Application-Aware Routing Policy based on application pattern traffic

Joe Carrión, Daniel Franco and Emilio Luque
Computer Architecture and
Operative Systems Department
Universitat Autónoma de Barcelona,
08193, Bellaterra, Spain
Email: joe.carrion@caos.uab.es, daniel.franco@uab.es, emilio.luque@uab.es

***Abstract*—Search engines are deployed over large datacenters and they support queries of thousands of users about a set of heterogeneous content, a typical configuration of a search engine includes three main components, a Front Service (FS) for user requests, Cache Service which is a subset of the most frequently used data and the full data set called Index Service (IS). Queries generate thousands of messages between nodes. The message delivery process should be performed efficiently. This paper is focused on improving the efficiency for allocating network resources. It is proposed the analysis of communication patterns of the system and the work balance of the network. The traffic pattern defines the behavior of each service and the workload on each router. The proposed mechanism makes decisions based on the historic behavior of the IS, FS, CS and the application pattern. The experimental results are evaluated using simulations techniques. Workloads from a real search engine are injected to the simulator and finally the results are compared with conventional routing mechanisms.**

Keywords*—Routing algorithm, interconnection networks, application-aware network.

I. Introduction

Search engines can be oriented for users all over the world and the content provided can be heterogeneous (general purpose search engine) we call them as Horizontal Search Engine (HSE). HSE are supported for a set of Vertical Search Engine (VSE). The VSE is a component of a HSE and they can be focused on specific content (commercial, scientific, cultural, etc.) for users from a delimited geographic area. Each VSE processes the requests of limited subset of users. Using VSE is an approach to balance the workload of the HSE.

On environments of VSE, the datacenter and network design have the capacity to scale from low to high periods of workload. The frequency of queries submitted is unpredictable and it changes from hundreds to thousands very quickly. This behavior throws an unexpected traffic to the network resources, hosts and routers. Therefore an efficient network resources allocation policy is desired.

Network protocols, devices and services have been designed with a huge set of configurable features, this approach allows extend the range of supported applications on expenses of performance and cost. However for datacenters focused on a reduced set of applications a specially designed datacenter and network is required.

Regarding to datacenter in [1], some best practices related to power efficiency are proposed. In [2] some guidelines to datacenter designs from the industry are proposed. Secondly, when the datacenter hosts a delimited set of applications, the network design should include an exhaustive traffic analysis to support the specific application hosted. The output should be an application-aware network (AAN). AAN is a mechanism "for boosting utilization of network resources based on customer demand" [3]. Since this point of view network design should be based on the hosted applications. The goal is a network based on the profile application. There are different parameters to draw the profile of an application. From literature we can mention [11], which includes number of terminals, latency, message size, traffic pattern and others. This information can be captured on runtime using monitoring standard protocols like sFlow and Netflow. They use a sampling mechanism [3] applied to the traffic network. The application monitoring process returns the delay of the messages of clients, servers and network devices, response time, number of new connections, bytes and packets submitted, packet lost and latency. Then the monitoring application generates very useful information for allocating network resources on demand.

In this paper we introduce a routing policy based on the application pattern analysis for a specific application using trace files from a real VSE. This policy computes the workload of the applications. A runtime monitoring process allows us balance the traffic and allocate network resources fairly. We evaluate the results with two standard network metrics: latency and throughput.

II. Related work

Application pattern analysis involves collecting traffic and data analysis. [4] analyzes the Intra-Rack and Extra-Rack traffic, link utilization and hot-spot behavior to create a profile of an application. [7] proposes a static mechanism based on a traffic analysis to identify the best routes between couples of nodes, after, they combine the best routes to create a set of new optimal routes. Furthermore network resources management techniques have been proposed, for instance [5] describes Generic-Adaptive-Resource-Control (GARC) as a control mechanism of connectivity to reduce the overall performance, GARC is a mediator between applications and network. Most specific techniques like routing policies allow load balance, [6] introduces Application-Specific Routing Algorithm (APSRA) to model the application using graphs, and

the output is a static routing table. The application pattern has impact on the network status [11], so adaptative routing algorithms aim to make decisions based on network status. [8] introduces Distributed Routing Balancing (DRB) which uses an algorithm based on maximize the use of resources by creating new paths between nodes and minimizing the path-length, the output is a fairly message distribution. In [9] PR-DRB (Predictive DRB) extends to real applications by monitoring the best paths and storing them to make routing decisions based on alternative paths.

III. Application Pattern Analysis

A. Background

We conduct our research over a VSE. The main software components of a VSE are: Front Service (FS), Cache Service (CS) and Index Service (IS) deployed on a Fat-tree topology [13].

