U.P.B. Sci. Bull., Series C, Vol. 76, Iss. 1, 2014 ISSN 2286-3540

FROM OLD-SCHOOL ROUTING TO CONTENT AWARE ROUTING

Cernat George Cristian[1]

The author proposes an analysis to motivate the necessity for new routing technique using the specific details of various existing entities in an increasingly complex Internet from any point of view. First three sections are a preamble intended to justify the reasons to upgrade existing routing protocols and implementations, this is the preferred option compared to building some new ones. A review is also presented of some mature implementations widely used in present days, emphasizing their technical elements that might support the desired direction for extension.

Keywords: content aware network; content aware routing; BIRD, Quagga, XORP

1. Introduction

All entities involved in the distribution and consumption of content are experiencing a strong need for speed and performance growth on all technical elements responsible with network access, data indexing and storage. Therefore, there will always be enough room to optimize the technical means used to locate content and deliver it to the user in real time using the best possible route from source to destination.

In terms of services and content types the global Internet becomes more complex every day; this causes a positive growth trend for the number of technologies, protocols, types of hardware, data encapsulation methods and other sorts of products. Content and services have complex needs for resources allocation (hardware, software, time) from all the actors involved: Content Provider, Service Provider, Network Provider, Consumer and others [1].

When a user accesses multimedia content having IPTV as core a service, the entire hardware and software infrastructure must decide in real time on some important aspects such as: what content elements must be delivered to user device, what special requirements are to be considered for each content piece (maybe authentication), fetch metadata for each content piece and more.

Besides these entire complex operations performed on the content, the network has to find the best path on the way to end user destination.

[1] PhD Student, Faculty of Electronics and Telecommunications, University POLITEHNICA of Bucharest, Romania, e-mail: cernatcristi@gmail.com

2. CONTENT AWARE ROUTING

A. Classic routing

Routing is a process designed for packet forwarding decisions that originally was able to analyze only a few details strictly related to the transport network, such as bandwidth of links and the cost of each link statically assigned by the network operator. It can be easily noticed that these parameters are not related to any specific details about various content types, or to any entities present on the nowadays Internet: Content Provider, Service Provider.

B. Next Generation Routing

Content Aware Network (CAN) concept captures the attention of researchers for a few years now, one of the main drivers supporting CAN is the concept of Content Aware Routing (CAR), the motivation is simple: since the network will be able to distinguish between different types of providers, types of content or categories of consumers, then the network will also be able to select the best origin and path for a content element (User Object - OU) among several options available, process based on known metadata about content and all entities involved in its use [2]

Basically, the objective is to increase the number of factors underlying routing decisions, there will exist many other criteria besides bandwidth and link cost.

3. TECHNICAL BACKGROUND

The objective of building a routing protocol able to perform previously mentioned functions has two possible solutions: a totally new protocol is built or an existing protocol is enhanced to support any additional criteria that must be taken into account.

There are several reasons to prefer the last option of using existing techniques as a starting point:

- All new routing rules don’t exclude the old ones, only new ones would appear.
- The most widely used routing protocols (RIP, OSPF, ISIS and BGP) have reached a degree of maturity that makes them very efficient.
- There are several very good implementations of these protocols, developed for many years: BIRD (Bird Internet routing daemon), Quagga, XORP (eXtensible Open Router Platform) and more.

For an effective validation of the idea, all cited open-source implementations must be augmented with the necessary intelligence able to process the characteristic metadata of various types of content and services.

A. Experimental implementation

The extensions for quoted implementations should be designed in a manner which allows the determined routes to depend on specific details of a service or content types, additional software modules performing the following general functions are needed [3]:

- Characteristic details (bandwidth, link cost and more) of transport network must be known to new modules;
- Collect consumer characteristic metadata;
- Collect the relevant metadata characteristic to distributed content by CP;
- Assessment of the routes determined by the standard routing protocol (OSPF, ISIS, BGP) utilizing the new metadata learned about entities (CP, SP, UO, User);
- Selection of new optimal route or create a new one.

It is obvious that the new CAN modules must have access to the forwarding information database. For CAR routing we need to alter the network graph constructed by standard routing protocols [4].

