

## REFERENCES


[1] IETF Congestion Exposure (conex) working group web site, http://datatracker.ietf.org/wg/conex/charter/

[2] T. Moors, “A critical review of “End-to-end arguments in system design”,” Proceedings of IEEE International Conference on Communications, 2002, vol.2, no., pp. 1214- 1219 vol.2

[3] D. Papadimitriou, M. Welzl, M. Scharf, B. Briscoe, “Open Research Issues in Internet Congestion Control,” RFC 6077, 2011

[4] Pier Luca Montessoro, Daniele De Caneva, “REBOOK: a deterministic, robust and scalable resource booking algorithm,” Journal of Network and Systems Management (Springer), Volume 18, Issue 4 (2010), pp. 418-446 ISSN: 1064-7570 (print) 1573-7705 (online), DOI: 10.1007/s10922-010-9167-8

[5] Pier Luca Montessoro, “Distributed Linked Data Structures for Efficient Access to Information within Routers,” Proceedings of IEEE 2010 International Conference on Ultra Modern Telecommunications, 18-20 October 2010, Moscow (Russia), pp. 335-342, ISSN: 2157-0221, print ISBN 978-1-4244-7285-7, DOI: 10.1109/ICUMT.2010.5676616

[6] Pier Luca Montessoro, “Efficient Management and Packets Forwarding for Multimedia Flows,” Journal of Network and Systems Management (Springer), 2012, DOI: 10.1007/s10922-012-9232-6

[7] M. Welzl, “Network Congestion Control - Managing Internet Traffic,” John Wiley & Sons Ltd, Chichester, West Sussex, England 2005

[8] F. Blanchini, D. Casagrande, P.L. Montessoro, “A novel algorithm for dynamic admission control of elastic flows,” Proc. of 50th FITCE congress, Palermo, Italy, August 31th – September 3rd, 2011, pp.110-115

[9] D. Katz, “IP Router Alert Option, ” RFC 2113, 1997

[10] P. Montessoro, R. Bernardini, “REBOOK: a Network Resource Booking Algorithm”, Internet Draft.