The services of the VSE are deployed on a large cluster of computers. The nodes are arranged on arrays of P x D, where P define level of data partitioning and D the level of replication of the data, a full description of the architecture is published in [13], we show a basic overview on figure 1 and we remark the following terminology useful for this paper: FS as Front Service, IS as Index Service, ISR as Index Service Replica, CS as Cache Service and CSR as Cache Service Replica.

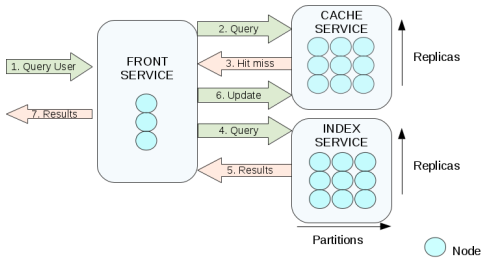

Fig. 1. Overview of traffic flow between the main components of the VSE.

B. Communication pattern

VSE communication pattern is defined for the messages between nodes. The load of the system depends on two elements, the volume of user requests and the size of the content stored in the system. The volume of users submitting queries in a period of time is unpredictable. This rate of input triggers an unstable communication pattern if we compare it with traditional synthetic traffic patterns used to evaluate network performance.

Let N a Fattree topology network with 64 nodes. Axis Y represents the range of sources and destinations and axis X is used for time. We can see the trend of source and destination and the distribution of each couple. The traffic pattern is defined by a couple of Sender and Receiver (S-R).

On figures 2 and 3, we depict a set of samples of synthetic traffic from literature [11]. Synthetic traffic patterns used are Uniform (UTP) and Shuffle (STP). For synthetic traffic we are using 64 nodes to represent plainly instead of 128 nodes of the real VSE network.

Using UTP the Sender in predictable through the time, in contrast Receiver is randomly selected. Figure 2 shows that traffic is distributed in a large set of nodes as source (from node 0 to 64) and the destination goes in the same interval. The result is a distributed traffic over all network. Although the pairs S-R are unpredictable, the traffic goes through a big range of network resources. Then we conclude traffic is distributed between a big set of resources.

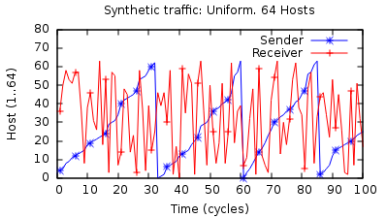

Fig. 2. Uniform Traffic Pattern.

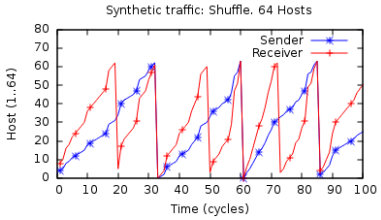

Fig. 3. Shuffle Traffic Pattern.

With STP the S-R pairs are predictable because R is computed from S. The traffic pattern goes from lower nodes to the highest. The result is a traffic distributed on specific part of the network. The used region of the network moves uniformly.

We represent the traffic of a VSE using a trace file from a VSE on figure 4. We call this traffic VSETP (VSE traffic pattern). Although the couples S-R are unpredictable through the time, they move over a specific set of nodes. VSETP is distributed around a reduced set of nodes through the time. For instance on figure 5, axis X shows the time, interval is 25 to 35, senders are less than 10 and receivers are from 30 to 50. The same pattern appears on range 55 to 65 and 115 to 125. The range of nodes define locality and time define frequency (or repetition). Figure 6 shows the repetition and locality for receiver. Periods of time are defined from 92 to 102, 112 to 122 and 145 to 165. Figure 7 shows the trend for using a configuration of 256 nodes. The locality is defined for senders from 0 to 50 and receivers from 150 to 200 when time goes from 20 to 40. This trend is repeated on period 90 to 110.

The traffic of those periods of time can be managed by a specific policy taking into account the repeatability and locality.

C. Traffic Flow

User queries are accepted by the FS, it submits the queries to CS. If the query has been performed previously the query has been cached by the CS (based on a cache policy), thereby CS checks these conditions and returns the output of the query (cache hit) otherwise when the output is not cached, the output is a cache failure. FS redirects the unsuccessfully queries to

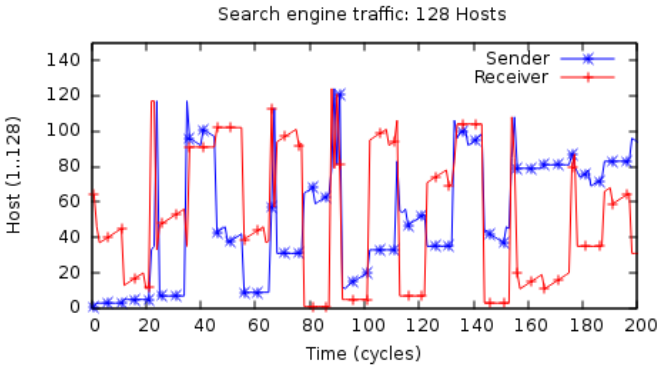

Fig. 4. VSE traffic pattern. Trend of each S-R pair with 128 nodes.

