Available online at www.sciencedirect.com
SCIENCE @ DIRECT®
ELSEVIER
Computer Communications 28 (2005) 1692–1699
computer communications
www.elsevier.com/locate/comcom

Configurable software-based edge router architecture

Wajdi Louati*, Badii Jouaber[1], Djamal Zeghlache[2]

Wireless Networks and Multimedia Services Department, Institut National des Télécommunications, 9 rue Charles Fourier, 91011 Evry Cedex, France

Received 9 February 2005; accepted 9 February 2005
Available online 28 March 2005

Abstract

This paper explores and proposes the use of open programmable router technologies to achieve dynamic configuration, adaptation and management of network entities. The objective is to provide highly dynamic and flexible edge router technology to support the interconnection between emerging short range wireless technologies such as WPANs and IP based networks. To improve current software based edge router designs, the paper merges the Click framework (an open source software architecture for the forwarding plane) and the Forwarding and Control Element Separation (ForCES) principle. The proposed architecture is implemented on a real testbed based on standard PCs and open source operating systems. Results on achievable performance using these software based solutions and Click are reported. The impact on traffic flows and applications in terms of packet losses and delays is evaluated.

Keywords: Software-based router; Plane separation; ForCES; Adaptive; Edge router; Dynamic configuration; Autoconfiguration

1. Introduction

The emergence of wireless personal area networks (WPAN) involving personal devices and terminals is setting new requirements on networks in discovery, naming, addressing and networking of devices, terminals and more generally nodes constituting the user personal area network. Even if the support of WPAN and personal network (PN) services can come from various sources and ways, the role of edge routing technologies to assist WPANs and foster the emergence of new personal services and business opportunities is seen as fundamental. The availability of highly flexible edge routing technologies can ease the integration of WPAN services in legacy and future generation networks.

The edge routers could also provide a multitude of services for the WPANs including security, dynamic tunneling, discovery, name resolution, addressing, fast forwarding and possibly proxy a number of services such as transcoding, adaptation for multimedia flows and so on.

These WPAN and PN services must also be offered according to ambient conditions and context. Thus requiring high flexibility and dynamic adaptation over multiple time scales. Latency, scalability and resilience requirements are expected to increase also considerably with the emergence of these WPANs. User centricity and seamless service ubiquity are not likely to be achieved without the introduction of high flexibility and performance of edge routing technologies. These should ideally be programmable and open to adapt to context and enable personalisation of services on behalf of the users in order to fulfil user centricity requirements. Evidently, achieving this evolution will take far more than edge technology flexibility and advances alone. This paper concentrates, however, only on the possible contribution of edge technologies to this envisioned evolution.

As mentioned edge routers are expected to play a major role as they will partake in the control of most connections, sessions, QoS, security and mobility management functions but should offer at the same time enough flexibility for network managers and providers. In peer to peer scenarios, the edge routers will typically have more responsibility in supporting the WPAN and PN services and only a limited

* Corresponding author. Tel.: +33 1 60 76 45 84; fax: +33 1 60 76 4578.
E-mail addresses: wajdi.louati@int-evry.fr (W. Louati), badii.jouaber@int-evry.fr (B. Jouaber), djamal.zeghlache@int-evry.fr (D. Zeghlache).
[1] Tel.: +33 1 60 76 42 08; fax: +33 1 60 76 42 91.
[2] Tel.: +33 1 60 76 47 15; fax: +33 1 60 76 42 91.

0140-3664/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2005.02.014

interaction with transport networks governed mostly by security and service level agreements.

Moreover, to ensure the convergence of mobile and fixed networks technologies, internetworking and cooperation will mostly be achieved at the edges while fast forwarding will ensure rapidity and simplicity in core network routers.

Unfortunately, most existing routers have static architectures with networking functions rigidly implemented by vendors. A major challenge is to modularize and standardize the router architecture in order to increase the adaptability and flexibility to support new services while at the same time offering higher packet forwarding performance.

There are basically three major router architectures [1]: software based routers, Application Specific Integrated Circuit (ASIC) based routers and network processor based routers [2]. According to [1], network processor based routers are the optimal building block for next generation networks (NGN) as they provide the speed of ASIC-based routers and the flexibility of software-based routers. Both of ASIC-based and NP-based routers are based on the control and data planes separation principle [3]. Recently, many extensible frameworks have been proposed to improve software-based router development [4,5,6]. In [7], the Click router [8] is proposed as a flexible and modular software architecture to ease the design and development of the forwarding plane. However, Click is not dynamically programmable and requires a management and control plane to achieve a whole software router framework capable of deploying new services through standard interfaces.

