УДК 621.391

O. B.TKACHOVA, Ph.D., ABDULGHAFOOR RAED YAHYA,
HASSAN MOHAMED MUHI-ALDEEN

PERFORMANCE ANALYSIS OF LOAD BALANCING MECHANISMS IN SOFTWARE-DEFINED NETWORKING

Introduction

The widespread distribution of different services on demand and business applications requires a designing and implementation the highly-productivity networking solutions. Traditional multiservice network solutions often not allow to provide services with guaranty QoS characteristics. First of all, such situation connected with a high overloading of network environment – the data forwarding equipment today provide a lot of addition management and monitoring functions that influent to the efficient of service provision greatly. The implementation of Software-Defined Networking (SDN) paradigm and different virtualization technology, i.e. cloud computing is became very popular today. The main principles of SDN paradigm and cloud computing are provide the availability of different type of services which are allocated at different parts of the distributed multiservice network to users who wants to access those resources from their work place in the form as a service through an optimized and reliable service provider maintaining convenience and ubiquity [1]. The services implementation through cloud computing has many advantages: increasing network scalability, performance, reliability, fault tolerant and decreasing cost of implementation and utilization. The multiservice networks based on SDN solutions has a lot of benefits. The productivity, control of transport quality, transparent services provision, network function virtualization (no overhead of encapsulation), and flexible traffic management are the most significant benefits of SDN implementation.

However, the complication of SDN paradigm is not finishing process. The traffic engineering mechanisms and service provision scenarios are not completed. The challenges in services provision in SDN-based networks connected with low interoperability between hosts, lake of resource control and nodes migration control. The prediction of load intensity and effective load distribution between different computation nodes is still actual question [2]. In this way the finding load balancing solution for clouds is not easy process.

The analysis of main policies and characteristics like round trip time, throughput that archived by using of different load balancing algorithms and resources utilization give ability to choose and implement better traffic engineering solutions for SDN-based networks.

Overview of load balancing techniques

In general, load balancing techniques include a scope of methods and engineering solutions of optimal load distribution across multiple resources through appropriate network paths. The implementation of appropriate load balancing algorithms according to the current network conditions allows achieving optimal resources utilization, throughput maximization and minimization round trip time minimization [3].

The general task of load balancing mechanisms can be broken into two sub-tasks:

1. Transfer the incoming traffic with a maximal possible intensity.
2. Smooth the load distribution in the network.

In general, the task of effective distribution network bandwidth can be formalized as follows:

$$Q(\Lambda^{IN}, Th_{in\alpha}, Th_{out\alpha}) \rightarrow \max_{CP_\alpha}, \qquad (1)$$

where $Q()$ – the objective function or optimization function, Λ^{IN} – total flow of incoming traffic, $Th_{in\alpha}$ and $Th_{out\alpha}$ – the throughput of network channels, CP_{α} – policy control network load (select network channel).

In this case, the optimization task can be reduced to the following form:

$$\Lambda^{IN} \to \max_{CP_\alpha} | Th_{in\alpha}, Th_{out\alpha} . \qquad (2)$$

In the proposed method of optimal allocation of bandwidth is achieved by distributing the traffic flows on all communication channels permitted in proportion to the available bandwidth of the channel at the current time or the time of receipt of the application time.

The load balancing techniques can be divided into two categories - static [4] load balancing techniques and dynamic [5] load balancing techniques. The division depend of current network policies and management mechanisms.

Static load balancing algorithms are based on the information about the average behavior of system; transfer decisions are independent of the actual current system state. Static load balancing procedures are used in the presence of prior knowledge about the services and applications of statistical information about the network environment. The goal of static load balancing method is to reduce the execution time and minimize the communication delays. Round Robin [6], Randomized [7], Central Manager [8] and Min Min [9] algorithms are the static load balancing algorithms.

In Round Robin algorithm the execution processes are divided between all processors. Each process is assigned to the processor in a round robin order. Distributions of data and incoming request between the computation nodes (web-, mail-, ftp- servers) are equal but the time of execution for different computing node are not same. It is depend of type of processor.

