WILDCARD CONVENTION IMPLEMENTING IN TCAM TECHNIQUE FOR SOFTWARE DEFINED NETWORKS

Sowmiya.R[1], Radhakrishnan.C[2],

PG Student, Department of Computer Science Engineering,kongunadu college of engineering & Technology,

Assistant Professor,Department of Computer Science Engineering ,kongunadu college of engineering &Technology, Tamil nadu,India

[1]sowmiyacse003@gmail.com

[2]radhamayil@gmail.com

Abstract:-**The ascension of wireless content access implies the requirement for content placement and programming at wireless base stations. We tend to check a system below that users unit of measurement divided into clusters supported their channel conditions, and their requests unit of measurement delineate by whole totally different queues at logical front ends. Requests can be elastic (implying no burdensome delay constraint) or dead (requiring that a delay target be met). Correspondingly, we have got request queues that indicate the number of elastic requests, and deficit queues that indicate the deficit in dead service. Caches unit of measurement of finite size and will be unweary periodically from a media vault. we tend to require under consideration two worth models that correspond to dead requests for streaming hold on content and fundamental measure streaming of events, severally. we tend to vogue provably best policies that stabilize the request queues (hence guaranteeing finite delays) and reduce average deficit to zero [hence guaranteeing that the quality-of-service (QoS) target is met at little worth. we tend to tend as an instance our approach through simulations.**

Keyword- **Software-defined-networking, TCAM, rule dependency problem, wildcard rules caching algorithm.**

I. INTRODUCTION

Once being enclosed in an exceedingly route, the adversary starts dropping packets. within the most severe kind, the malicious node merely stops forwarding each packet received from upstream nodes, fully disrupting thepath between the supply and also the destination. Eventually, such a severe denial-of-service (DoS) attack will paralyze the network by partitioning its topology. Even though persistent packet dropping will effectively degrade the performance of the network, from the a packet loss rate at the malicious nodes makes this sort of attack simple to be detected. A malicious node that's a part of the route will exploit its knowledge of the network protocol and therefore the communication context to launch associate insider attack an attack that's intermit- tent, however are able to do an equivalent performance degradation effect as a persistent attack at a way lower risk of being detected. Specifically, the malicious node might measure the importance of varied packets, so drop the little amount that are deemed extremely vital to the operation of the network. Detecting selective packet-dropping attacks is very challenging in an exceedingly extremely dynamic wireless setting. The difficulty comes from the need that we want to not only observe the place (or hop) wherever the packet is born, but conjointly establish whether or not the drop. The match field determines which packets will match the principles. Once a packet matches a rule, the OpenFlow device can apply actions in actions field to the packet. Otherwise, the packet are born or sent to the controller by default rule. Counters field is employed when controller wishes statistics of the network reminiscent of link bandwidth or error rate. In the rules caching system, since wildcard rules could overlap with one another within the field area, packet classification policy assigns completely different completely different} priorities to different rules to avoid conflicts. If one packet matches multiple rules in an exceedingly flow table, the OpenFlow switch can apply the actions on the best priority matched rule to the packet. sadly, the rule dependency drawback needs to be solved within the wildcard rule caching system.

After caching wildcard rules in the TCAM, once cache miss happens, the RCR algorithmic rule would find victim rules and replace them with the cache miss rule in keeping with current temporal and spatial traffic localities. This paper has following 2 contributions. First, the wildcard rules caching algorithmic rule will utilize the TCAM house a lot of efficiently compared with the cover-set caching algorithmic rule. Second, the RCR algorithmic rule considering traffic section would maintain the hit magnitude relation high. Simulation results show that our wildcard rules caching algorithmic rule performs higher in cache hit ratio than cover-set caching algorithmic rule. And, the RCR algorithmic rule gets higher cache hit magnitude relation than least recently used (LRU) algorithm, random replacement (RR) algorithmic rule, and adjustive replacement cache (ARC) algorithmic rule.

II. RELATED WORK

Generally, the scale of a packet classification policy is far larger than the capability of Associate in Nursing OpenFlow switch. The thought of rules distribution on traffic route is moldering an outsized packet classification policy into smaller ones then distributing them across the network. If packets enter the network, they must traverse on their routes within the network to perform the complete packet classification. FIB caching is totally different from ancient caching schemes which may cause a cache-hiding drawback. If a packet features a matching prefix within the cache, it's going to not be the proper entry for forwarding the packet if there's a extended matching prefix in the full FIB. The planned theme will forestall the cache-hiding problem. In [21], the authors aim to hurry up the rule matching of high entropy packet fields. The high entropy packet fields are the values of packet fields that square measure often modified from packet to packet flowing through a switch, like layer four port field. The planned rule will support flow caching of forwarding choices as well as L4 headers, while not requiring every new forwarded transport affiliation to be handled by the slow path. The OpenFlow switch can cache the bucket matched by current packets within the first stage of flow tables. At an equivalent time, all wildcard rules falling within the field space of the bucket are cached within the second stage of flow tables. With such a bucket theme, the management information measure is saved and the linguistics correctness of packet matching may be complied.

