

Dynamic Traffic Engineering in the Future (?) Internet

Nils Kammenhuber

Chair for Network Architectures and Services, Prof. Georg Carle
Technische Universität München, Munich, Germany

Abstract—Traffic in data networks is known to fluctuate heavily, which can result in sudden service degradation. Handling these traffic demands that are bursty and hard to predict is a major challenge. We present REPLEX, a distributed algorithm for fast dynamic traffic engineering that is based on game theoretic principles. Previous experiments have shown drastic performance improvements when using REPLEX in traditional networking contexts. In this work, we argue that REPLEX why perfectly suited for application in future networks.

I. INTRODUCTION

Traffic in the Internet is known to change over time and exhibit volatile behaviour. One major challenge in communication networks is the problem of handling these traffic demands that are bursty and hard to predict. Current traffic engineering techniques operate on time scale of several hours, which is too slow to react to quick phenomena such as flash crowds or BGP reroutes. On the other hand, TCP congestion control reacts quickly, but can only help to reduce overload situations along a single path, rather than to find alternative uncongested paths. The obvious solution, load sensitive routing, is frowned upon, since routing decisions at short time scales can lead to oscillations. This has prevented load sensitive routing from being deployed since the early experiences in Arpanet, the predecessor of today’s Internet.

However, theoretical results have shown that a re-routing policy based on game theory provably avoids such oscillation and in addition can be shown to converge quickly [6]. Based on these results, we developed the REPLEX algorithm, a distributed, dynamic algorithm for dynamic traffic engineering with low signalling overhead [5], [7]. In large-scale simulations involving realistic topologies and fractal TCP traffic, REPLEX showed quick convergence without oscillations while being TCP-friendly and incurring performance improvements that are equal to or even better than traditional (i.e., static) traffic engineering methods (chapter 7 in [7]).

In spite of our evaluations so far having only considered setups that are typical of today’s networks, usage of the REPLEX algorithm is not limited to the current Internet, actually not even to data networks (the underlying Wardrop model comes from road traffic research). Due to its flexibility and genericity, REPLEX is actually ideally suited for the future Internet for a number of reasons.

Before we elaborate on this, we first give a brief overview on REPLEX’ functionality. This makes it easier to understand its requirements on the network into which it is to be embedded.

II. HOW REPLEX WORKS

In the underlying game-theoretical Wardrop model, an infinite number of agents at each end host can determine the route of an infinitesimally small quantum of traffic. Each agent tries to selfishly maximize the performance of its own traffic with regard to some global performance metric. Agents continuously watch the performance that other agents achieve and mimic successful routing decisions with a certain probability.

From this rather theoretical model that does not reflect the reality of the Internet very well, we derived the REPLEX algorithm. The basic idea is that an underlying routing protocol (e.g., OSPF or

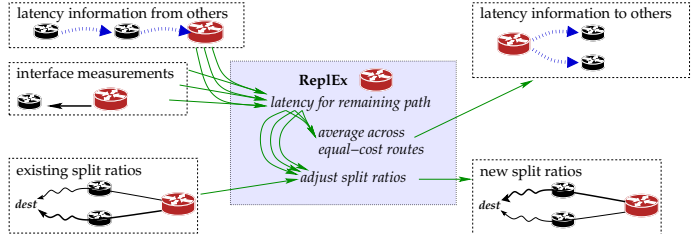


Fig. 1: Data and control flow within a single REPLEX instance.

IS-IS) provides a routing infrastructure, with preferably as many multipath routes as possible. The agents from the theoretical model residing at the network edges are replaced by chains of routers within the network, each of which runs a REPLEX instance that aggregates the behaviour of multiple agents. Each REPLEX-enabled router permanently monitors the network conditions, and adjusts the split ratios for the multipath routes accordingly. In multipath routing across n different paths, each path normally is assigned $\frac{1}{n}$ of the traffic share (usually by treating a hash value over the packet headers as a “random” decision variable). Instead of keeping these shares constant, REPLEX continuously adjusts the ratios for the sub-routes of a multipath route, so as to maximize traffic performance w.r.t. some given metric. This adjustment of the split ratios between a set of given route alternatives does not immediately reflect their measured performance differences; rather, the ratios are slowly adjusted over time. This approach provably avoids overreaction and thus oscillation in the underlying game theoretical Wardrop model [6], which simulations confirmed for REPLEX [5], [7].