Randomized algorithm is a process that can be handled by a particular node with different probability. The process allocation order is maintained for each processor independent of allocation from remote processor. This algorithm works well in case of processes are of equal loaded. Randomized algorithm does not maintain deterministic approach.

Central Manager algorithm works on the principal of dynamic distribution. Each new request that arrived to the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available.

The Min-Min algorithm firstly finds the minimum execution time of all tasks. Then it chooses the task with the least execution time among all the tasks. The algorithm proceeds by assigning the task to the resource that produces the minimum completion time. The same procedure is repeated by Min-Min until all tasks are scheduled [6].

In dynamic load balancing algorithms work load is distributed among the processors at runtime. The master assigns new processes to the slaves based on the new information collected [7].

Token Routing [10], Central Queuing [11], Least Connection algorithms are the main types of dynamic load balancing algorithms.

Token Routing algorithm minimizes the overload in software-defined networking by use special tokens (agents). Agents gather statistics and distribute traffic according this statistic. The algorithm provides the fast and efficient routing decision.

Central Queuing algorithm works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue.

Least connection is a method of dynamic scheduling of incoming data. The number of connections for each computation node are counted that, the load distribution is estimated the according to this amount of connection. The SDN controller (load balancing module) records the connection number of each server. This algorithm is suitable in case when the amount of nodes changes in some thresholds.

The experimental evaluation of static and dynamic load balancing techniques

In the work such parameters as round trip time and throughput are suggested to measuring the effectivity of different load balancing algorithms of SDN-based networks. The round trip time in such situation is the average amount of time between requests message from end users received response from server. The round trip time greatly impact on such quality of service parameters as delay and jitter. Throughput is the amount of data transferred in one direction over a link divided by the time taken to transfer it, usually expressed in bits or bytes per second.

For evaluation the efficiency of proposed algorithm of load balancing for SDN-based networks the network simulation tool called mininet was used. The software POX SDN controller and two OpenFlow Switches was simulated. The fragment of the experimental network is shown in Fig. 1.

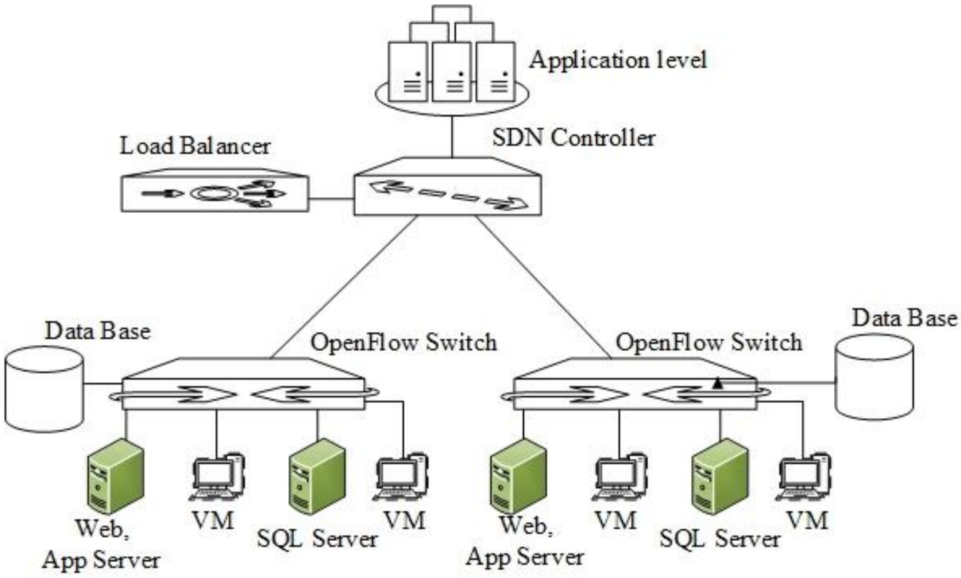

Fig.1. Fragment of the experimental network (mininet interpretation)

Response time per size of message and round trip time per number of users are evaluated in the work. The fixed number of request-response pairs (12 pairs request-response) for messages with different size take into account. The message with size 16kB, 32kB, 64kB, 128kB, 256kB, 512kB, 1024kB and 2048kB generate for experiment. The different amounts of users take into account for evaluation response time per users (from 10 to 700 end nodes).