New data structures used by routing process to find the optimal path may contain elements that are relevant only for a few entities on the path from source to destination. As an example, we can assume the existence of a special purpose server designated to provide additional security for a video stream passing through a non-trusted domain (encryption or VPN - based on request of Content Provider), this server is owned and operated by a Service Provider that would not be concerned about the diversity of hardware equipment on destinations, possibly consisting of several users each one having various access networks and different screen resolutions [5].

A particular user may want its video traffic received from this Content Provider to be recorded and stored on the network by some Service Provider, therefore routing protocol functions (Content Aware Routing) are highly complex since the traffic must be channeled to a storage server only for a specific user and the stored data must be labeled with appropriate metadata.

Some factors required for routing are uncertain by nature, at the moment when the routing protocol starts to search for optimal route it may not know some important information essential for finding it [6].

B. Content Aware Routing control plane

The solution to the aforementioned uncertainty is the existence of a control plane allowing the routing protocol instances to exchange information in order to enable each other to determine an optimal network path; these instances may be located on different network nodes on distinct autonomous areas. The control information transmitted over network have an opaque character for transited nodes, other similar devices located between two routers engaged in such conversations should not be interested in the topic.

The existence of two major types of routing protocols, inter-domain (BGP) and intra-domain (OSPF, ISIS), suggests the necessity to design at least two distinct layers on the Routing Content Aware Control Plane.

An example of utility for this control plane is a border router using BGP between two autonomous areas; this router is asked if one of his adjacent areas can provide some additional security services for traffic passing through at the request of Content Provider (previous example case). The choice for one or another route of several alternatives may depend on this answer. Another possible scenario is a Content Provider required to augment the video stream of a specific user with some additional data (watermark, movie subtitle or other), for this use case the content aware routers must redirect video stream to pass through a server that performs the content augmenting job.

In conclusion, content aware extensions to an existing routing protocol implementation must be provided with the ability to negotiate parameters between instances located on separate physical devices likely found in different autonomous domains, before an optimal route is selected.

C. Router extensibility

One way a router can achieve extensibility is to be independent of the details about where those routing processes are running and what the composition of the forwarding plane is. The routing processes and network interfaces could be running on one machine or distributed across a cluster of machines that appear as single router. A necessary feature once routing protocols are running in distinct processes and potentially on distinct machines is an inter-process communication mechanism [7].

All mentioned implementations are open-source products, used in real life as part of various networks, which have already reached maturity after several years of development and testing, thereby allowing the integration of extensions for new functionality in a content aware fashion: XORP (eXtensible Open Router Platform), Quagga and BIRD (Bird Internet Routing Daemon) [8].

D. Packet Inspection vs. Metadata Inspection

In-depth analysis of packets is not a realistic method for the purpose of classifying different data streams passing through a router, the computational resources required are significantly high and the level of accuracy is not even close to 100% when the best known statistical algorithms are used. It is therefore appropriate to propose a hybrid classification method based on:

- Minimal inside analysis of data packets, without too much examination performed on application level data (only ports from application level, IP source/destination, standard headers and other).
- Analysis of metadata attached by all entities (CP, SP, HB and other) to data transiting CAN (Content Aware Network), this requires a powerful system for metadata management.

4. EXISTING IMPLEMENTATIONS

A. XORP

XORP implements a wide range of IPv4 and IPv6 routing protocols, offers multicast capability and a centralized system for management of all modules. New protocols, features and functionality can be integrated into the modular architecture of XORP.

The goals of XORP's Inter-Process Communication (XIPC) are: provide all of the IPC communication mechanisms that a router is likely to need (sockets, ioctl's, System V messages, shared memory), provide a consistent and transparent interface irrespective of the underlying transport mechanism used, transparently select the appropriate IPC mechanism when alternatives exist, asynchronous interfaces, wrapper communication with non-XORP processes (like HTTP or SNMP servers), convertible to human readable form so XORP processes can read and write commands from configuration files. In contrast to traditional inter-process communication schemes, the scheme employed in the XORP project can utilize multiple transport protocols and potentially communicate with unmodified components through these protocols, for instance SNMP or HTTP [7].

