Paper Review by [Chencheng Liang]

1. Paper Information

Paper Title: HULA: Scalable Load Balancing Using Programmable Data Planes

Type of Paper: Full Paper (9 or more pages)

2. Summary of the Paper

This paper proposes a data-plane load-balancing algorithm called HULA(Hop-by-hop Utilization-aware Load balancing Architecture) to address two main issues (memory shortage and infeasible software) existing in congestion-aware load-balancing techniques.

In the first section, this paper introduces the issues existing in congestion-aware load-balancing techniques briefly, and explains how HULA can resolve them roughly.

In the second section, this paper illustrates four challenges existing in large datacenter networks, including large path utilization matrix, large forwarding state, discovering uncongested paths, and programmability. HULA also needs to face these challenges.

In the third section, it explains how HULA deals with these four challenges. For the first challenge, a HULA switch only maintains a table that maps the destination ToR (Top of Rack switche) to the best next hop as measured by path utilization. when receive multiple probes coming from different paths to a destination ToR, a switch picks the hop that saw the probe with the minimum path utilization. And the switch sends its view of the best path to a ToR to its neighbors. For the second challenge, HULA's best hop table eliminates the need for separate source routing in order to exploit multiple network paths. For the third challenge, HULA has Automatic discovery of failures and Proactive path discovery. HULA relies on the periodic arrival of probes as a keep-alive heartbeat from its neighboring switches. For the programmability, Processing a packet in a HULA switch involves switch state updates at line rate in the packet processing pipeline. This data-plane architectures can be configured through a common programming language like P4, which allow operators to program stateful data-plane packet processing at line rate. And by dealing with these challenges HULA becomes Scalable, Proactive, Adaptive, and Programmable.

The fourth section explains HULA design by four subsections.

The probe replication mechanism: it makes it easy to incrementally add switches to an existing set of multicast groups for replication.

The logic behind processing probe feedback: it process probes to update best path by the probe propagation procedure, which ensures that if the best path changes downstream, then that information will be propagated to all the relevant upstream switches on that path.

How the feedback is used for flowlet routing: the time gap between consecutive flowlets will absorb any delays caused by congested paths when the flowlets are sent on different paths.

How HULA adapts to topology changes: HULA tracks the last time bestHop was updated using an updateTime table. If a bestHop entry for a destination ToR is not refreshed within a last threshold for detecting failures, then any other probe that carries information about this ToR (from a different hop) will simply replace the bestHop and pathUtil entries for the ToR.

The probe overhead (probeSize*numToRs*100/probeFreq*linkBandwidth) and optimization(by allowing only one probe is sent per destination ToR within this time window on any link) are also discussed.

In fifth section, First it introduces what is P4, and Programming HULA in P4 allows a network operator to compile HULA to any P4 supported hardware target. The P4 program has two main components: one, the HULA probe header format and parser specification which is shown in Figure 3 (a), and two, packet control flow, which describes the main HULA logic and is shown in Figure 3 (b), in which, HULA probes are processed in line 4-13, and flowlet forwarding is described in line 15-20.

The sixth section illustrates the effectiveness of the HULA load balancer by implementing it in the ns-2 discrete event simulator and comparing it with another two load balancing schemes ECMP and CONGA. It compares HULA and another two load balancing schemes in four aspects: performance in the baseline topology, performance when there is asymmetry in the network, response time for adapting to changes in the network, and robust.

The last section mentions some related works, such as stateless or local load balancing, centralized load balancing, modified transport layer, and global utilization-aware load balancing.

In the conclusion section, this paper emphasizes again that HULA is effective enough to quickly adapt to the volatility of datacenter workloads with a satisfied performance and stability, and HULA is also simple to implement at line rate in the data plane on emerging programmable switch architectures. No further work is mentioned.

3. Contributions of the Paper

(What are the major issues addressed in the paper? Do you consider them important? Comment on the degree of novelty, creativity, impact, and technical depth in the paper.)

Congestion-aware load-balancing techniques such as CONGA have two main limitations.

- For overcoming memory limitation in switches, this paper propose that instead of having the leaf switches track congestion on all paths to a destination, each HULA switch tracks congestion for the best path to a destination through a neighboring switch. Instead of maintaining path utilization for all paths to a destination ToR, a HULA switch only maintains a table that maps the destination ToR to the best next hop as measured by path utilization.
- For avoiding to use customized hardware, HULA can be run on programmable chipsets for overcoming the custom hardware problem. And this paper run a simulation by ns-2 to show that it delivers up to 3.3 times better flow completion times than state-of-the-art congestion-aware load balancing schemes at high network load.

4. Strengths of the Paper

(What are the main reasons to assess this paper as high-quality? e.g. innovative ideas/technologies, paper presentation, detailed performance evaluation.)

- This paper illustrates the basic knowledge well for understanding its idea. For example, it illustrates why we need to resolve these issues, and what challenges we have in data center load balancing.
- This paper has few distractions and redundant content. The main issues(the purpose of this paper) always been mentioned in different part of this paper, so the reader can always keep focus.

5. Weaknesses of the Paper

(What are the most important reasons to assess this paper as low-quality? e.g. writing skill, technical errors, unrealistic assumptions, unanswered questions, limited measurements or evaluation)

- This algorithm contains many steps and formulas, but there is no any mathematical reasoning that can directly clarify some formulas or some steps that are the best or better mathematically.
- There are figures for illustrating the processes and procedures of the algorithm, but I think it is better to add a simple example which can as much as possible cover all steps and procedures of this algorithm.
- In the evaluation part, the parameters of the reference group (ECMP and CONGA) are not specified. Maybe if change the parameters of reference group to proper parameters, some performances of reference group my surpass HULA.

- In section 7, many related works are mentioned briefly, many weaknesses are pointed. But it is better to talk more about what relations between the weaknesses and HULA, where HULA can improve, where these works may doing better or worse than HULA. And where is differences between these works and HULA.
- Some procedures just been proposed without specifying why (maybe author can say where he/she get this inspiration, why she/he can get this idea, this idea based on what) , such as in section 4.2 the procedure of calculating the min-max path utilization.

6. Open Issues and Future Work

(What are the options for future work that you think are important? If you are asked to work on the problem investigated in this paper, what will you do differently?)

Open Issues:

- Even though there are simulations which shows HULA has better performance in many aspects, but there is no mention or comparing about theory complexity of this algorithm and other algorithm.
- It is better to give a mathematical proof that can prove that HULA has a better algorithm complexity or space complexity, because in simulation, there are many different parameters and situations, which can only represent a specific scenario.

Future work:

- For the parameters, can we find an algorithm or method which can generate the optimized parameters according to the conditions.
- Except compare with ECMP and CONGA, we can compare this algorithm with other new or old algorithms find the weaknesses and improve it.

7. Your Questions to the Authors

(What (technical) question would you ask the authors if you were in the audience of their talk?*)*

- The algorithm is proved useful when it is compared with other algorithms in simulation. There are two parameters, one is flowlet gap which is set to 100 µs, two is the probe frequency which is set to 200 µs. If we adjust this two parameters, weather the performance of HULA will be worse than other algorithm?
- In section 4.2, the link utilization is based on an exponential moving average generator (EWMA), why use this form?