The information that is required for a REPLEX node to set its split ratios comes from two sources: First, from local measurements (e.g., queue lengths or packet drop counters); second, from information from its immediate neighbours, which continuously send reports on network conditions along the further path downstream. Apart from being used for adjusting the ratios, information from both sources is aggregated and sent to the neighbours upstream. In conclusion, the signalling of information on network traffic conditions thus is performed through an efficient, low-overhead distance vector like protocol. The entire process is graphically summarized in Fig. 1.

III. WHAT REPLEX NEEDS

REPLEX is very generic and flexible, as it has only very few requirements on the networks into which it is to be embedded.

First, REPLEX is not restricted to specific topologies or specific routing infrastructures. Rather, the only requirement that is inherent to REPLEX is that some underlying routing infrastructure (be it one or several routing protocols, be it manually configured routes, or any combination) provides multipath alternatives to as many destinations as possible — the more, the merrier.

Second, REPLEX instances can be used within the core of the network (as we did in [5], [7]); or, like MPLS LSP ingress nodes, at nodes at the network edge that have the capability to do source

routing (as in the original Wardrop model), or in any combination thereof.

Third, there is only one requirement on the performance metric that REPLEX tries to optimize: It must be possible to calculate this metric in a distributed fashion, i.e., through individual measurements from different points in the network that can be aggregated in a distance-vector like fashion. This is a very loose requirement, as it holds for virtually all metrics that are of interest to network optimization, e.g., min, max or mean of packet losses, link utilization, remaining capacity, delay, throughput, signal strength, reliability, trust level, financial revenue, etc. Moreover, it is possible to combine multiple parameters into a common metric, e.g., using a product, a weighted sum, a min/max comparison, or some other kind of combination operations. This flexibility allows to use REPLEX for pursuing completely orthogonal or even conflicting goals with different weights or priorities in parallel, e.g., minimizing packet loss, maximizing throughput, minimizing delay, maximizing reliability, etc.

IV. REPLEX IN A FUTURE INTERNET

We have seen that REPLEX features great flexibility, which implies a quite universal applicability. In particular, this makes it ideally suited not only as a mechanism for performing dynamic traffic engineering in today's Internet, but also for an Internet of the future, possibly even as an additional distributed flow control mechanism. As of now, the networking community are far from having reached a consensus on what "the Future Internet" will actually be—it is not even clearly defined what this fuzzy term actually means (e.g., "revolutionary" clean slate vs. "evolutionary" incremental approaches). Nevertheless, we conjecture that REPLEX can be of great use in a future Internet, whatever it may look like.

From the rapidly increasing number of publications that have the label "future Internet" attached to them, we will outline some of the most prominent design ideas we have come across, and we will show how they may influence and shape the application of REPLEX to such networks. In order to keep the number of pages small, we only cite papers that are either of an concluding, generalizing overview nature, or papers that are exemplaric. The publication list is thus far from being exhaustive.

A. Mobility, sensor networks

To REPLEX, neither mobility of end nodes, nor mobility of network nodes is of great concern, provided that (a) communication with neighbouring REPLEX instances is not disturbed to a great extent and that (b) the underlying routing protocol provides a reasonably good service without drastically changing the network topology (and thus the multipath destination routes offered to REPLEX) too often.

However, it is possible to include network parameters that are typical of mobile networks into the calculation of the metric that REPLEX strives to optimize, e.g., signal quality or connection reliability. In the case of sensor networks or other networks with specific demands, it may make sense to include parameters such as power consumption and remaining battery capacity.

Furthermore, the increased mobility may result in an increased penetration of multi-homed end hosts, e.g., through different access networks [3]. This implies a greater number of route alternatives, which in turn is beneficial to REPLEX.

B. Cross-layer design, disappearance of layers

As the classical "hourglass" model with IP being the wasp's waist has grown more and more "love handles" over the years (e.g., MPLS, IPSec, DNS, etc.), some propose to either weaken the strict separation of the layers ("cross-layer design"), or to abandon the strict layer

concept altogether and replace it by some kind of heap of more or less arbitrarily pluggable building blocks (e.g., [4], [9]).