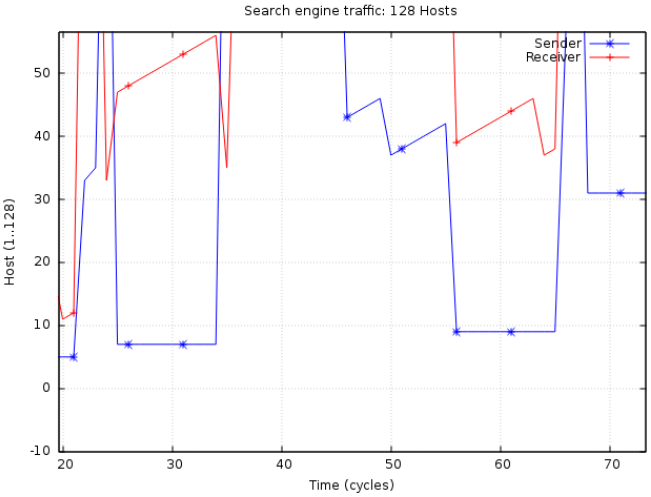

Fig. 5. Trend of repetition and locality for Sender with 128 nodes.

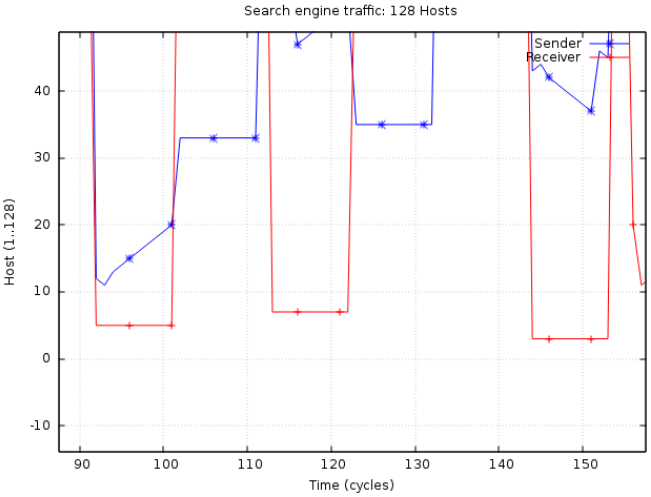

Fig. 6. Trend of repetition and locality for Receiver pair with 128 nodes.

IS, which will create a top-k results to send back to FS. The detailed business rules about a VSE are out of the scope of this paper, a full description is published in [13].

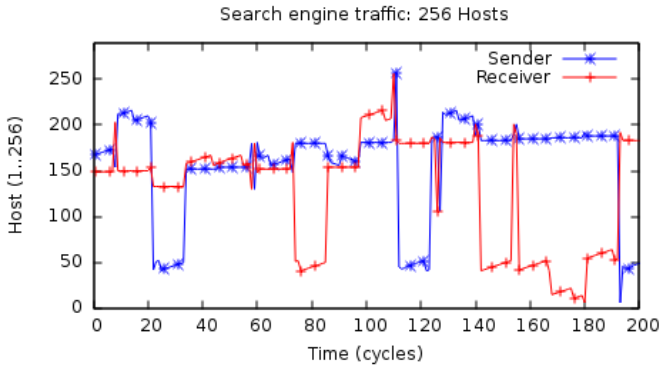

Fig. 7. VSE traffic pattern. Trend of each S-R pair with 256 nodes.

D. Workload analysis

The workload analysis needs a real trace file. We take two networks with 128 and 256 nodes. Firstly we evaluate the workload for the whole system. To depict the load of the network we take the traffic generated by the FS as source (sender) and the destination (receiver) the remaining components (IS, ISR, CS and CSR). The trace file is generated after processing 100000 queries.

On figure 8 axis Y shows the number of messages. We show the workload of each service thought the time in axis X. We can see the load is predominated by ISR, a closer view shows that IS traffic is proportional to traffic ISR; also CS and CSR keep almost constant. Figure 9 shows the workload for a network with 256 node and we can see that ISR traffic behavior prevails.

Fig. 8. Workload with 128 hosts.

There are periods of time with more messages in the network and there are periods with a low number of messages; for example in figure 10, we show the 128-nodes network to the period of time from 50 to 60. The number of messages by unit time is more than 6000 messages. So the injection rate is very high compared to the period of time from 65 to 70.

The variance of the number of messages causes a variance of the injection rate; the application traffic pattern requires proportional allocation resources accordingly to the traffic of each service. There are more messages from ISR compared to other services.

We use this behavior to allocate network resources using a routing policy to balance the workload. The first approach is focused on allocate routers buffers according to the service.

