SCALABILITY ANALYSIS OF A MEDIA AWARE NETWORK ELEMENT

Marius Vochin, Eugen Borcoci, Dragoş S. Niculescu, Mihai Stanciu

Telecommunication Dept., ETTI, University Politehnica of Bucharest
e-mail:{mvochin, dniculescu, eugenbo, ms}@elcom.pub.ro

ABSTRACT

An architecture based on new concepts of Content Aware Networking (CAN) and Network Aware Applications (NAA) is proposed in FP7 ALICANTE research project as a better support for multimedia flows across the internet. A virtual CAN is a multi-domain overlay network based on light virtualization, provisioned to enable customized treatment of data flows and especially media streams. While it uses legacy QoS technologies, such as core IP/MPLS/Diffserv in the core routers, it proposes a new edge router, called MANE (Media Aware Network Element), CAN capable. This paper presents a modular MANE implementation solution using off-the-shelf hardware and open source software like Click modular router to implement flow classification, MPLS encapsulation and decapsulation, separation between VCANs, and QoS enforcement with Linux TC (traffic control) mechanism. Based on performance measurements, it is shown that the implementation imposes minor overheads over existing routing infrastructure and scalability is achievable.

Index Terms: *content-aware networking, network aware applications, quality of services, multimedia distribution, Future Internet, media-aware network element, scalability.*

I. INTRODUCTION

One of the new paradigms of the Future Internet (FI) is "content orientation", which is supposed to improve the user experience related to the new digital multimedia services and networked media content. This trend is recognized also by the European commission, which defined the "Objective ICT-2009.1.5: Networked Media and 3D Internet" in the FP7 Call 4 [3][4]. In this call new directions are defined as content-aware networks (CAN) and network-aware applications (NAA). This approach breaks (partially) the classic TCP/IP and OSI stack network neutrality and application-transport separation concepts. The challenge is to get better performance without losing modularity of the architecture. CAN-NAA means the capability of the overall system to adjust network resource allocation based on limited examination of the nature of the content, while network-awareness means to process and distribute the content, based on limited understanding of the network conditions. Dynamic optimization of network traffic flows is desired, with policies taking into account the content and adaptation needs, the user contexts, requirements and social relational network. The FI should enable multiple user roles, e.g., as content producer, user, or manager.

The work of this paper is a part of the effort inside of an European FP7 ICT research project, "MediA Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments", ALICANTE, [2][5]. An innovative architecture is proposed, for a "Networked Media Ecosystem", supporting flexible cooperation between providers, operators, and users. Three interworking environments are defined: Network Environment (NE) including Network Providers, Service Environment (SE) including Content and Service Providers and User Environment (UE) including all End-Users. The architecture validation and implementation are currently in progress and will be deployed in a large-scale international pilot.

The above environments are nowadays present in real deployments, but actually the collaboration between them is weak. The current architectures do not exchange content-based and network-based information between the network layers and upper layers. This principle was considered many years as a basic one governing the Internet, however it begins to show some disadvantages in the context of the content orientation of the FI.

The main objectives of this paper are to validate the functional capabilities and to measure the forwarding performance and other relevant scalability factors of a MANE implementation in a real testbed.

The paper is organized as follows: Section II presents some of the related work existent in the field. The ALICANTE architecture and its main concepts are defined in Section III. Section IV is focused on user and kernel space MANE implementations. Section V presents performance and scalability related measurements. Conclusions, open issues, and future work are presented in Section VI.

II. RELATED WORK

The content-aware networking (CAN) and network-aware applications (NAA) approach is a new mode to design the layered architecture, with a running debate about the benefits of better interactions as opposed to the penalty of losing modularity of the architecture.

In such a context, both CAN and NAA are of interest both for research communities and industry, in the process of re-thinking the architecture of the FI.

The capability of content-adaptive network awareness to offer optimization for video transmission is analyzed in [6]. In [7], it is considered that CAN and NAA can offer a way for evolution of networks beyond IP.

In [8], it is discussed how the CAN/NAA approach can lead to a user-centric FI and telecommunication services. The content adaptation issues in the FI as a component of CAN/NAA approach is discussed in [9].

The better QoE/QoS capabilities of the CAN/NAA architecture is analyzed in [2], [10]. Further gains are obtained if context awareness is also considered [11], [12].