The objective in this paper is to assess the feasibility and viability of software based solutions to design open programmable routers that not only exhibit high flexibility but also acceptable performance. This is achieved by introducing some of the advantages and paradigms of the network processor based routers into the software based technologies. We get our inspiration from the IETF ForCES (Forwarding and Control Element Separatison) [10] working group which defines a set of standard mechanisms to separate the control and forwarding planes for NGN programmable networks.

This paper describes the design, implementation and evaluation of an extensible, scalable and programmable software-based edge router architecture relying on the ForCES approach in which the data plane functionality of the router can be adapted and extended dynamically through the use of the SMP Click [9] modular framework.

The goals of the proposed model are to achieve:

- *extensibility*: through new API services that can be dynamically added at run time in the forwarding plane,
- *flexibility*: using basic control and data plane building blocks that can be configured dynamically and
- *performance*: through the implementation of fast path forwarding with the Click SMP language to produce a high performance software router that performs per-packet processing.

These introduced features while maintaining high flexibility will maximize packet-forwarding performance. Moreover, they facilitate protocol and service deployment for third party applications through standard API interfaces.

Section 2 of this paper reviews and discusses existing router architectures. The proposed software-based router model is described in Section 3. Performance evaluation results of the router implementation are reported in Section 4.

2. Challenges in building programmable IP router

Three major architectural solutions [1] are candidate to support NGN: software-based routers, ASIC-based routers and network processor-based routers.

2.1. Software-based routers

Software-based solutions [11] are made of general-purpose microprocessors (GPP). Both control and packet forwarding functions are performed by software. The main advantage of this first solution is *flexibility*. Vendors can easily introduce new features but the growing complexity of value added services, such as VPN and QoS management functions, require in addition detailed processing at the packet level. This may lead to performance degradation in the packet-forwarding plane for the software based routers.

In order to overcome the performance limitation, the other programmable router technologies separate router functions into two sets: control plane and data plane (forwarding). The control plane is implemented in software over general-purpose processors and includes the routing protocols and the network management software. The data plane is built with programmable and configurable hardware entities. ASIC-based and network processor-based routers belong to this class.

2.2. ASIC-based routers

In ASIC-based routers, the forwarding plane is implemented with hardware. This leads to higher speed *performance*. However, with ASICs there is a lack of flexibility since once they are designed, it is not possible to introduce new features and functions.

2.3. Network processor-based routers

Network Processors (NP) or ASIPs (Application Specific Instruction Processors) are expected to provide the speed of ASIC-based routers and the *flexibility* of software-based routers. They are specifically conceived for network function purposes and are supposed to be used in parallel to allow workload balancing and plane separation. NP can meet the requested scalability and *performance* requirements for next

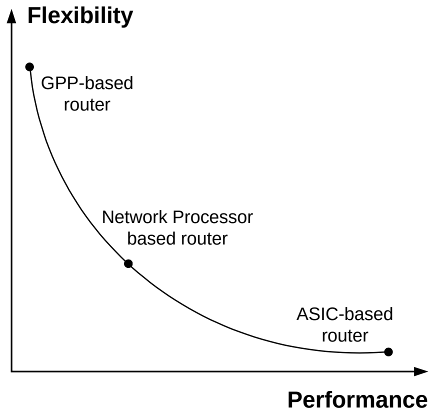


Fig. 1. Router architecture comparison.

generation routers. NP offers also the flexibility to add new services without changes in packet forwarding hardware and performs per-packet processing at line rates.

Fig. 1 gives a summary comparison between the three mentioned technologies on the basis of flexibility and performance.

As an alternative to hardware solutions, the data plane can use high performance software capable of processing packets at run time. For the same reasons, nothing prevents software based routers to adopt the same notion of separation provided high performance software performs the per-packet operations at line rates [12]. This is exactly the motivation of our work whose goal is to verify the feasibility and viability of such an approach and possibly enhance the flexibility and performance of software based routers. This will also provide open source frameworks to the community.

3. Software-based edge router architecture

This section presents our proposed model combining plane separation approach and ForCES architecture to achieve the desired flexibility and performance from software based routers. The proposed solution is based on the use of the high performance Click-SMP software language for the forwarding plane router.

3.1. Plane separation approach

The next generation building-block model breaks the traditional monolithic edge router architecture into three interdependent functional blocks connected by standard interfaces [3].