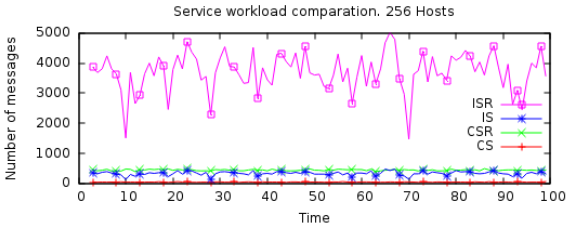

Fig. 9. Workload with 256 hosts

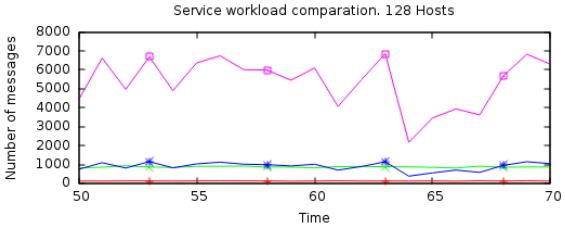

Fig. 10. Changes of the workload by kind of service of the VSE traffic.

E. Buffer Occupancy Analysis

We analyze the workload over the network; this paper is focused on the buffer of input channels (B). Taking into account each message is divided in packets to go from a source to a destination. When a packet (m) arrives to a router (R), it is allocated into an input buffer to wait for an output channel.

Router buffers have a size (BS) and the size should be computed accordingly to workload application (WL). Buffers contain a set of packets, the number of packets define the buffer occupancy (BO). BO goes from 0 to BS. We define 3 thresholds for BO. If BO goes from 0 to 25% of BS we call BO low (BOL), Medium BO (BOM) when BO goes greater than 25% to 50% and High BO (BOH) for BO higher than 50%. The higher BO the more congested is the channel.

An overview of BO for a typical network configuration using UTP is showed in Figure 11. The network configuration used is a Fat-tree topology, with k=10 and n=3, then we have routers arranged in 3 levels (Level 0, 1 and 2, Level 0 is the root of the tree). There are 300 routers, and we have more than 1000 B. Axis X shows the buffer routers. There are 3 groups of bars, one for each level of routers.

Figure 12 zooms only from the 900 to 1100 BO. The size of the bars represents the number of events that BO was higher than LBO. Therefore for UTP the workload is distributed over all routers. The most congested B belongs to routers of Level 2, and the less congested B belongs to Level 0. This distributed traffic is desired for real traffic patterns. However this tendency does not appear on the pattern generated for a VSE.

We apply the same analysis for VSETP and we can see an unbalanced behavior. Figure 13 draws BO for the same threshold LBO. BO is higher than 25% in most of the routers, but there are some buffers without occupancy. Also there are events when BO is higher than 50%. Figure 14 show the behavior for HBO.

Figure 13, shows that workload distribution is not balanced, on Level 2, the occupancy in concentrated in a reduced set of

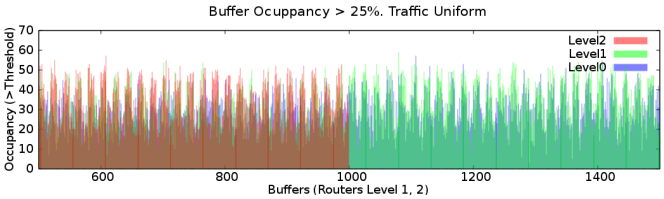

Fig. 11. Full buffer occupancy using Uniform synthetic traffic.

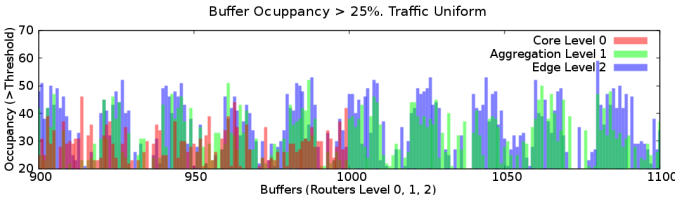

Fig. 12. Buffer occupancy using Uniform synthetic traffic.

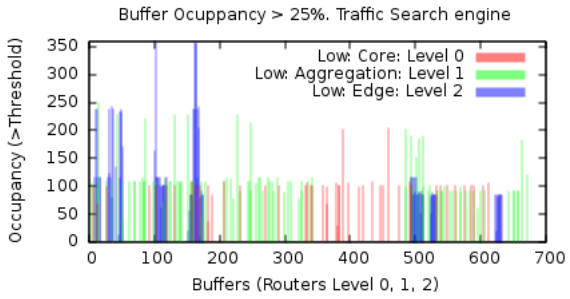

Fig. 13. Buffer occupancy using traffic in the VS with Threshold 25%.