To simplify the evaluating process assumed that each user generated equal amount of data. Comparing round trip time for load balancing algorithms depending on the message size and the number of users is shown in Fig. 2 and Fig. 3.

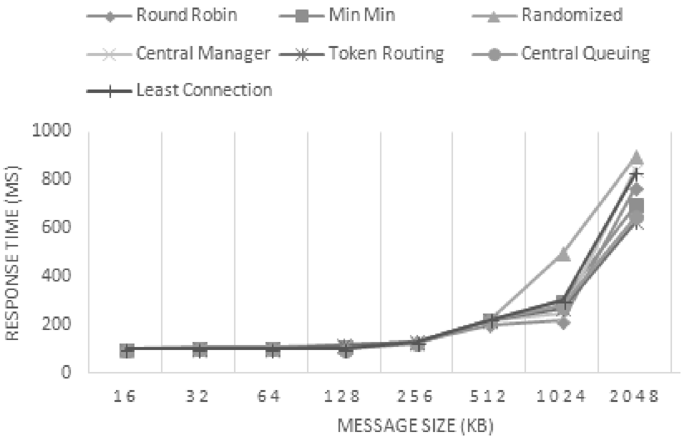

Fig. 2. Round trip time depending on the message size

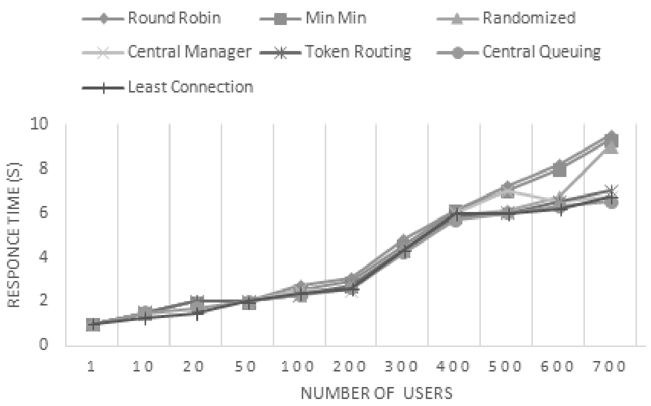

Fig. 3. Round trip time depending on the number of users

The obtained results show that Randomized algorithm gives the highest round trip time depending on the message size. This connected with type of data proceeds. In randomizes algorithm all calculations and choose processor in current time and any prediction not created. Min Min algorithm introduces results near to Randomized algorithm. This is associated with the delay occurrence for finding the fastest processor. Central Queuing, Least Connection and Token Routing algorithms give the best result, especially when the message size is more than 1024 kB. This algorithms predicate on query that can be made frequently.

The round trip time for all analysed algorithms until 200 active users is small. Response time for Min Min algorithm and Round Robin algorithm grows rapidly in the case when amount of end users more than 500. Central Manager and Central Queuing algorithms give the smallest round trip time.

The throughput as a function of message size and as a function of number of users, which generate proximately the same amount of data, shown on Fig. 4 on Fig. 5 respectively.