These planes are the:

- Management plane: including network management applications,
- Control plane: including routing and signaling protocols,
- Forwarding plane (or data plane): which can implement the fast path with programmable network processing units (NPUs) or high performance software.

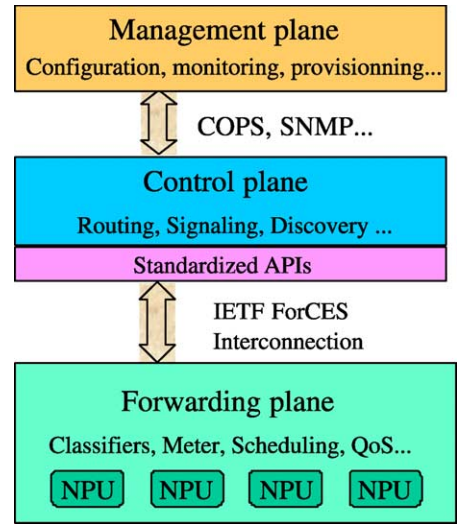


Fig. 2. Management, control and data plane separation.

In order to define, facilitate and manage interactions between these planes, the Network Processing Forum [13] has specified a set of standard interfaces called Services API (cf. Fig. 2). Using these APIs, control plane software vendors are completely independent from the implementation of data plane functions within the router thus providing design flexibility and higher reliability in the system.

3.2. IETF ForCES architecture

Due to plane separation, interconnection and independent protocols are needed to ensure communication between data and control planes.

ForCES (Forwarding and Control Element Separation) [10], defines and standardizes the required interfaces, protocols and the exchange of information between the separated planes. This standard separation mechanism allows each plane to evolve in parallel independently in a scalable and interoperable manner. As shown in Fig. 3, each Network Element (NE) (or router) is defined as a set of logical entities. Two types of NEs are defined: the control element (CE) which operates in the control plane and the forwarding element (FE) which operates in the forwarding plane. CEs include control functionality like routing and signaling protocols (BGP, RSVP, etc.), whereas FEs integrate packets processing functionality such as metering, classifying, scheduling, etc. CEs control the behavior of FEs in a master/slave mode. Communication between CEs and FEs is achieved via the ForCES protocol.

3.3. The proposed software router design based on the plane separation approach

To create the desired open programmable router architecture a packet processing mechanism is typically

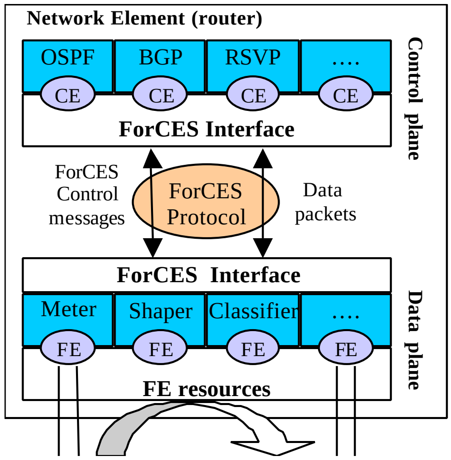


Fig. 3. ForCES architectural representation of network element.

added to the packet forwarding path in the data plane. In this work, the Click modular router [8], an extensible toolkit for packet processing on conventional PCs, has been selected to build the IP packet forwarding in the data plane (Fig. 4).

SMP Click [9] is derived from the Click modular router which provides both flexibility and high performance on multiprocessor platforms. It is a flexible software architecture for creating configurable and extensible routers. A Click router is built by assembling software components into a directed graph. Each component (or element) represents a unit of router processing such as packet classification, queuing, scheduling, and interfacing.

The SMP Click kernel runs in a multithread mode under Linux providing automatic parallel execution. It provides increased performance for packet processing. Moreover, SMP Click router can be easily extended to support new services and functionalities by inserting new elements in the Click configuration.

For instance the GNU Zebra [16] software, used for routing packets in networks, would be deployed in the control plane of the Click extensible router. This causes, however, a problem of robustness, flexibility and performance for the Zebra router because it would not be able to handle all routing requirements in the forwarding plane at run time. To circumvent these weaknesses and ensure flexibility and performance for this router while maintaining interoperability between the separate planes, the ForCES architecture is integrated into the proposed router. This is achieved by deploying the ForCES forwarding elements (FEs) into Click elements (see Section 3.3.2 for details). The Zebra routing daemon cooperates with the ForCES control elements (CEs) to achieve the desired routing functionalities. As depicted in Fig. 4, the CEs and FEs communicate with each other by running the ForCES protocol.