III. EXISTING SYSTEM

In software-defined networking, flow tables of Open Flow switches area unit enforced by ternary content address- ready memory (TCAM).Though TCAM will method input packets in high speed, it's a scarce and costly resource providing solely a number of thousands of rule entries on a network switch. Rules caching could be a technique to resolve the TCAM capability downside. However, the rule dependency downside could be a difficult issue for wildcard rules caching wherever packets will twin rules.during this paper, we tend to use a cover-set approach to resolve the rule dependency downside and cache vital rules to TCAM. we tend to conjointly propose a rule cache replacement algorithmic program considering the temporal and spatial traffic localities. Simulation results show that our algorithms have higher cache hit magnitude relation than previous works. Alerts as specified by the policies. Third, we have a tendency to perform extra experiments on the GENI test bed to guage the measurability of the planned framework mistreatment existing datasets of field networks.

DRAWBACKS OF EXISTING SYSTEM

Previous work on TCAM size capability downside falls into 3 main categories:

- Packet classification compression
- Rules distribution on traffic route
- Rules caching

IV. PROPOSED SYSTEM

We have a tendency to tend to develop algorithms for content distribution with elastic and dead requests. we have a tendency to tend to use document of invite queue to implicitly verify the popularity of elastic content. Similarly, the deficit queue determines the specified service for dead requests. Content is additionally recent periodically at caches. we have a tendency to tend to check two wholly totally different types of worth models, each of that's appropriate for a singular content distribution state of affairs. the first is that the case of file distribution (elastic) beside streaming of hold on content (inelastic), where we have a tendency to tend to model worth in terms of the frequency thereupon caches unit recent. The second is that the case of streaming of content that is generated in amount of your time, where content expires once a certain time, and additionally the worth of placement of each packet among the cache is taken into consideration.

ADVANTAGES OF PROPOSED SYSTEM

- It stabilizes the system load among the potential region.
- Minimizes the common expected worth whereas helpful the deficit queues

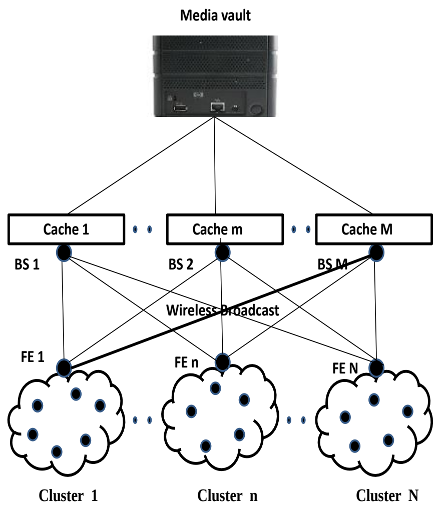

Fig .1 system arichitecture

V. IMPLEMENTATION

WILDCARD RULES CACHING ALGORITHM

Cover-set caching rule planned in calculates contribution worth of every rule R. The contribution worth of rule R is defined because the quantitative relation of R.weight to R.cost . The contribution worth may be a metric for caching rules. Cover-set caching algorithm would iteratively calculate contribution worth of every un-cached rule and cache the rule that has the most contribution value till there's no on the market TCAM entry. Denotes a dependency graph of wildcard rules in Fig. one and also the weight of every rule. In general, the lower priority rule has larger rule weight. Assume the TCAM size is three. First, the cover-set caching rule would cache. Our wildcard rules caching formula calculates accumulated contribution worth of every un-cached rule by formula one. Then, we have a tendency to cache a group of rules that has the most accumulated contribution worth. The higher than steps area unit referred to as a spherical in our wildcard rules caching formula. we have a tendency to repeat rounds till there is no remaining TCAM entry. We evaluate the performance of every caching algorithm within the cache hit quantitative relation. Second, we tend to compare our RCR formula with LRU and RR algorithms that area unit typical cache replacement algorithms in memory management. We also compare the RCR formula with ARC formula that maintains LRU lists that keep trailing of recently used parts and frequently used parts. Here, the cache hit quantitative relation is also used because the performance metric for the cache replacement algorithms.

RULE CACHE REPLACEMENT (RCR) ALGORITHM

After caching wildcard rules in step with their weights, the input packets will match rules either within the TCAM (cache) or code switch. Once cache miss happens, RCR algorithmic rule would replace victim cached rules with the cache miss rule. If the cache hit magnitude relation is high, we are able to avoid miss penalties. So, the hit magnitude relation is one in all the foremost necessary metrics for measuring the speed of process incoming packets of a switch. The goal of the RCR algorithmic rule is to extend the cache hit magnitude relation. We assign a counter for every rule to represent its temporal traffic locality. Once cache miss happens, the RCR rule replaces the victim cached rule that has all-time low worth with the cache miss rule. Then, set the cache miss rule's worth according to its neighboring rules' values (considering special traffic locality).