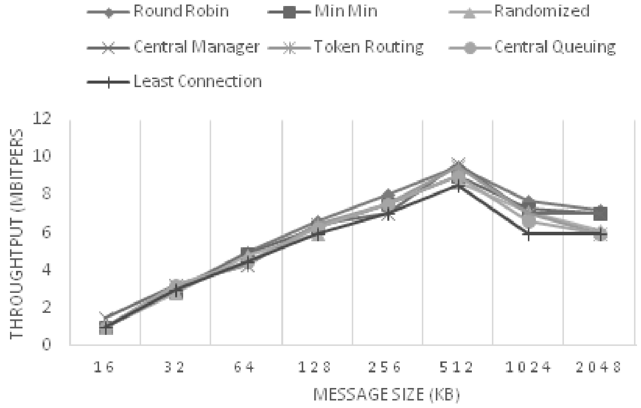

Fig. 4. Throughput depending on the message size

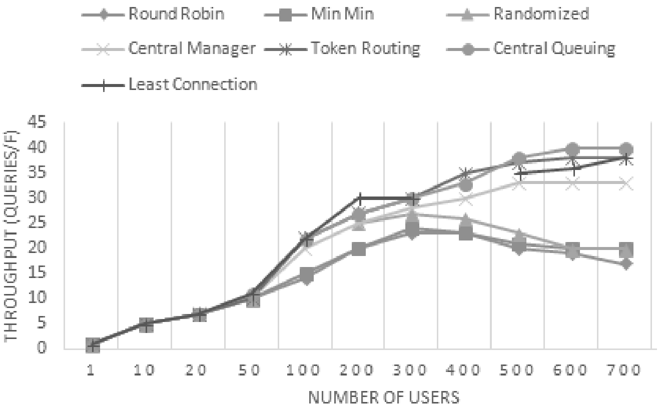

Fig. 5. Throughput depending on the number of users

Throughput comparing (Fig.4 and Fig.5) shows that Round Robin, Randomized and Min Min algorithms give the smallest throughput. Round Robin algorithm depends on Round Robin order, so there is no strategy to know the best loaded node. Min Min algorithms try to find a faster processor of computing nodes in this way overload generates. Randomized algorithm doesn’t have any strategy to find best processor too. Central Manager, Least Connection and Tokin Routing algorithms introduce the highest throughput in the case when the amount of end users grows.

Conclusion

The main purposes of load balancing are resource provisioning and scheduling tasks in distributed environment with appropriate quality of services. Choice of load balancing algorithm for cloud environment depends on many factors: network infrastructure type (overload or underload approach), equipment CPU, data type. According to the principles of work load balancing techniques can be divided on statics and dynamics. The results of the experiment revealed that static load balancing algorithms work more stable in situation when the traffic intensity is predictable. The average round trip time depends of messages size for static algorithms less than for dynamics load balancing techniques. But in case when amount of end users grows the dynamic load balancing methods shows better result. For example, Central Manager, Least Connection and Tokin Routing algorithms introduce the highest throughput in the case when the amount of end users grows and, consequently, the data intensity grows.

List of references: 1. *J. W. Rittinghouse, J. F. Ransome* Computing Implementation, Management, and Security. CRC Press, New York, ISBN: 978-1-4398-0680-7, 2010, 340 p. 2. *R. M. Bryant and R. A. Finkel* A Stable Distributed Scheduling Algorithm in Proc. 2nd Int. Conf. Dist. Comp., pp. 341-323. 3. *B. Godfrey, K. Lakshminarayanan, S. Surana, R. Karp and I. Stoica* Load Balancing in Dynamic Structured P2P Systems // Performance Evaluation, Vol. 63, No. 3, 2006, pp. 217-240. 4. *S. C. Wang et all.* Towards a load balancing in a three level cloud computing network. Proceedings of 3rd International Conference on Computer Science and Information Technology (ICCSIT), IEEE, July, 2010, pp.108-113. 5. *X. Ren, R. Lin, H. Zou.* A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE. 2011, pp.220-224. 6. *Saroj Hiranwal, K. C. Roy.* Adaptive Round Robin Scheduling using shortest burst approach based on smart time slice // International Journal of Computer Science and Communication Vol. 2, No. 2, 2010, pp. 319-323. 7. *M. Mitzenmacher, E. Upfal*. Probability and Computing: Randomized Algorithms and Probabilistic Analysis. Cambridge University Press, 2005, 110 p. 8. *B. Yagoubi, and Y. Slimani* Task Load Balancing Strategy for Grid Computing // Journal of Computer Science, Vol. 3, No. 3, 2007, pp. 186-194. 9. *Ren X, R Lin, H Zou.* A dynamic load balancing strategy for cloud computing platform based on exponential smoothing forecast. proc. International Conference on. Cloud Computing and Intelligent Systems (CCIS), IEEE, 2011, pp. 220-224. 10. *B. Montero et all.* Virtual infrastructure management in private and hybrid clouds IEEE Internet Computing, 13(5), 2009, pp.14-22. 11. *Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma* Performance Analysis of Load Balancing Algorithms // Academy of science, engineering and technology, issue 38, February 2008, pp.269-272.

Харьковский национальный
университет радиоэлектроники

Поступила в редколлегию 11.10.2016