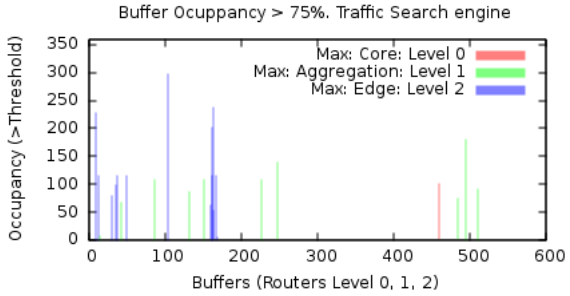

Fig. 14. Buffer occupancy using traffic in the VS with Threshold 75%.

routers. Some routers do not get the threshold for LBO. Level 1 also has the same unbalanced behavior although, it is less than Level 1 and finally on Level 0 the unfair workload prevails.

A routing policy should allocate buffers accordingly to the load of an application. We know the workload application for each service, then we can allocate routers based on each service. In order to detect the kind of service it is necessary identify the sender and receiver; the routing policy should redirect the messages based on workload and BO.

F. Host-Destination Analysis

This analysis is based on then traffic flow described on Section III-C. Each node only keeps contact with a delimited set of services, for instance a CS node submits messages to a

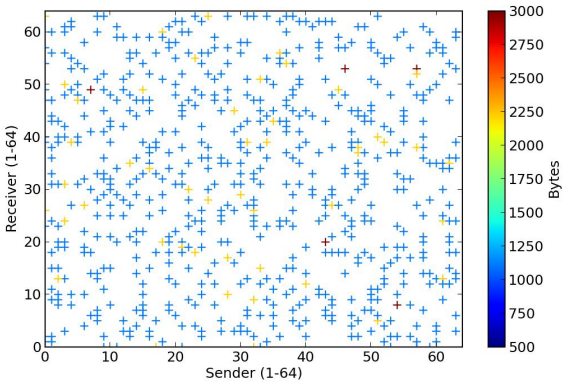

Fig. 15. Pairs SR using Uniform Synthetic traffic.

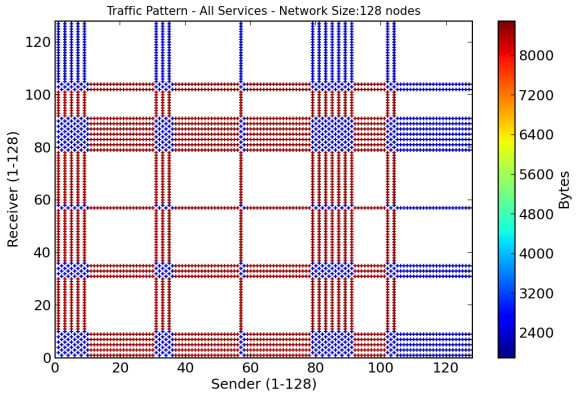

Fig. 16. SR using Flow-traffic conditions of the VSE.

FS node, and a IS node submits messages only to a FS node. With algorithms to generate synthetic traffic the behavior can be predictable, for random generation the set of couples is unpredictable, see figure 15. Each pair S-R appears randomly distributed on space. However using the Flow-traffic conditions of the VSE the set of couples is deterministic, figure 16 shows that each pair S-R is created with a specific set of nodes. For instance there is not traffic from node source 20 to node destination 20 and 40 (source) to 40 (destination). We have a well-defined empty area.

The tendency reduces the set of couples and it creates an unbalanced traffic. Areas with traffic are much defined and areas without traffic can result in network resources unused. On one hand if there is resources unused the network could be reduced, on the other hand there is buffers overloaded.

IV. Application-aware Routing Policy

Based on analysis presented in previous section, we present a routing policy called Application-Aware Routing (AAR) which allocates network resources based on application profile.

AAR can be outlined as follows. Three main components are introduced into router architecture. First one is the Buffer occupancy monitor (BOM), the second one is the Deep Packet Inspection (DPI) mechanism and finally a Decision Maker (DM). Figure 17 shows an overview of the components.

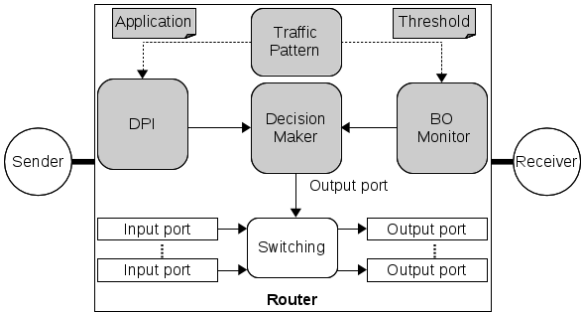

Fig. 17. Overview of the AAR.

BOM keeps historical information about buffer occupancy. DPI identifies if a packet belongs to a specific service. The DM based on information of BOM and DPI redirects the traffic by allocating output channels. Packets belonging to IS or ISR are redirected to output ports with less BO. This approach allow us allocate resources proportionally to service demand. The workload is distributed toward less used network area.