REPLEX can operate in these network architectures without problem, provided that some infrastructure module provides it with multipaths. Cross-layer technologies or additional measurement building blocks can provide it with more accurate information, which can help improve its decisions. Moreover, REPLEX also can be integrated as a separate service module that allows multi-path congestion control for, e.g., only specific flows or applications.

C. Quality of Service

Although many network researchers have been focusing on QoS issues for quite a long time, QoS applications in the Internet still are scarce, in spite of obvious potential benefits to infrastructure providers, service providers, and end users. A future Internet may thus deviate from today's best-effort design to achieve better, possibly even end-to-end control on Quality of Service (e.g., [1], [11]).

Our current REPLEX implementation groups individual packets into flows based on hash values of IP headers, so as to avoid reordering of packets within individual TCP connections (which would severely degrade TCP performance). Of the flows headed to some destination prefix, it accordingly assigns a certain subset of to one route alternative, another subset to another to another one, etc., instead of relying on purely statistical multiplexing of packets. Thus on the one hand, REPLEX may profit from QoS through the additional information that is available on the flows, e.g., through call admission data. This information allows REPLEX to make more accurate traffic splits between the route alternatives. On the other hand, in contrast to classical QoS approaches, REPLEX itself is not designed to yield performance guarantees. A possible solution, however, is to construct a QoS-aware optimization metric for REPLEX that would allow to guarantee specific service properties.

D. Flow routing

For a number of reasons, among them the increasing penetration of virtual networks, QoS requirements, optical switching and the emergence of quantum cryptography, flow switching or connection switching is discussed as an alternative for a future Internet over today's packet switching architecture [11]. Here, similar considerations apply as in the case of QoS mentioned above: Having clearly defined flows with more or less pre-defined properties can help REPLEX to better tune the route weights; so again, having more information on the flows can help improving the actual accuracy of the split ratios. However, if only a small number of flows is available, then it may be hard or even impossible for REPLEX to assign the flows to the offered route alternatives in the calculated ratios, especially if the number of flows is smaller than the number of alternative routes.

E. Resilience

Even though today's Internet can cope with link and node failures to some extent, it is not really resilient. Reaction mechanisms to failures are slow (OSPF: 100msec) to unbearable (BGP: several minutes); TCP connections are torn down if a link is temporarily unavailable and thus an ICMP "host/network unreachable" message is generated, even if it is only for a short time. Reliability and resilience mechanisms thus should be integrated into the design of a future Internet [12], [11].

REPLEX can help improving network resilience in two ways: First, due to its online traffic engineering properties, it can tackle temporary traffic bursts that are caused through statistical fluctuations or flash crowds—phenomena that are very unlikely to disappear in

a future Internet, since also a future Internet still will not be built for deterministic machines, but for non-deterministic non-linear humans. Second, using a metric that captures reliability may help to keep the majority of the traffic within more reliable parts of the network, even if it subject to continuous changes (e.g., ad-hoc wireless networks).

However, using REPLEX as the sole mechanism to integrate resilience concepts into a network is probably not enough: As mentioned, REPLEX' weight ratio adjustments do not take place instantaneously, but need a little time to emerge. Therefore, very sharp bursts that result in a sudden drastic change in the network metric (e.g., a suddenly very high packet loss rate) will not be fully dealt with immediately, but only after a short adaptation time span.

F. Security

When using obvious optimization metrics such as packet loss rates, REPLEX instances have to trust each other's reports. However, the possibility exists to include a security measure such as, e.g., "trust", into the optimization metric. This way, one can try to harden the protocol against adversaries that try to influence decisions of other REPLEX instances by injecting false information.

G. Business considerations, Tussle Space

So far, our evaluation scenarios investigated cases where REPLEX is used only within one single administrative entity. In contrast, mechanisms in a future Internet should take into account the so-called "tussle space", i.e., the various and partially contradicting interests of individual stakeholders of the network landscape [2]. By carefully designing the optimization metric to be used, REPLEX can be adapted to such requirements, e.g., through integration of business-related metrics similar to BGP policies. Moreover, its distance-vector like protocol allows to exchange relevant information on network conditions across network boundaries, without having the fear of leaking other information on such as the network topology to other networks that are potential competitors.