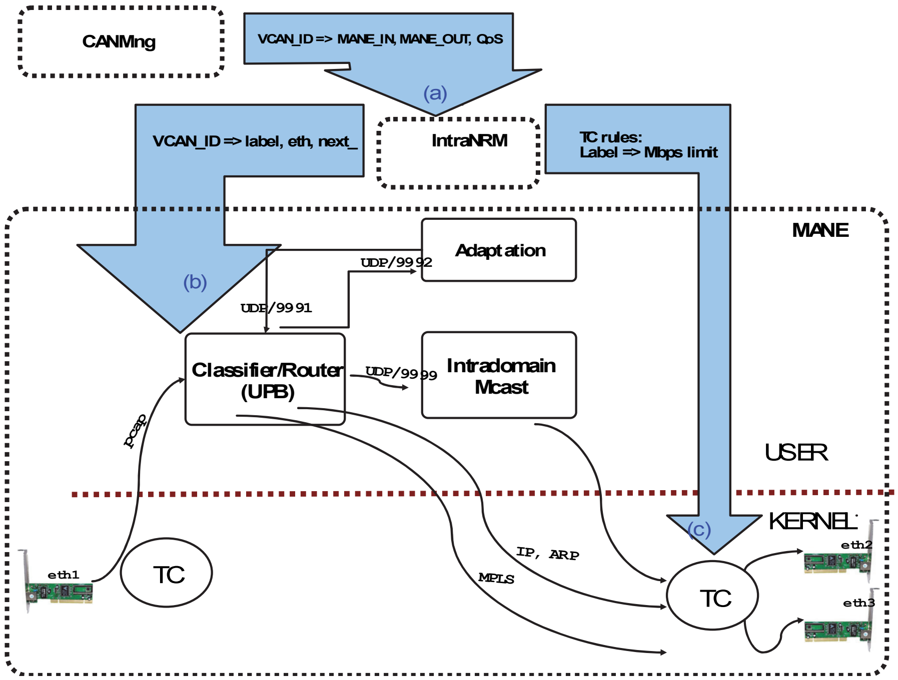

Figure 1. The ALICANTE Architecture: details on Virtual CAN Layer

The application layer traffic optimization (ALTO) problem defined by the IETF can be solved by the cooperation between the CAN layer and the upper layer, as in [13], [14].

However, no complete and open architecture currently exists, able to support multimedia distribution according to the CAN principles and scalable over sizeable networks and heterogeneous networking technologies.

III. ALICANTE SYSTEM ARCHITECTURE

A. Layers and entities

The ALICANTE architecture promotes concepts such as content-awareness to the network environment, user context-awareness to the service environment, and adapted services/content to the end-user for his/her best service experience while being either a consumer and/or producer.

Two new virtual layers are proposed on top of the traditional network layer: the CAN layer for network level packet processing and a Home-Box (HB) layer for the actual content delivery.

The CAN layer offers an enhanced support for packet payload inspection, processing and caching in network equipment. It is developed over traditional IP network/transport layer. It will improve data delivery via classifying and controlling messages in terms of content, application and individual subscribers; it improves QoS assurance via content-based routing and increases network security level via content-based monitoring and filtering. In such a way, content and application-aware networks are created to provide high levels of performance, end-user experience, and to enable application and subscriber-specific data forwarding. The specific components in creating the CAN layers and performing CA processing are the Media-Aware Network Elements (MANE), i.e., the new CAN routers under control of CAN managers.

The Home-Box layer is an upper layer, using CAN services and taking into account network-aware information delivered upward by the CAN layer. Thanks to this layer, inter-working with the User, Service, and Network Environments, one can elaborate network and context-aware applications and deliver the necessary inputs to create content-aware networks. The adaptation, service mobility, security, and overall management of services and content are being assured at this layer through a new specific middleware proposed by the project, working in conjunction with the other layers.

The upper SE layer uses information delivered by the CAN layer and enforces network-aware applications procedures necessary to perform the adaptation of the media resources to the user's preferences.

The main management and control entity in the CAN layer is the CAN Manager (CANMgr). Its main task is to create VCANs (seen as parallel logically isolated planes) on demand of the SP through negotiation concluded in Service Level Agreements/Specification (SLA/SLS). Then VCANs will be installed in the networks.