The XIPC (XORP's Inter-Process Communication) goals are realized through XORP Resource Locators (XRLs) and the XORP IPC (XIPC) library. XRLs are responsible for describing an inter-process calls and their arguments. An XRL may be represented in human readable form that allows for easy manipulation with editing tools and invocation from the command line during development. XORP processes export XRL interfaces to a process known as the Finder and inform it of which IPC schemes are available to invoke each XRL. The Finder is then able to provide a resolution service for XORP processes. When a process needs to dispatch an XRL it first contacts the Finder, obtains details on which IPC mechanisms are available to contact the process, and then instantiates a suitable transport. The XIPC library provides the framework for handling and manipulating XRLs, communicating with the Finder, and common protocol families for conducting IPC [9].

XRL communication is not limited to a single host, and so XORP can in principle run in a distributed fashion. For example, we can have a distributed router, with the forwarding engine running on one machine, and each of the routing protocols that update that forwarding engine running on a separate control processor system [10].

If the router is distributed, in the sense that some Forwarding Engines (FEs) are not in the same chassis as the control software, then the FEA will also handle control communications with the remote FEs.

The RIB process takes routing information from multiple routing protocols, stores these routes, and decides which routes should be propagated on to the forwarding engine.

In [5] we can an experiment proving that event driven route processing model leads to faster convergence than the traditional route scanning approach. 255 routes were introduced from one BGP peer at one second intervals and was recorded the time that the route appeared at another BGP peer. The experiment was performed on XORP, Cisco-4500 (IOS Version 12.1) and Quagga-0.96.5. With XORP the delay never exceeds one second, it illustrates that the multi-process architecture used by XORP delivers similar performance to closely-coupled single-process architecture. Fast convergence for Cisco and Quagga routers was not possible because of scanner-based approach, where all the routes received in the first 30 seconds are processed in one batch.

B. QUAGGA

Quagga also implements OSPFv2, OSPFv3, RIP v1 and v2, RIPng and BGP-4 routing protocols. New protocols, features and functionality can be integrated into the modular and extensible architecture of Quagga. Like XORP, Quagga uses Zebra as kernel routing manager [11].

Strong points of Quagga compared with XORP competing open-source platform: several commercial software and hardware products have implemented it (Vyatta Inc, Amazon VPC), less hardware requirements, is the most widely deployed platform. Also, Quagga software stack appears to be superior to XORP, with higher degree of scalability, performance, and stability under load, particularly with complex routing policies. Also, should be remembered that Quagga has a smaller code base, theoretically making it easier to understand, debug and less prone to bugs (Quagga around 220K lines and XORP around 826K lines).

Disadvantages of Quagga, compared with BIRD, are fewer features in the multicast area where some extensions are developed externally (OSPFv3 MDR, PIM SM).

In [4] we have a performance evaluation of Quagga vs. XORP vs. proprietary hardware router (Cisco 2801). The results obtained on a PC-based router equipped with Quagga routing software and the Cisco 2801 shows that the main component of switching time is the Shortest Path First (SPF) computation time for Dijskstra’s algorithm. Quagga performs better than the Cisco 2801 only when the number of routers R in the emulated topology is smaller than 45. The Cisco 2801 performed better for a larger number of routers. That is due to a sub-optimal implementation of Dijkstra’s algorithm in Quagga.

2.3. BIRD

BIRD is an Internet Routing Daemon designed to support all the routing technology used in the today's Internet or planned to be used in near future and to have a clean extensible architecture allowing new routing protocols to be incorporated easily. BIRD has one or more routing tables which may or may not be synchronized with OS kernel and which may or may not be synchronized with each other [12]. Strong points of BIRD compared with Quagga competing open-source platform [13] [14]: BIRD can handle well higher loads of peers compared to Quagga; BIRD uses less CPU and memory resources on the same loads compared to Quagga; BIRD converges more rapidly and performs well in live environment, not only in lab testing.