H. Self-*, Auto-* properties

With its plethora of protocols, hosts, and organizational units, the Internet is becoming increasingly complex to administrate. This also holds for its subsets, e.g., single ISP networks. It is thus vital that network operators do not get lost in micro management tasks, but obtain the ability to define general goals and policies on a more abstract level, while the network infrastructure configures and optimizes itself automatically [11]. In this regard, REPLEX may play an important role, since it does not require any attention, apart from initially setting up some global parameters [7], [5].

I. Locator-ID split, overlay networks, virtualization

Neither of these concepts affects the operation of REPLEX: First, it does not care whether the locators of the presented information packets (or flows) also serve as identifiers. Second, REPLEX is not restricted to operate on the IP layer (which actually is itself an overlay network on top of the link layer)—it may as well operate layers below or above IP. It is, however, subject to further research how REPLEX-enabled networks on different overlay levels may affect each other's decisions if they are not aware of each other [10].

V. CONCLUSION

We have presented REPLEX, an algorithm that can be used for dynamic traffic engineering, as well as for other dynamic network optimization purposes. Its genericity and wide spectrum of possible optimization metrics make it ideally suited for incorporation into the

Internet of the future. For further information on the algorithm, we refer the reader to [5], chapters 6 and 7 in [7], and the REPLEX home page [8].

VI. ACKNOWLEDGEMENTS

I thank Georg Carle, who influenced the structure of this paper, and without whom this paper would not have been written. I also thank Simon Fischer, with whom I developed REPLEX.

REFERENCES

- [1] H. Che, W. Su, C. Lagoa, K. Xu, C. Liu, and Y. Cui. An integrated, distributed traffic control strategy for the future Internet. In *ACM SIGCOMM Workshops*, 2006.
- [2] D. D. Clark, J. Wroclawski, K. R. Sollins, and R. Braden. Tussle in cyberspace: defining tomorrow's internet. *IEEE/ACM Trans. Netw.*, 13(3):462–475, 2005.
- [3] T. Dreibholz and E. Rathgeb. Towards the future Internet—a survey of challenges and solutions in research and standardization. In *Workshop on Visions of Future Network Generations (EuroView)*, 2007.
- [4] R. Dutta, G. N. Rouskas, I. Baldine, A. Bragg, and D. Stevenson. The SILO architecture for services integration, control, and optimization for the future Internet. In *Proceedings of IEEE ICC*, 2007.
- [5] S. Fischer, N. Kammenhuber, and A. Feldmann. REPLEX—dynamic traffic engineering based on Wardrop routing policies. In *Proceedings of ACM CoNext*, Lisboa, Portugal, 2006.
- [6] S. Fischer, H. Räcke, and B. Vöcking. Fast convergence to Wardrop equilibria by adaptive sampling methods. In *Proc. 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 653–662, Seattle, WA, USA, May 2006. ACM.
- [7] N. Kammenhuber. *Traffic-Adaptive Routing*. PhD thesis, Technische Universität München, Germany, 2008.
- [8] N. Kammenhuber et al. REPLEX home page. <http://www.net.in.tum.de/~hirvi/replex/>.
- [9] B. Reuther and J. Götze. An approach for an evolvable Future Internet architecture. In *1st Workshop on New Trends in Service and Networking Architectures*, 2008.
- [10] S. Seetharaman and M. Ammar. On the interaction between dynamic routing in the overlay and native layers. In *Proceedings of IEEE INFOCOM*, 2006.
- [11] M. Siekkinen, V. Goebel, T. Plagemann, K.-A. Skevik, M. Banfield, and I. Brusci. Beyond the future internet—requirements of autonomic networking architectures to address long term future networking challenges. In *FTDCS '07: Proceedings of the 11th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 89–98, Washington, DC, USA, 2007. IEEE Computer Society.
- [12] H. Wang, Y. R. Yang, P. H. Liu, J. Wang, A. Gerber, and A. Greenberg. Reliability as an interdomain service. *SIGCOMM CCR*, 37(4):229–240, 2007.