To preserve the autonomy of core network domains but create a multi-domain context for VCANs, each Autonomous System has associated one CANMgr, and an Intra-domain Network Resource Manger (Intra-NRM). The CANMGr has the following roles: to (re)define the CANs (according to the enhanced connectivity service targeted) and perform all related actions to configure, maintain and update CANs; to advertise and negotiate the CAN usage with upper layers, using SLA/SLS contracts; to communicate with other CAN managers in order to establish multi-domain VCANs, again, using SLA/SLS contracts; to communicate with its own IntraNRM. The IntraNRMs have the ultimate authority upon the network provider resources, thus conserving each domain's independency.

B. The Content-Aware Network Router

The MANE, a content-aware network router, is an intelligent network node. It performs appropriate processing (routing, filtering, adaptation, security operations, etc.) considering content type and properties, ,described by special metadata inserted in the data packets by the Content Servers named Content Aware Transport Information-CATI, or extracted by protocol field analysis, and also depending on network properties and network status. The MANES are instructed what to do with different flows via the Management and Control Plane by the CAN Managers, via Intra-NRMs. The results of the content related information analysis provide metrics, which allow to decide the best strategy to adopt for the best content repurposing and publishing methods. The MANE basic set of functions are:

Content-aware routing and forwarding: the MANE will decouple the higher level routing process from the lower level forwarding. The routing is a QoS constrained process, establishing appropriate inter or intra-domain paths, associated with VCANs. The forwarding is done after packet classification, based on CATI analysis if it exists or deep packet inspection (DPI) if conventional Content servers are used not capable to insert CATI.

Content-aware QoS and resource allocation: the MANEs will be able to deduct the QoS requirements of different flows based on the flows content. The current status of the CANs will be monitored in the CAN layer. The MANE will maintain an aggregated image of flows that they forward, and for every recognized flow type, a VCAN will be assigned, depending on the level of QoS guarantees and network status. This will optimize resource allocation in the network, depending on traffic types and QoS requirements. The CAN level will interact with the domain network resource management in order to perform mapping onto different L2/L3 QoS-aware technologies (e.g., MPLS/Diffserv or Carrier Ethernet). Some amount of relatively infrequent dynamic re-allocation of the network resources between different CANs is possible, optimizing resource usage. The MANE has also an adaptation role, deployed at different points in the delivery chain: at the service creation, during the transport by the CAN routers, and at the Home-Box site;

An issue, beyond privacy, which is addressed by using special fields and/or metadata to describe the content, is the processing time required by deep packet inspection; eliminating the need for this procedure will significantly improve the performance of the CAN-enabled routers.

IV. MANE High Level Architecture

The main interactions of the MANE are presented in Figure 1. The control plane interactions are indicated with thick shaded arrows, and the data path is indicated with thin arrows. From above, the Intradomain NRM has the role of providing the means to create FEC associations for entry in the MPLS domain. Each packet is marked with a VCAN header by the generating HB/SB, and a path is decided through CAN Manager – IntraNRM collaboration. The type of VCAN and the entry into the MPLS tunnel are then provided for each MANE router. The core of the MANE is a classifier/router module which identifies incoming traffic based on its VCAN header, encapsulates it into the MPLS header, and sends it to the appropriate LSP.
The architecture is completely modular so that