A. Buffer occupancy monitor

BO tracks the buffer occupancy on runtime. This tracking process is based on three thresholds. High, medium and low occupancy (BOH, BOM and BOL respectively). When a packet arrives it is allocated a buffer. There is a BO counter for each buffer, which is updated when it reaches a threshold. These three levels allow us tune the policy accordingly to the profile application. For instance in Section III-D we conclude that VSETP is predominated by the ISR. We configure the routing policy with BOH to apply the policy to redistribute the traffic of the service with higher workload. The next step involves a detection service action.

B. Deep Packet Inspection and Decision Maker

This component takes as input two parameters, a mapping of hosts-services and application profile based on the workload. When a packet arrives to the router the policy merges the mapping table and the workload. The output is a candidate packet to be redirected. This packet is passed to the DM. The remaining packets are redirected applying the default routing policy of the system.

Taking as input the BOM and the candidate packet, the packet is redirected based on the BOM. BOM sends the packet to the output port with lower occupancy.

V. Experimentation

We evaluate AAR using a modified version of Booksim simulator published in [10]. We present the results comparing AAR with conventional Nearest Common Ancestor (NCA) introduced in [11]. The network is configured with the same characteristics of the real VSE. So we use 128 and 256 nodes arranged in a Fat-tree topology of tree levels. The result are analysed using two standard metrics for networks: Average Network latency (ANL) and Accepted Packet Rate (APR).

Firstly we focus on the overall AAR and NCA comparison with VSE traffic pattern. It takes the whole period of time of simulation.

On figure 18 we depict the ANL. Axis X is the time, (in K-cycles of simulation). Axis Y is the number cycles need to deliver a packet. The first overview shows the curve of AAR under the curve of NCA almost in the full period of time. This tendency exemplifies a decrease of the network latency using AAR.

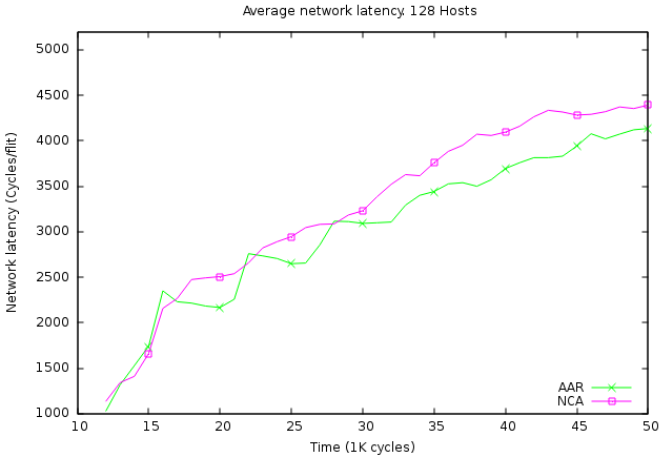

Fig. 18. AAR and NCA using 128 nodes and traffic pattern of VSE

In table I we show that network latency (NL) is 3339 K-cycles using NCA, we take this value as baseline and we compare the result of AAR. AAR gets 3119 Kcycles. Then AAR reduces the network latency in 7.26%.

TABLE I. COMPARISON BETWEEN AAR AND NCA. NODES:128. TRAFFIC PATTERN: VSE

Metric	AAR	NCA
Average network latency	3119	3339
Percentage of decrease	92.73%	100%

Regarding to the analysis of APR, on figure 19 we can see the curves of APR using two routing algorithms. Axis X is the time and axis Y is the number of messages delivered on each unit time. The AAR curve is over the NCA. It shows that AAR delivers more packets on each unit time against NCA.

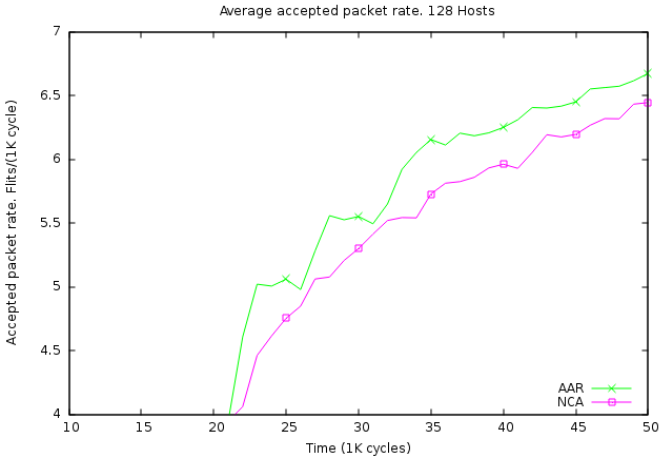

Fig. 19. AAR and NCA with 128 nodes for traffic pattern of VSE

Table II shows APR, using NCA is 4.91 packets accepted by each K-cycle. Using AAR the APR grows up to 5.12. This increase represents the 4.4%.