In [15] we have a comparison made on Linux platform for BIRD vs. Quagga (0.99.10) regarding full IPv4 BGP table import with about 300k routes. Without OS routing table sync, Quagga required 87 MB of memory and 30 seconds from CPU and BIRD required 30 MB of memory and 7 seconds from CPU. The same performance advantage for BIRD was recorded for the case when OS routing table sync was executed.

3. Conclusion

None of all three presented solutions has all the advantages; therefore it makes sense setting a list of priorities in the assessment of solutions in order to choose the best way to achieve objectives. Most important detail is the support for major routing protocols (BGP, OSPF, ISIS and RIP): all implementations considered shows about the same level of support for routing protocols. It may be considered that certain differences are only aspects related to the compliance with details specified in the protocol RFCs or other recommendations. All three solutions are supported by a development team and a community of users that ensure bug fixes and features improvement.

Next two criteria are performance and scalability evaluated on similar workload: BIRD is a winner in this section, tests show a better score compared with Quagga.

The following assessment detail is the level of extensibility, meaning the ease with which solutions can be extended to support new protocols and functions. XORP seems to lead this section thanks to a fully modularized architecture having clearly defined interfaces between various components.

In terms of integration with external entities for providing information about existing topology at a given time, all three solutions possesses such features, XORP having a slight advantage over the rest.

Ultimately, the presence of advanced features for Inter-Process Communication is an important asset for XORP as it may work in a distributed manner as shown in this presentation; entities running in different nodes already have the necessary infrastructure to exchange data between them.

Concluding, BIRD is recommended to be used as a base platform for routers in a Content Aware Network, in which must be added new features for Inter-Process Communication similar to those in XORP and to ensure that software interfaces are available for other 3rd party entities to query in order to get info about network topology.

Acknowledgments

This work was supported partially by the EC in the context of the ALICANTE project (FP7-ICT-248652) and partially by the projects POSDRU/89/1.5/S/62557 together with POSDRU/88/1.5/S/61178.

R E F E R E N C E S

[1]. *D. Medhi and K. Ramasamy*, “Network Routing Algorithms, Protocols, and Architectures”, Morgan Kaufmann Publishers by Elsevier Inc., 2007.
[2]. *Robert R., A. Gluhak, A. H. Aghvami*, "Multicast in Third-Generation Mobile Networks - Services, Mechanisms and Performance", WILEY 2009.
[3]. *Cernat George Cristian, Vlad Andrei Poenaru and George Suciu*, "Metric model for IPTV video streaming services distribution in content aware networks", 9th International Conference on Communications COMM2012, in press.
[4]. *Vincenzo Eramo, Marco Listanti and Antonio Cianfrani*, “OSPF Performance and Optimization of Open Source Routing Software”, INFOCOM Dpt, University of Rome “La Sapienza”, 2007
[5]. *Mark Handley, Eddie Kohler, Atanu Ghosh, Orion Hodson, Pavlin Radoslavov*, “Designing Extensible IP Router Software”, International Computer Science Institute, University College, London, UCLA
[6]. *Elisa Jasinska, Chris Malayter*, “Using Route Servers”, NANOG 48
[7]. XORP Inc. and individual contributors, XORP Inter Process Communication Library Overview, Version 1.8-CT June 1, 2010, xorp-users@xorp.org
[8]. *Keith Mitchell*, “Open Source Router Projects”, ISC - uknof21
[9]. “XORP Inter-Process Communication Library Overview”, Version 1.8-CT, XORP, Inc. and individual contributors, June 1, 2010
[10]. *Mark Handley*, “XORP Tutorial”, Department of Computer Science, 2005
[11]. *Kunihiro Ishiguro*, et al., “Quagga Routing Suite Documentation”
[12]. *Ondrej Filip, Pavel Machek, Martin Mares, Ondrej Zajicek*, “BIRD Programmer's Documentation”
[13]. *Ondrej Filip*, “BIRD Internet Routing Daemon”, RIPE59
[14]. *Tim Preston*, “BIRD Route Server Daemon Deployment”, UKNOF 15
[15]. *Ondrej Filip*, "BIRD Internet Routing Daemon", NANOG-48, Austin, TX