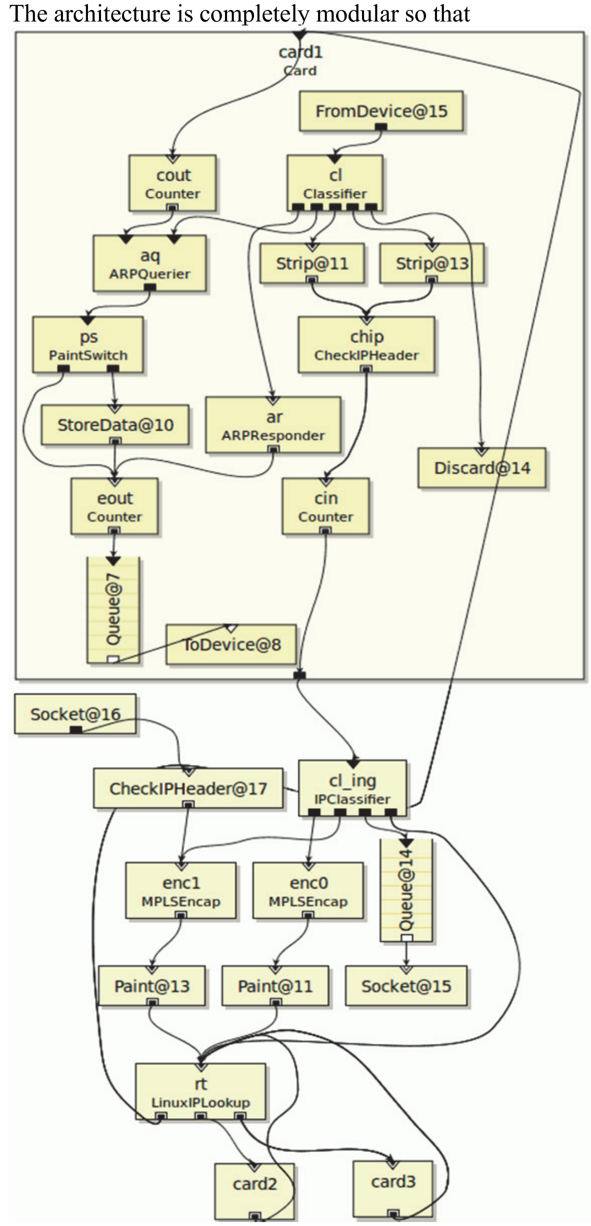

Figure 2. Userspace MANE classifier/router implementation using Click elements

functionality can be developed in parallel. Modules all run in user space and are interconnected using UDP/IP so they can run on either the same, or on different machines. Another advantage is that it is easy to reconfigure the architecture by simply changing the UDP ports, or by inserting new modules.

The central module is the Classifier/Router, which harvests packets from the incoming interface and distributes them to the other modules, or encapsulates them into the MPLS paths. The classifier needs to know the association between VCAN IDs present in all incoming packets so that it can route traffic to the appropriate VCAN. VCANs are assumed to be configured in advance by the CAN Manager and provisioned through the IntraNRM. In particular, the MANE needs to be explicitly instructed by the IntraNRM on the association between the VCAN_ID and a MPLS label to be used (shaded arrow marked b)). This module also decapsulates MPLS traffic that comes from the domain, before forwarding it to the appropriate HB/SB.

The core router part is not represented here, but we assume it has complete MPLS support and is implemented either with specialized hardware, or with Linux machines requiring a specially patched kernel. The IntraNRM manages al the labels and the bandwidth provisioning for each path. In fact, bandwidth provisions are sent down to both MANEs and core routers to be enforced, perhaps with *tc* functionality (marked as shaded arrow c).

The Classifier/Router module is implemented both in user and kernel space and uses Click modular router [15], as shown in Figure 2. For close to Linux performance it could be moved down into the kernel space in the final phases of development. It performs MPLS and IP routing both in and out the domain. It performs FEC associations for incoming IP traffic, and MPLS decapsulation for traffic outgoing to HB. It dispatches traffic to local modules (Adaptation, multicast, etc), but also accepts traffic from them so they don't need to handle routing or encapsulation tasks. The convention in implementing the MANE is that eth0 interface is used for testbed support and therefore not part of ALICANTE.

Interfaces eth1, eth2, eth3 … are used either as ingress into, or egress from the MANE. The main classification task is performed by a dedicated classifier element, called cl_ing (for ingress traffic), that identifies the VCAN of the incoming packet and uses the appropriate MPLS label to decide the policies for forwarding, shaping and policing. The elements grouped in the element class Card handle all the bookkeeping necessary to IP and MPLS to exchange packets on the local network (for card2 and card3, the internal details are omitted).

V. EXPERIMENTS

Using the MANE implementation described in the previous section, we built a topology comprising three HBs, three MANEs, and two MPLS core routers.

All the elements in the topology run Ubuntu Linux 10.04 LTS x64 on quad CPU Intel Xeon W3520, 6GB memory, Gigabit Ethernet Intel Pro/1000 82574L and Realtek 8111 PCIx Ethernet cards. We measured performance in four different routing configurations: using standard IP, using kernel MANE MPLS, then using user MANE IP and MPLS. The results are summarized in the table below:

For the RTT tests, we used ping with large and small packets. The MANE implementations brings a minor increase in end to end transit time and a slight increase in the standard deviation of RTT. For throughput measurements, test traffic was generated and measured in user-space using the *Iperf* traffic generator. The data rates achieved with the MANE implementations are comparable to plain Linux data rates, but packets per second throughput offered by kernel MANE implementation is slightly less. User MANE implementation offers considerably less throughput; part of this difference can be accounted by the current implementation of all modules in user space.

