Simultaneously Reducing Latency and Power Consumption in OpenFlow Switches

M. Dharani Priya and S. Perumal

Abstract--- Ethernet LAN that uses switches to connect individual hosts of segments. This type of network is sometimes called a desktop switched Ethernet. In the case of segments the hub is replaced with a switching hub. TCMAs takes lot of amount of power and when it is matched flows in OpenFlow switch , put a bound on switch latency. As network technologies converge upon a single Ethernet fabric, there is ongoing pressure to improve the performance and efficiency of the switch while maintaining flexibility and a rich set of packet processing features. The enhancement is that using a OpenFlow Ethernet with the pre-port packet prediction so that the latency is reduced and the reduction in consumption of power which is enabled by the OpenFlow architecture. Under the TCMAs it was that the power consumption was high. But in OpenFlow Ethernet it is reduced.

Keywords--- Open Flow Switches, SDN Domain, Network Architecture

I. Introduction

Open Flow Switch

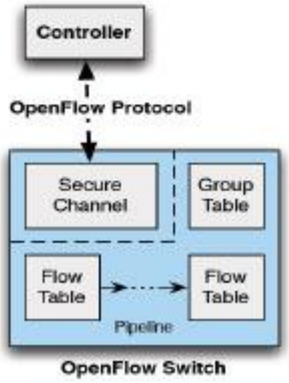
In a classical router of switch, the fast packet forwarding (data path) and the high level routing decisions(control path) occurs on the same device. An open flow switch separates these two functions.

The data path portion still resides on the switch, while high level routing decisions are moved to separate flow protocol, which defines messages, such as packet received, send-packet-out, controller, typically a standard server. The open flow switch and controller communicate via the open modify-forwarding table, and get-stats.

The data path of an open flow switch presents a clean flow table abstraction; each flow table entry contains a set of packet fields to match, and an action (such as send out port modify field or drop). When an Open Flow switch receives a packet it has never seen before, for which it has no matching flow entries, it sends this packet to the controller. The controller then makes a decision on how to handle this packet it can drop the packet, or it can add a flow entry direction the switch on how to forward similar packets in the future.

The open flow switch may be programmed to:

- Identify and categorize packets from an ingress port based on a various packet header fields
- Process the packets in various ways including modifying the header
- Drop or push the packets to a particular ingress port to the open flow controller



Latency:-Latency is the amount of time a message takes to traverse a system. In a computer network it is an expression of how much time it takes for a packet of data to get from one designated point to another. It is sometimes

M. Dharani Priya, B.C.A., MSc (Computer Science), Purusing M.Phil(VELS University)
S. Perumal, Head, Computer Science Department.

measured as the time required for a packet to be returned to its sender . It depends on the speed of the transmission medium and the delay in the transmission by devices along the way a low latency indicates a high network efficiency.

SDN (Software Defined Networking)

Software defined networking programmable network aligned to business applications delivers agility.

Unleashing a new wage of network innovation in security cloud mobility and big data.

SDN Domain and Network Archticture

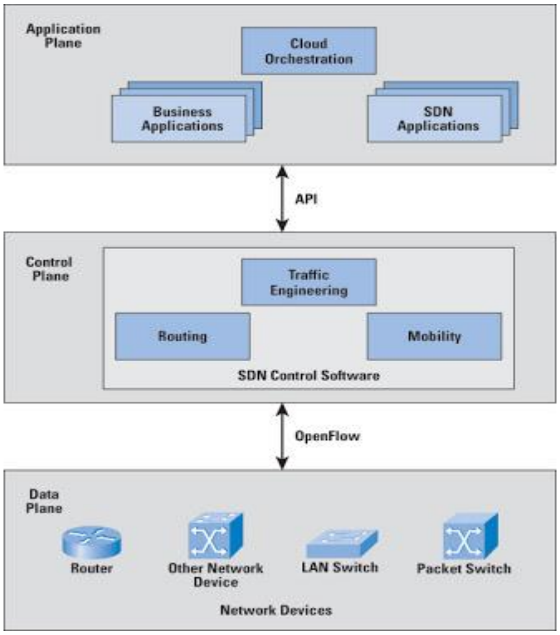
Software defined networking domain is used for the reason:

- Scalability: The number of device an SDN control can thus a reasonably large network may need to deploy multiple SDN controllers.
- Privacy: A carrier may choose to implement different SDN domain.
- Incremental deployment: A carrier network consists of portions of traditional and newer infrastructure. Dividing the network into multiple individually manageable SDN domains allow for flexible incremental deployment.