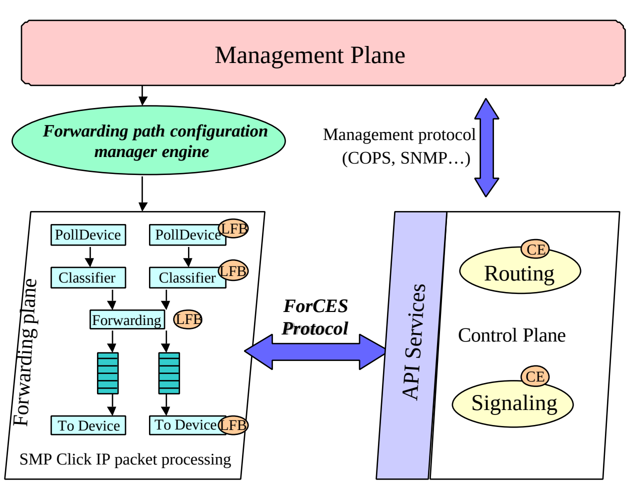


Fig. 4. Design of a software base router with ForCES architecture.

For the practical side of this router, the control and data planes are physically separated boxes, interconnected with a high speed LAN connection such as a Gigabit Ethernet [10].

The proposed router design of Fig. 4 provides the desired features of flexible, good performance and scalable implementations as in the previous example of the Virtual Router (VR) concept used for Network based IP VPN Architectures [17]. A virtual router (VR) is an emulation of a physical router in the software layer which handles VPN connections. Many instances of VRs may be running on a single router with each VR having independent routing and forwarding tables isolated from each other. Two main functions are performed by a virtual router. The determination of the appropriate path between VPN sites via routing tables constructed using any routing technologies (like OSPF, BGP, RIP). The second function consists of forwarding packets to the next hops within the VPN domain. To isolate traffic between VPNs separate routing and forwarding capabilities for each VR are necessary. The ForCES architecture and the proposed router design can be a solution to implement a virtual router concept that achieves dynamic VPN tunnel establishments for personal networks.

3.3.1. Dynamic and automatic data plane configuration

The major drawback of Click is that router configuration, once installed, cannot be changed at runtime. In order to modify some parameters or to insert elements, a complete configuration change is necessary. Taking into account the frequent changes that may happen at the edge router (Mobility, QoS provisioning, security, etc.) in the context of evolving access networks, the network manager cannot intervene, each time, to modify manually the Click router configuration by uninstalling the old and full Click script and reinstalling a new one.

In order to enhance the dynamic Click configuration functionality and to overcome this manual configuration problem, a 'forwarding path configuration manager engine' (FPCME) has been developed with Perl scripts. This engine can achieve automatic run time configuration of a Click router and can control interaction between routing protocols and the data plane through the ForCES protocol and associated API services (Fig. 5).

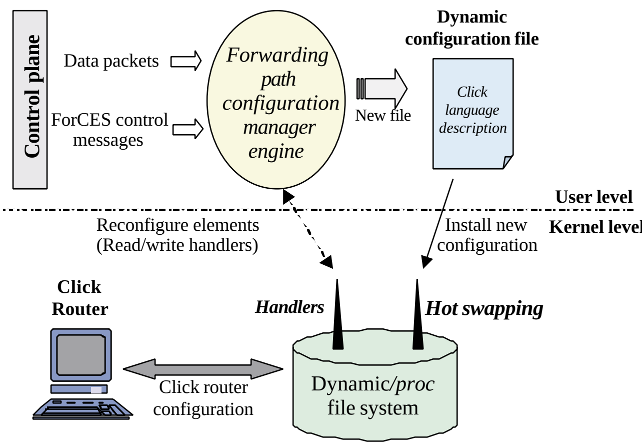


Fig. 5. Dynamic forwarding path configuration manager.

The FPCME is responsible for determining and installing the appropriate combination of Click elements that fulfill the requirements (QoS, Mobility, etc.) specified by the control and management planes. Thus, it can dynamically insert, remove and replace router components or modify their associated parameters. This is possible since the Click kernel uses the dynamic Linux '/proc' file system to communicate with user processes.