In the following test scenarios Linux Traffic Control functionalities where used in order to determine QoS performance of our MANE implementation.

	IP	Kernel MANE	User MANE, MPLS	User MANE, IP
ping 32 byte pk RTT/stddev [ms]	0.599/ 0.032	0.764/ 0.058	0.776/ 0.043	0.770/ 0.059
ping 1460 byte pk RTT/stddev [ms]	0.575 0.029	0.740 0.063	1.75/ 0.78	0.900/ 0.064
UDP Rate [Mbps]	906	899	899	900
TCP Rate [Mbps]	870	856	845	
Packet rate [pps]	482000	446000	260000	280000

In order to test MANE QoS capabilities, we started one *iperf* process that generates test traffic (about 1Gbps) for 10 seconds and an increasing number of queues were installed, each one with a single associated u32 filter, inspecting the destination port only.

When test traffic entered the first class (being matched by its associated filter), the link capacity remained at maximum level, even for 10000 classes and associated filters installed.

Figure 3 presents a worst case scenario, when test traffic is associated only to the last class installed (the last associated filter matches all the traffic, and all 10000 filters subsequently test each packet).

Also, we tested the situation when there are only 2 classes installed, and we installed up to 10000 associated filters. Performance was similar to the previous case, so we decided that the number of installed filters impacts the system performance most. This additionaly brings a variable delay, since a packet can be matched by the first installed filter (in this case we have FIFO comparable performance), or it can be treated by the last installed filter (introducing an important performance penalty).

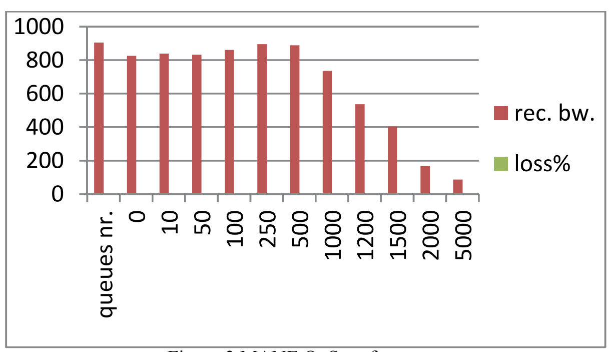

Figure 3 MANE QoS performance

Since the pps rate (traffic size and traffic mix) is the largest single factor in router scalability, we used an *iperf* process to generate maximum packets per second (~122Mbps bandwidth of small packets), and an increasing number of queues where installed, each one with an associated filter installed, inspecting IPsrc, IPdst, Dstport and Srcport fields. Figure 4 shows that our MANE implementation can handle a large number of packets, even with hundreds of particular QoS policies installed.

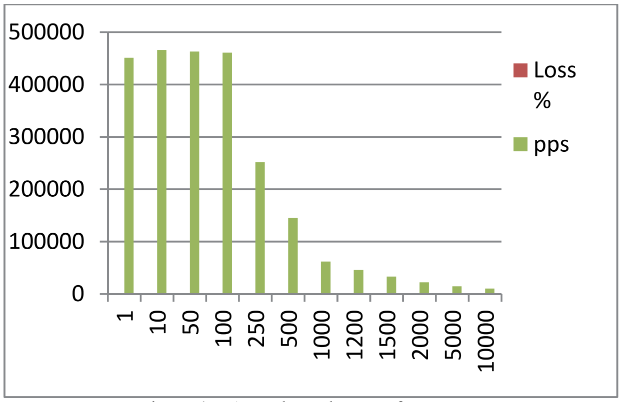

Figure 4 MANE throughput performance

VI. Conclusions And Future Work

We presented the high level architecture of a media aware network, which aims at virtualizing network resources for the purpose of offering higher QoS to media flows. The MANE (Media Aware Network Element) is an edge router that has a central role in implementing the separation between networks, by classifying incoming traffic and distributing it to appropriate MPLS paths inside each domain.