PACKET CLASSIFICATION COMPRESSION

Packet classification compression tries to use less variety of TCAM entries to represent another semantically equivalent packet classification. The compression theme planned in will solely be employed in prefix classifier. Prefix classifier is a reasonably packet classification wherever every field of a rule's predicate is specified as a prefix. All the *'s square measure at the tip of the ternary string. Bit Weaving is that the first compression scheme for non-prefix packet classifiers. In Bit Weaving, if match fields of 2 entries disagree by only 1 bit and their action fields square measure same, these 2 entries will be united. The distinction bit is replaced with a wildcard bit (*). we have a tendency to might either use Bit Weaving alone or create Bit Weaving as a post-processing routing to upset different compression schemes. however, the network application or administrator cannot get statistics of the original rule (e.g. flow count). the rationale is that once we combine 2 rules, each of their counter fields also are united. We can solely get the combined statistics instead of separated statistics of the first rule.

CONCLUSION

In wildcard rules caching, the cover-set is Associate in Nursing efficient talent to solve rule dependency drawback. Compared with the cover-set caching algorithmic program, we have a tendency to calculate the accumulated contribution value of a group of rules rather than the individual contribution value of a rule. Therefore, the performance of our rules caching algorithm is healthier than the cover-set caching one. Besides, we propose Associate in Nursing RCR algorithmic program to contemplate the

Traffic temporal and spatial localities. If the cache miss happens and therefore the current traffic locality is near to the incomprehensible rule, we have a tendency to replace a victim rule with the cache miss rule and set a high worth to the in comprehensible rule to stay the rule the TCAM. Otherwise, the worth of cache miss rule is about to low and it are often replaced shortly once. Simulation results show that the RCR has higher cache hit magnitude relation than LRU, RR, and ARC schemes. Our future work includes: Investigate different potential wildcard rules caching algorithmic program to improve the cache hit magnitude relation and refine our cache replacement algorithm to calculate the burden worth of the cached rules which is additional adapt to the traffic neighborhood.

REFERENCES

[1] M. Reitblatt, N. Foster, J. Rexford, C. Schlesinger, and D. Walker, "Abstractions for network update," in Proc. ACM SIGCOMM, Helsinki, Finland, Aug. 2012, pp. 323–334.

[2] N. P. Katta, J. Rexford, and D. Walker, "Incremental consistent updates," in Proc. 2nd ACM SIGCOMM Workshop Hot Top. Softw. Def. Netw., Hong Kong, Aug. 2013, pp. 49–54.

[3] X. Jin et al., "Dynamic scheduling of network updates," in Proc. ACM SIGCOMM, Chicago, IL, USA, Aug. 2014, pp. 539–550.

[4] N. Gude et al., "NOX: Towards an operating system for networks," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 3, pp. 105–110, Jul. 2008.

[5] Floodlight [Online]. Available: http://www.projectfloodlight.org/floodlight/

[6] RYU [Online]. Available: http://osrg.github.io/ryu/

[7] N. McKeown et al., "Openflow: Enabling innovation in campus net- works," ACM SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69–74, Apr. 2008.

[8] K. Lakshminarayanan, A. Rangarajan, and S. Venkatachary, "Algorithms for advanced packet classification with ternary CAMs," in Pro. ACM SIGCOMM, Philadelphia, PA, USA, Aug. 2005, pp. 193–204.

[9] K. Kogan, S. Nikolenko, O. Rottenstreich, W. Culhane, and P. Eugster, "SAX-PAC (scalable and expressive packet classification)," in Proc. ACM SIGCOMM, Chicago, IL, USA, Aug. 2014, pp. 15–26.

[10] M. Casado et al., "Rethinking enterprise network control," IEEE/ACM Trans. Netw., vol. 17, no. 4, pp. 1270–1283, Aug. 2009.

[11] C. R. Meiners, A. X. Liu, and E. Torng, "Bit weaving: A non-prefix approach to compressing packet classifiers in TCAMs," IEEE/ACM Trans. Netw., vol. 20, no. 2, pp. 488–500, Apr. 2012.

[12] Y.-C. Cheng and P.-C. Wang, "Packet classification using dynamically generated decision trees," IEEE Trans. Comput., vol. 64, no. 2, pp. 582– 586, Feb. 2015.

[13] A. X. Liu, C. R. Meiners, and E. Torng, "TCAM razor: A systematic approach towards minimizing packet classifiers in TCAMs," IEEE/ACM Trans. Netw., vol. 18, no. 2, pp. 490–500, Apr. 2010.

[14] X.-N. Nguyen, D. Saucez, C. Barakat, and T. Turletti, "Optimizing rules placement in openflow networks: Trading routing for better efficiency," in Proc. 3rd Workshop Hot Top. Softw. Def. Netw., Chicago, IL, USA, Aug. 2014, pp. 127–132.