The open flow network architecture consists of 3 layers

- One or more open flow virtual and or physical switches
- One or two open flow control less and
- One or more open flow application(s).

II. OPEN FLOW NETWORK ARCHITECTURE:



The architecture is similar to one of many described in [11], but with an emphasis on flow-based switching where the logic is implemented in a single chip on the line card and takes advantage of a TCAM for flow matching. The line cards are equipped with separate input and output memory, lookup logic, backplane fabric interfaces, and per-port prediction circuitry. The lookup and policy logic may involve multiple layer-2 and layer-3 address tables in addition to a TCAM used to support per-flow forwarding features the high-level switch architecture.

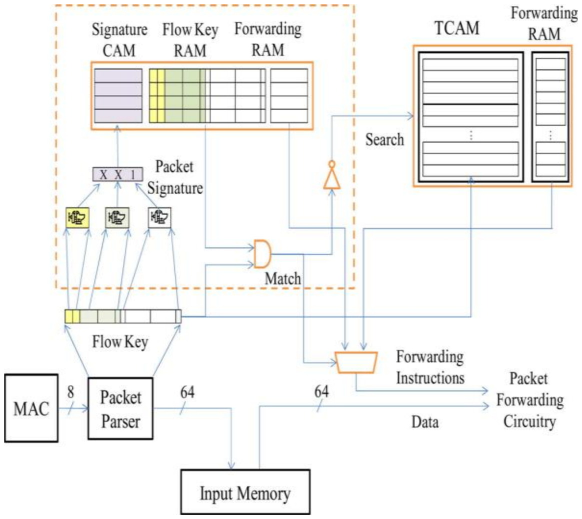
III. PACKET PREDICTION CIRCUITRY

Once a complete flow-key has been received and assembled by the packet parser, there are three possible scenarios that may occur with respect to the prediction logic and speculative forwarding.

1) Prediction hit: The flow-key matches a flow-key found by the prediction logic. In this case, a correct prediction has occurred and there is no need to take any further action. No lookup or search is required of the master TCAM, and the power required to perform that search is saved. The lowest possible latency is achieved because packet forwarding has already started and the speculation was correct.

2) Incorrect prediction: A signature is found, but the flow-key does not match. In this case, an incorrect prediction has occurred, and the current speculative transfer must be aborted. The master TCAMmust be searched to determine the correct forwarding instructions, and the local prediction cache must be updated. The power for the prediction cache searches, and the partial packet transfer is wasted.

3) Prediction miss: No flow-key was found by the prediction logic. In this case, the prediction cache did not find amatch, and the full lookup process must be invoked. The local prediction cache must be updated. The power required for searching and updating the prediction cache is wasted.



This paper considers two different methods that trade off implementation complexity for accuracy.

1) Direct Map: The extremely simple Direct Map method extracts bits from predefined locations in the packet as it is arriving. The offset locations have been chosen to include bit fields that vary the most between subsequent flows. There is no logic that parses the packet and adjusts the offsets according to frame encapsulation or protocol.

Bits are blindly extracted at predetermined offsets. As a consequence, bit offsets that would normally align with the TCP port fields in an untagged Ethernet packet will be unaligned if the packet is VLAN tagged. Similarly, these bit offsets may point to random payload data if the packet is an IP fragment (which contains no TCP header). For a practical implementation of the Direct Map method, a different set of offsets should be considered based upon the port configuration.

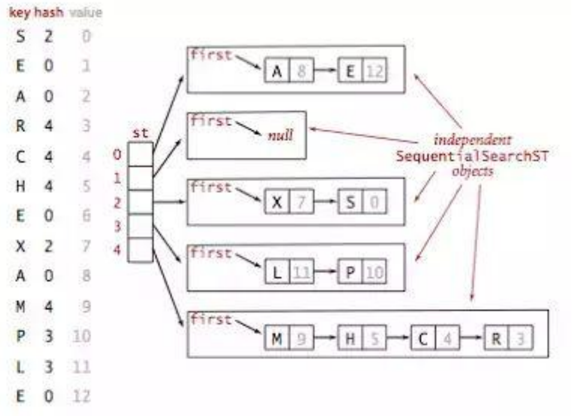
As the bits arrive from fixed offsets, the circuitry builds a partial signature to present as a key to the fully associative prediction cache. Missing bits that have not yet arrived are marked as don’t care conditions for the match. If no matching entries are found, there are clearly no previous elements from this flow in the cache and the packet must wait for the full flow lookup to complete. If there is precisely one entry found, then there is a chance that this entry is an exact match and the speculative forwarding of the packet may start immediately. This method forwards the packet as soon as possible, but can experience a higher misprediction rate than methods that perform more intelligent parsing.