We implemented the MANE using off-the-shelf hardware, using Click modular router to interface the components: classification, routing, adaptation, multicast, MPLS FEC association, encapsulation and decapsulation. Our preliminary implementation shows a modest increase in processing overhead when compared with traditional IP/MPLS processing. It can be seen that a MANE system can treat 1000 flows without introducing an important performance penalty.

In the future, we aim at developing the MANE in two directions: adding deep packet inspection functionality, to assist in classification of traffic not yet associated with a VCAN, and integration with high speed network processing cards, to target operation at line speed. QoS performance improvements would be possible by using an hierarchical filter structure that permits hashing. Both these directions aim at creating a MANE that can be deployed in the field by service providers.

Acknowledgment

This work was supported in part by projects POSDRU/107/1.5/S/76903, POSDRU/89/1.5/S/62557 and in part by the EC in the context of the ALICANTE project (FP7-ICT-248652)

References

[1] Dragoş S. Niculescu, Mihai Stanciu, Marius Vochin, Eugen Borcoci, “Implementation of a Media Aware Network Element for Content Aware Networks”, Fourth International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ 2011.

[2] Eugen Borcoci, Daniel Negru, Christian Timmerer, "A Novel Architecture for Multimedia Distribution Based on Content-Aware Networking,"pp.162-168, 2010 Third International Conference on Communication Theory, Reliability, and Quality of Service, CTRQ 2010.

[3] Networked European Software and Sevices Initiative (NESSI) Strategic Research Agenda, Vol. 3. FP7-2.exec, NESSI Roadmap, Feb. 2008.

[4] European Commission, FP7 ICT Work Programme 2009-2010.

[5] FP7 ICT project, “MediA Ecosystem Deployment Through Ubiquitous Content-Aware Network Environments”, ALICANTE, No248652, http://www.ict-alicante.eu/ (last accessed: Mar. 2010).

[6] Maria G. Martini, et. al., “Content Adaptive Network Aware Joint Optimization of Wireless Video Transmission”, IEEE Communications Magazine, vol. 45, no. 1, Jan. 2007, pp. 84-90.

[7] T. Kourlas, “The Evolution of Networks beyond IP”, IEC Newsletter, vol. 1, Mar. 2007. Available at www.iec.org/newsletter/march07_1/broadband_1.html (last accessed: Mar. 2010).

[8] C. Baladrón, “User-Centric Future Internet and Telecommunication Services”, in: G. Tselentis, et. al. (eds.), Towards the Future Internet, IOS Press, 2009, pp. 217-226.

[9] T. Zahariadis, et. al., “Content Adaptation Issues in the Future Internet”, in: G. Tselentis, et. al. (eds.), Towards the Future Internet, IOS Press, 2009, pp.283-292.

[10] Á. Huszák and S. Imre , “Content-aware Interface Selection Method for Multi-Path Video Streaming in Best-effort Networks”, Proc. of 16th International Conference on Telecommunications, Marrakech, Morocco, Jul. 2009, pp. 196-201.

[11] N. Baker, “Context-Aware Systems and Implications for Future Internet”, in: G. Tselentis et. al. (eds.), Towards the Future Internet, IOS Press, 2009, pp. 335-344.

[12] S. B. Kodeswaran, et. al., “Content and Context Aware Networking Using Semantic Tagging”, Proc. of 22nd International Conference on Data Engineering Workshops (ICDEW'06), Atlanta, Georgia, USA, Apr. 2006, pp. 67–77.

[13] E. Rainge, “The Inevitable Failure of Content-Aware/DPI Network Devices and How to Mitigate the Risk”, Sept. 2008 (adapted from Worldwide Network Test and Measurement 2008.2012 Forecast and 2007 Market Shares, Available at http://www.breakingpointsystems.com/resources/white-papers/idc-white-paper/content-aware-testing.pdf (last accessed: Mar. 2010).

[14] H. Xie, A. Krishnamurthy, A. Silberschatz, and Y. Yang, “P4P: Explicit Communications for Cooperative Control Between P2P and Network Providers”, Available at http://www.dcia.info/documents/P4P_Overview.pdf (last accessed: mar. 2010)

[15] Eddie Kohler, Robert Morris, Benjie Chen, John Jannotti and M. Frans Kaashoek. 2000. The click modular router. ACM Trans. Comput. Syst. 18, 3 (August 2000), 263-297