TABLE II. SUMMARY OF COMPARISON BETWEEN AAR AND NCA. NODES: 128. TRAFFIC PATTERN: VSE

Metric	AAR	NCA
Average accepted packet rate	5.12	4.91
Percentage of increase	104.4%	100%

Now we analyze the behavior through the time. If we divide vertically the curves of ANL and APR in two areas, the part of the right shows that AAR and NCA are overlapped when the time is lower than 25. Also, the ANL is less than 3000K cycles. However, this tendency change later if we see the left part, when the time is higher than 25. The difference between two curves increases and then we compare the routing algorithms after a period of time.

At the beginning there is not information about the buffer occupancy. Then AAR performance is the same as NCA. When AAR has enough information about buffer occupancy, its performance improves against NCA. NCA allocates network resources in a deterministic way. On the other hand AAR takes advantage of the historic information.

On table III we show the results when AAR have information about BO. We compare the performance when the latency is higher than 3000 Kcycles. After this period of time AAR has collected information about buffer occupancy then the policy is applied for more packets compared to the beginning of network operation.

The final results show that AAR improves latency on 7.93% and the accepted packet rate increases in 4.78%.

TABLE III. SUMMARY OF COMPARISON BETWEEN AAR AND NCA. NODES: 128. TRAFFIC PATTERN: VSE. AFTER WARM UP PHASE

Metric	AAR	NCA
Average network latency	3662	3978
Percentage of decrease	92.06%	100%

TABLE IV. SUMMARY OF COMPARISON BETWEEN AAR AND NCA. NODES: 128. TRAFFIC PATTERN: VSE. AFTER WARM UP PHASE

Metric	AAR	NCA
Average accepted packet rate	6.21	5.92
Percentage of increase	104.78%	100%

Next set of experiments were carry out using a configuration for 256 nodes. The workload is a VSETP. Figure 20 shows the latency. The AAR curves goes over the NCA curve.

As we can see in table V network latency (NL) is 5213 Kcycles using NCA and we compare the result of AAR. AAR gets 4996 Kcycles. Then AAR reduces the network latency in 4.16%.

TABLE V. SUMMARY OF COMPARISON BETWEEN AAR AND NCA. NODES: 256. TRAFFIC PATTERN: VSE.

Metric	AAR	NCA
Average network latency	4996	5213
Percentage of decrease	95.83%	100%

The APR is showed on figure 21. The period of time less than 10 show the warm-up period when there is not traffic.

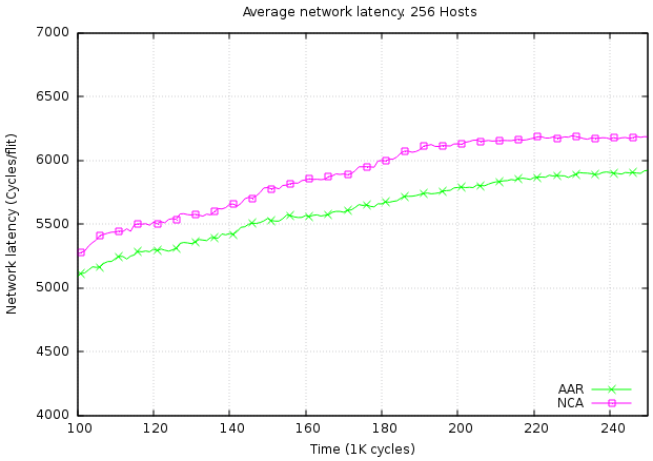

Fig. 20. AAR and NCA with 128 nodes for traffic pattern of VSE

From period 10 to 50, NCA have AAR have almost the same performance. AAR needs information about buffer occupancy to distribute the packets, after collecting information the performance provided by AAR increases.

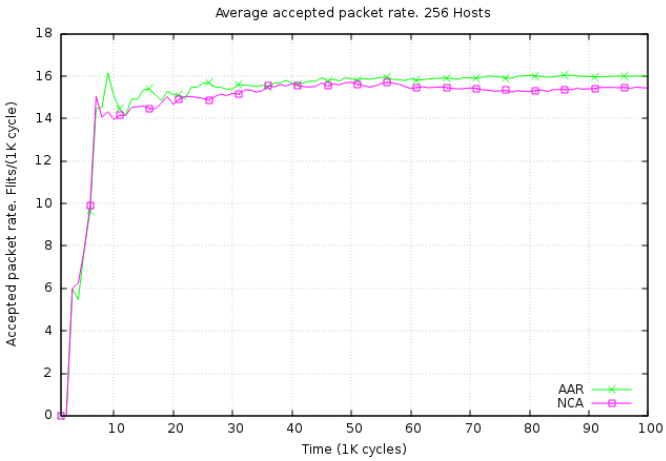

Fig. 21. AAR and NCA with 256 nodes for traffic pattern of VSE

Table V shows that APR is 14.96 packets using NCA, and the result using AAR is 15.54 packets. Then AAR improves the throughput in 3.92%.