As shown on Fig. 5 the FPCME analyses messages sent by the control plane such as ForCES control messages and data packets including management rules, monitoring information and routing instructions. These parameters are transcoded into a set of click elements and arguments that can be added to or removed from the running click kernel driver. Thus, the FPCME creates new configurations of the Click router and installs them into the '/proc/click/hot-config' with a 'hot-swapping' option. Therefore, part or the entire router configuration can be modified at runtime including connection changes into the click graph and even dynamically extend the number of ports provided by an element's configuration string. This resolves the fixed port number problem of some click elements such as 'Classifier' elements.

In addition, since Click kernel level includes 'Handlers' that provide access to parameters for read or write operations on elements, it is possible to completely automate the router configuration. At user level, handlers may be accessed via a TCP/IP-based protocol.

The developed forwarding path configuration manager interface thus enables dynamic adaptation through the 'hot-swapping' option and configuration through the 'Handlers'.

3.3.2. Expected data plane design

Our on going proposal for enhanced data plane design relies on the FE model of the IETF ForCES [14]. The FE model includes classes called LFBs (Logical Function Block) representing basic building blocks that represent a logical and distinct packet processing operation in the data path. With this model, it is possible to configure the attributes of each LFB and possibly enable dynamic extensions of the router data forwarding behavior using the LFB classes library. To achieve this evolution of the router architecture, these logical function blocks (LFBs) will be integrated within Click elements. In fact, LFB classes define all of the information to exchange between FEs and CEs via the ForCES protocol. Thus, using this model, Click elements in the data plane can be connected with the routing protocol in the control plane. This proposed inherent and embedded capability is missing in most programmable router solutions. Enabling communications between the two planes via the ForCES protocol should enhance the capabilities and flexibility of software based programmable routers and ease the introduction of new VPN, QoS and overlay services.

4. Testbed and performance results

4.1. Testbed description

In order to validate and evaluate our proposal, a testbed[3] has been setup as depicted in Fig. 6. The testbed includes a traffic transmitter, a traffic receiver, a network monitoring system and a Bandwidth Broker (BB).

The core network is composed of meshed Click routers running over Gigabit Ethernet[4] links. Two multiprocessors edge routers using SMP Click are used in the installed platform.[5] The network is DiffServ enabled.

4.2. Performance and results

The objective of the testbed is to evaluate the impact of dynamic and automatic configurations performed by the FPCME, in the open programmable routers based on SMP Click, on applications and traffic flows. The first experiment concerns a policy-based DiffServ traffic management with dynamic configuration of the policies. The objective is to configure the routers dynamically according to SLA changes initiated in the Bandwidth Broker (BB) domain.

The establishment of Service Level Agreements (SLA) for traffic management is achieved using the COPS (Common Open Policy Service) framework that has been integrated in the management plane of the Click edge router.

The AAA (Authentication, Authorization and Accounting) Server first authenticates the user. The BB verifies after authentication the availability of resources and performs the allocation procedures. Once the resources are allocated, the policy decision point (PDP) sends the user SLA parameters to the PEP (Policy Enforcement Point) in the edge router

[3] The testbed is partially funded by RESACO, a research project financed by the GET (Groupe des Ecoles de Télécommunications) and the PERCEVALE project.

[4] Intel PRO/1000 Gigabit cards, in a 64 bits slot, 100 MHz.

[5] Intel XEON Processor MP, 1.5 GHz.

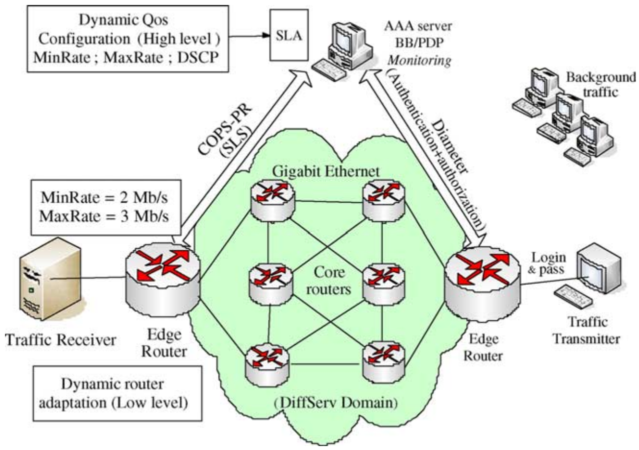


Fig. 6. Dynamic adaptation and configuration of edge routers.