2) Sub-FieldHash: The Sub FieldHash method intelligently parses incoming packets to extract precise subfields of packet headers and uses a simple hashing algorithm to construct segments of the packet signature. The goals of this method are to minimize the number of incorrect predictions, but also to support low latency by including a relatively aggressive eager approach to searching the prediction cache.

The popular DJB hash function [21] is applied to subfields of the 29-B flow-key as they arrive. The DJB hash function was chosen because of its simplicity, efficiency, and distribution characteristics over smallfields. The small hash results are combined to create partial signatures. Similar to the Direct Map method, the prediction cache is searched as soon as a partial signature has been formed, where missing bits of the signature are marked as don’t care conditions. The number of times the prediction cache is searched depends upon the length of the packet signature and the result of previous searches. A number of different packet signature sizes have been chosen to evaluate this sensitivity. The Sub-Field Hash method generates signatures of lengths 8, 16, 24, and 32 bits. For an8-bit signature, the flow-key is divided into two parts; the MAC header is hashed to create a 4-bit partial signature, and the IP and TCP headers are used to create another 4-bit quantity. These two quantities are combined to create an 8-bit signature with which the prediction cache is searched at most two times per packet. Longer signature types allow a greater number of partial signatures and thus may be more aggressive at speculating the forwarding of the packet. However, they will also search the cache more frequently, consuming more power. For example, in the implementation the 16-bit signature is made up of 5 hashes and in the worst case will search the cache 5 times. The 24-bit signature includes 9 hashes, and the 32-bit signature includes 11 hashes.

Algorithm used for this openflow switch is

It is generalized as n-tuple that is defined by a set H={H1,H2,…..Hn}of fields from packets.



Hashing with separate chaining for standard indexing client

Reducing TCAM Power: Reducing the power consumption of network devices that use TCAMs has been the focus of a number of different studies. A common technique in most previous approaches is to segment the TCAM into blocks and only search individual blocks as needed. Additional front-end logic is provided to locate the individual block that likely contains the search entry. Power is saved by only driving current through the lines of that individual block. The use of TCAM segmentation and intelligent organization to reduce power consumption during search is orthogonal to the prediction enhancement and in fact is compatible with the approach that further aims to reduce power by simply bypassing these operations.

Latency Reduction Using Prediction: The goal of reducing the number of pipeline stages in a routing switch by using predictive switching was proposed in [40]. The proposed technique looks at the forwarding history of an ingress port to predict the egress port, irrespective of the contents of the packet. The 2-D torus network for which this prediction scheme was developed is a connection-oriented switch for specialized high-performance computing applications. The authors achieved 77% prediction accuracy using the NAS parallel benchmarks.

IV. CONCLUSION

Enhancing an OpenFlow switch with per-port packet prediction circuitry is an effective means for simultaneously reducing power and switch latency without sacrificing flexibility and rich packet processing. Two different prediction methods that trade off per-port complexity for accuracy where shown to be effective. The more accurate Sub-Field Hash method is more effective at reducing power consumption because of a lower incorrect prediction rate, while equivalent latency reduction can be achieved even with the simplistic Direct Map method. The results of simulations using real network data have shown that packet prediction can reduce the latency of a traditional store-and-forward switch by nearly a factor of 8 and reduce the already low latency of a cut-through switch by a factor of 3. Depending upon the locality of the network traces, the average energy required in the lookup phase of an OpenFlow-based Ethernet switch can simultaneously be reduced as well. While all parts of the network topology can benefit from a switch with the proposed circuitry, the high-performance computing cluster gains the most.

REFERENCE

[1] Cisco Systems, Inc., San Jose, CA, USA, “Cisco Nexus 3548 switcharchitecture,” 2012.

[2] B. Salisbury, “TCAMs and OpenFlow—What every SDN practitioner must know,” SDN Central, 2012 [Online].

[3] R. Ozdag, “Intel ethernet switch FM6000 series—Software defined neworking,” Intel Corporation, 2012, p. 8.

[4] Arista Networks, Inc., Santa Clara, CA, USA, “7150 series 1/10 GbE SFP ultra low latency switch,” 2012.