TABLE VI. SUMMARY OF AAR AND NCA COMPARISON. NODES: 256. TRAFFIC PATTERN: VSE

Metric	AAR	NCA
Average accepted packet rate	15.54	14.96
Percentage of increase	103.92%	100%

The set of experiments demonstrates that AAR delivers more packets while it reduces the network latency. The performance prevails for networks with 128 and 256 nodes.

VI. CONCLUSION

We have proposed the Application-Aware Routing policy, AAR. This policy is based on the analysis of an application pattern of a vertical search engine. The algorithm keeps historical information about buffer occupancy and the application profile. The network resources are allocated on application demand. The performance of AAR is analyzed with two standard network metrics: latency and throughput. Experiments show AAR improves the network performance by reducing the latency and delivering more packets in the same period of time. Future work is oriented to extend the analysis to other network components and test our AAR with different topologies.

ACKNOWLEDGMENT

This research has been supported by : MINECO (MICINN) Spain under contract TIN2011-24384, SENESCYT[1] Ecuador government under contract 2013-AR7L335.

Authors would like to thank to Veronica Gil-Costa, Mauricio Marin and Yahoo! Research Latin America.

REFERENCES

[1] Greenberg, et al,*Best Practices for Data Centers: Lessons Learned from Benchmarking 22 Data Centers*, Summer Study on Energy Efficiency in Buildings, ACEEE 2006, pp. 7687.

[2] Cisco Systems, Inc. 2007, Cisco Data Center Infrastructure 2.5 Design Guide. http://www.cisco.com

[3] MRV,*Application-Aware Networking at A Glance*, White Paper 2013, http://www.mrv.com.

[4] Theophilus Benson, Aditya Akella, and David A. Maltz. 2010. *Network traffic characteristics of data centers in the wild*. In Proceedings of the 10th ACM SIGCOMM conference on Internet measurement (IMC '10). ACM, New York, NY, USA, 267-280. DOI=10.1145/1879141.1879175 http://doi.acm.org/10.1145/1879141.1879175

[5] J. Mueller, T. Magedanz. *Towards a Generic Application Aware Network Resource Control Function for Next-Generation-Networks and Beyond*. In *IEEE* Proceedings of the International Symposium on Communications and Information Technologies (ISCIT 2012), pp. 777882.

[6] Palesi, M., Holsmark, R., Kumar, S. Catania, V. *Application Specific Routing Algorithms for Networks on Chip*. Parallel and Distributed Systems, IEEE Transactions on 20, 316-330 (2009).

[7] N. Michael, M. Nikolov, A. Tang, G. E. Suh, C. Batten, Proceedings of the Fifth ACM/IEEE International Symposium , 9-16 (2011).

[8] D. Franco, I. Garcés, and E. Luque. 1999. *A new method to make communication latency uniform: distributed routing balancing*, ICS '99, ACM, New York, NY, USA, 210-219, http://doi.acm.org/10.1145/305138.305195

[9] Carlos Nunez Castillo, Diego Lugones, Daniel Franco, Emilio Luque, Martin Collier: 2013. *Predictive and Distributed Routing Balancing, an Application-Aware Approach*, ICCS 2013, 179-188.

[10] Nan Jiang Becker, D.U. ; Michelogiannakis, G. ; Balfour, J. ; Towles, B. ; Shaw, D.E. ; Kim, J. ; Dally, W.J.. 2013. *A detailed and flexible cycle-accurate Network-on-Chip simulator*. ISPASS 2013: 86-96.

[11] William Dally, Bryan Twles. 2003. *Principles and Practices of Interconnection Networks*. Morgan Kaufmann, ELSEVIER, San Francisco, p.550.

[12] George Michelogiannakis, Daniel Becker, Brian Towles and William J. Dally. N Jiang. 2013. *BookSim 2.0 User's Guide*. https://nocs.stanford.edu/cgi-bin/trac.cgi/wiki/Resources/BookSim.

[13] Jair Lobos, Veronica Gil-Costa, and Mauricio Marin, Alonso Inostrosa-Psijas. 2012. *Capacity Planning for Vertical Search Engines: An Approach Based on Coloured Petri Nets. Yahoo! Research Latin America.*

[14] Alexander Loukissas, Amin Vahdat Mohammad Al-Fares. 2008. *A Scalable, Commodity Data Center Network Architecture*. SIGCOMM'08, August 17-22, p. 12

[1]SENESCYT: Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación, http://www.senescyt.gob.ec/