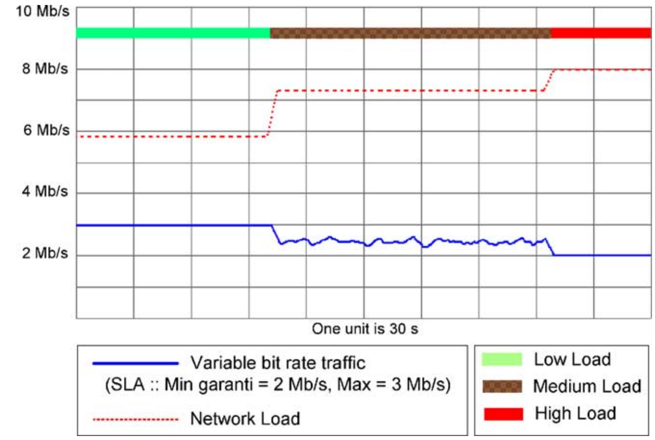


Fig. 8. Dynamic behavior of an auto-configuration Click.

through the COPS-Provisioning protocol. The decisions received by the COPS PEP are installed on the router through the FPCME that transcodes rules and modifies the appropriate parameters in the forwarding plane. Dynamic modification of the SLA when requesting a configuration change in the edge router was successfully implemented and accomplished. The FPCME successfully modified the DiffServ Meter (represented by a Click element) parameters without interrupting the router functionality. The Click Handlers were used to modify the Meter parameters.

A second experiment, focussing on auto-configuration, is performed. A traffic monitor is designed and included within the proposed engine. According to the measured load, the edge router can dynamically decide what actions must be applied (Fig. 7) to guarantee each traffic flow their target QoS. The edge router dynamically chooses to accept, shape or drop out-of-profile traffic.

According to the monitored information, the proposed interface can install a new block (Dropper or Shaper) or even modify part or the entire Click router graph to allow the optimal processing of out-of profile packets. Fig. 8 shows the dynamic behavior of the auto-configurable edge router as modeled with Click.

The dotted curve represents the network load. The solid line curve correspond to a variable bit rate traffic that is associated with a service level agreement with a minimum guaranteed throughput of 2 Mbps and a maximum throughput of 3 Mbps. As shown on Fig. 8, the edge successfully adapts its different DiffServ parameters and mechanisms (shaper, policer, etc.) according to the traffic conditions. The solid line curve shows that the variable bit rate is accepted at maximum throughput when enough resources become available, policed at 2 Mbps when the router is highly loaded and shaped when load conditions are intermediate.

The effects of run time configuration actions on both TCP and UDP flows have been evaluated and found negligible in terms of delays and packet losses at the edge router.

For each given configuration with hardware and driver limitations on Gigabit Ethernet cards and limitations of the Click router, a Maximum Loss Free Forwarding Rate (MLFFR) is defined and measured to assess performance [8].

The first experiment aims at estimating this MLFFR for the Click router. A single UDP flow produced by the traffic generating tool *Iperf* [15] is used to evaluate packet loss. The results depicted in Fig. 9 show no packet losses for throughput below 150 Mbps or equivalently 13,280 packets per second (packets are 1412 byte datagrams). For throughputs higher than 150 Mbps, less than 10^{-4}% of the packets are lost for rates up to 200 Mbps. These losses are due to hardware

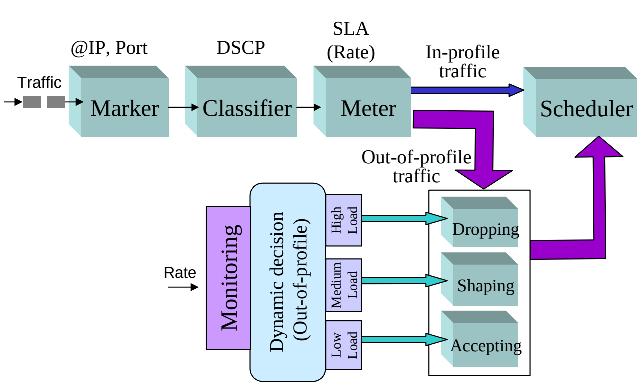


Fig. 7. Dynamic decision in the edge router.

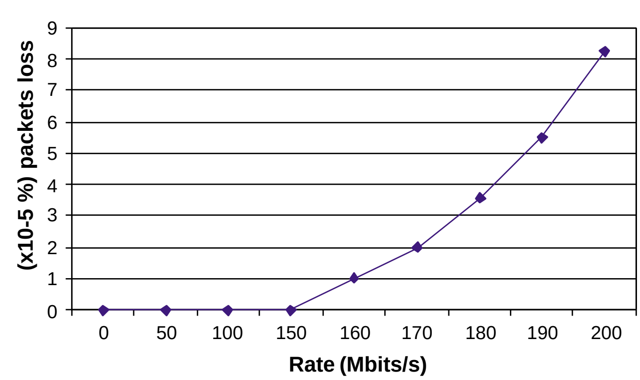


Fig. 9. Losses versus aggregate rate.

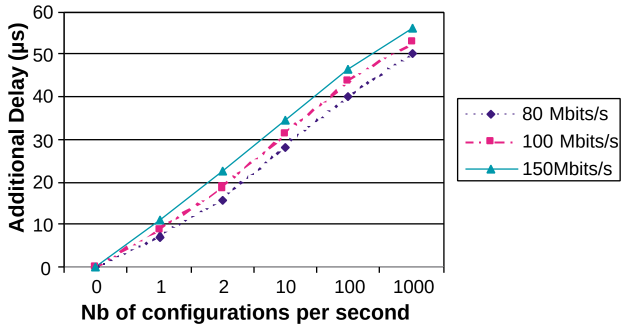


Fig. 10. Additional delay caused by dynamic configurations.

limitations and compatibility issues between the CPU and the Ethernet card memory rather than the Click router.

A second experiment enables estimation of the additional delay induced by the dynamic configurations through the FPCME of the Click router on the data flows. This additional delay was evaluated at the ingress and egress port of a single router. As shown on Fig. 10, the delay grows linearly with the dynamic configuration frequency. However, the average packet delay remains below 58 µs for throughputs below 150 Mbps. For these conditions, the configuration interface and the actual configuration actions within the Click router affect only marginally the end to end delay and cause no packet losses.

The last experiment addresses the effect of dynamic configurations, at run time, on the end-to-end delay at higher operating rates by using 10 parallel TCP flows with an aggregate throughput of 680 Mbps. Each flow corresponds to the transfer of a 67.7 M bytes file. Dynamic configurations on the router are applied to all the ongoing flows. The average RTT is measured on an additional Ping flow running in parallel with the above-mentioned TCP flows (Fig. 11). Even though the reported test concerned aggregate flows, note that configuration actions on a given flow using Click does not affect any other flow.

As depicted in Fig. 11, the average RTT per packet increases from 684 to only 698 µs when the number of configurations grows from 0 to 1000 per second (note that zero configuration corresponds to a static router that serves

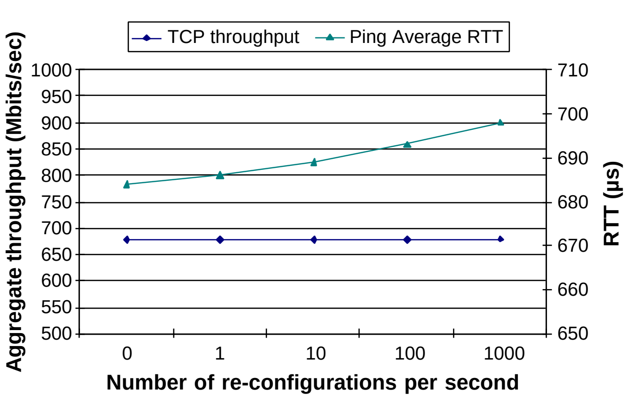


Fig. 11. Effect of dynamic configurations on the round trip time.

as a reference for the RTT evaluation). This 2% increase in RTT represents a marginal degradation for the TCP flows. This result indicates that the Click router technology, used to achieve dynamic configuration and adaptation in edge devices, will induce only negligible additional delays on traffic flows.

5. Conclusion

This paper reports on the feasibility and viability of using open programmable edge routers based on the ForCES architecture and the SMP Click language to achieve dynamic configuration and adaptation in next generation networks. Performance results indicate that the impact on TCP and UDP flows is marginal when conducting dynamic configurations in the edge routers at run time. The experienced marginal delays, packet losses and jitter are a favorable sign for the use of software based routers using the separation principle and the SMP Click language. Larger scalability studies should comfort these findings and foster the use of these routers.

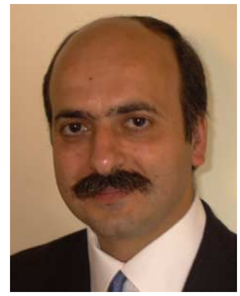
References

[1] Network Processors: The Optimal Building Block for Next Generation IP Routers, Alcatel, November 2000.
[2] T. Spalink, S. Karlin, L. Peterson, Y. Gottlieb, Building a Robust Software-based Router Using Network Processors, Proceedings 18th ACM Symposium on Operating Systems Principles (SOSP), Chateau Lake Louise, Banff, Alberta, Canada, October 2001 pp. 216–229.
[3] H. Hegde, D. Larson, IPv6 router solution using intel internet exchange architecture, Intel Developer Magazine March (2002).
[4] D. Decasper, Z. Dittia, G. Parulkar, B. Plattner, Router Plugins: a Software Architecture for Next Generation Routers, Proceedings of ACM SIGCOMM 1998.
[5] XORP: Extensible Open Router Platform. http://www.xorp.org/
[6] D. Mosberger, L.L. Peterson, Making Paths Explicit in the Scout Operating System, in: Proceedings of the Second USENIX Symposium on Operating System Design and Implementation (OSDI), Seattle, WA, USA, October 1996, pp. 153–167.
[7] Y. Gottlieb, L. Peterson, A Comparative Study of Extensible Routers OpenArch'02, June 2002.
[8] E. Kohler, R. Morris, B. Chen, J. Jannotti, M.F. Kaashoek, The click modular router, ACM Transactions on Computer Systems 18 (3) (2000) 263–297.
[9] Benjie Chen, Robert Morris, Flexible Control of Parallelism in a Multiprocessor PC Router, Proceedings of the 2001 USENIX Annual Technical Conference (USENIX'01) 2001 pp. 333–346.
[10] L. Yang, R. Dantu, T. Anderson, R. Gopal, Forwarding and Control Element Separation (ForCES) Framework, RFC 3746.
[11] X. Qie, A. Bavier, L. Peterson, S. Karlin, Schedulling Computations on a Software-Based Router, Proceedings of SIGMETRICS 2001, June, 2001.
[12] David M. Putzolu, Software API Framework, Implementation Agreement, Revision 1.0, NPF Software Working Group.
[13] Network Processing Forum, http://www.npforum.org/

[14] L. Yang, et al., ForCES Forwarding Element Model, Work in Progress, February 2004, <draft-ietf-forces-model-02.txt>.
[15] http://dast.nlanr.net/Projects/Iperf/
[16] GNU Zebra Open-Source Routing Software, http://www.zebra.org/
[17] Paul Knight, et al., Network based IP VPN Architecture using Virtual Routers, April 2004, <draft-ietf-l3vpn-vpn-vr-02.txt>.

M. Wajdi Louati received the BS Degree in Computer Science in 2002 from National Engineering School of Sfax (Tunisia) and the MS Degree in Computer Science (2003) from Pierre et Marie Curie University (Paris, France). Currently, he is a PhD student at Institut National des Télécommunications (Evry, France). He conducts his research within the Wireless Networks and Multimedia Services Department. His main research interests include programmable routers and networks, Virtual Private Networks (VPN) and Personal Networks.

Dr Badii Jouaber holds a PhD (1998) in Computer Science. He joined Alcatel-CIT in 1999 where he contributed to the specification and optimization of the Alcatel GPRS solutions. He then moved to INT in 2002 where he is currently an Associate Professor within the wireless and mobile networking department. His main research interests include end-to-end Quality of Service, mobile networks (GPRS, UMTS), traffic management and resource optimization.



Djamal Zeghlache graduated from SMU in Dallas, Texas in 1987 with a PhD in Electrical Engineering and joined the same year Cleveland State University as an Assistant Professor. In 1990 and 1991, through a Summer Faculty Fellowship, he worked with the NASA Lewis Research Center on mobile satellite terminals, systems and applications. He continued working with NASA Lewis while at Cleveland State University until 1992 when he joined the Networks and Services Department at the Institut National des Télécommunications (INT) in Evry, France. He is currently head of the Wireless Networks and Multimedia Services Department at INT. Professor Zeghlache is an active member of the IEEE communications Society. He serves in the steering committee and the technical committee of the ASWN workshop on a regular basis since 2001. In 2003 he was the technical chair of the Wireless Communications Symposium of the IEEE Globecom 2003 Conference. Professor Zeghlache has sustained research and scholarly activities in the field of wireless networks and services. He is currently involved in the European Research and Development project MAGNET and working on advanced Wireless PAN technologies. His interests and research activities span a broad spectrum of issues related to wireless networks including radio access (resource management and planning) and core networks with emphasis on end to end issues such as cooperation between networks, awareness, reconfigurability and adaptation.