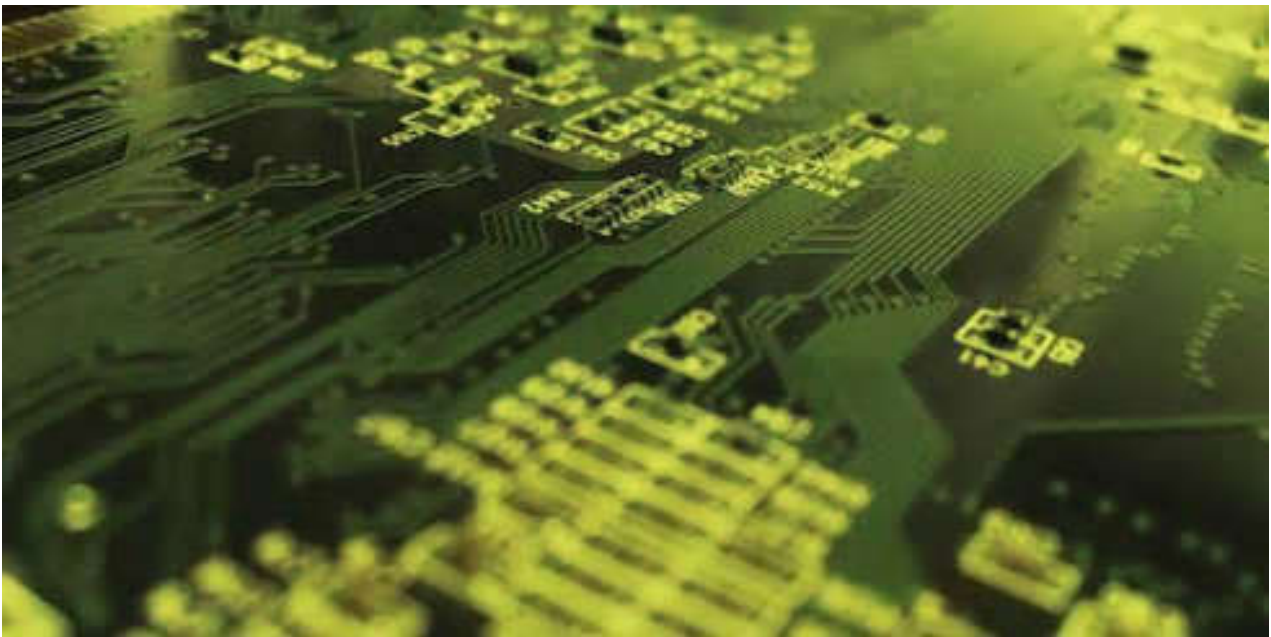


[JOIN](#)[LOGIN](#)

[SD-WAN](#) [5G](#) [MEC](#) [IoT](#) [SDN](#) [NFV](#) [Containers](#) [Cloud](#) [Security](#) [AI](#) [Converged DC](#)  
[SD-Storage](#) [NPM/APM](#)

# TCAMs and OpenFlow – What Every SDN Practitioner Must Know

SDN CENTRAL



Brent Salisbury (<https://www.sdxcentral.com/author/brent-salisbury/>)

July 1, 2012

4:00 pm PT

Trying to negotiate SDN-enabled hardware today is not an easy thing to do. This is primarily due to current networking hardware not designed for a decoupled control plane. Yet, the release of the [OpenFlow](https://www.sdxcentral.com/sdn/definitions/what-is-openflow/) (<https://www.sdxcentral.com/sdn/definitions/what-is-openflow/>) v1.0 Spec coupled with customer demand has driven vendors to provide OpenFlow-networking gear. Using existing off the shelf hardware, manufacturers have taken their network operating systems (NOS) and interrupted the normal pipeline processing to fork selected packet flows into the OpenFlow pipeline. In order to achieve this level of flexibility on such refined, purpose-built hardware switches, it requires flexibility and performance in its forwarding tables. The quick fix for the early stages of experimentation and proofing on real world hardware is leveraging Ternary Content Addressable Memory (TCAM).

**Why have we heard so much about TCAMs lately?**

## Related Articles



[Is There  
a](#)



[Security Crisis in the Open Source Software Supply Chain?](https://www.sdxcentral.com/articles/contributed/security-crisis-in-the-open-source-software-supply-chain/2018/05/?c_action=related_articles) ([https://www.sdxcentral.com/articles/contributed/security-crisis-in-the-open-source-software-supply-chain/2018/05/?c\\_action=related\\_articles](https://www.sdxcentral.com/articles/contributed/security-crisis-in-the-open-source-software-supply-chain/2018/05/?c_action=related_articles))



[Securing the Multi-Cloud: It Takes a Village](https://www.sdxcentral.com/articles/contributed/securing-the-multi-cloud-it-takes-a-village/2018/05/?c_action=related_articles) ([https://www.sdxcentral.com/articles/contributed/securing-the-multi-cloud-it-takes-a-village/2018/05/?c\\_action=related\\_articles](https://www.sdxcentral.com/articles/contributed/securing-the-multi-cloud-it-takes-a-village/2018/05/?c_action=related_articles))



[Get Ready for Real Disruption: The State of NFV](https://www.sdxcentral.com/articles/contributed/the-for-real-disruption-the-state-of-nfv/2018/05/?c_action=related_articles) ([https://www.sdxcentral.com/articles/contributed/the-for-real-disruption-the-state-of-nfv/2018/05/?c\\_action=related\\_articles](https://www.sdxcentral.com/articles/contributed/the-for-real-disruption-the-state-of-nfv/2018/05/?c_action=related_articles))



[How to Manage Cloud Expenses With a Solution You Probably Already Have](https://www.sdxcentral.com/articles/contributed/managing-cloud-expenses/2018/05/?c_action=related_articles) ([https://www.sdxcentral.com/articles/contributed/managing-cloud-expenses/2018/05/?c\\_action=related\\_articles](https://www.sdxcentral.com/articles/contributed/managing-cloud-expenses/2018/05/?c_action=related_articles))



[Supply Chain Management Is Critical for Carrier Neutral SD-WAN](https://www.sdxcentral.com/articles/contributed/supply-chain-management-is-critical-for-carrier-neutral-sd-wan/2018/05/?c_action=related_articles) ([https://www.sdxcentral.com/articles/contributed/supply-chain-management-is-critical-for-carrier-neutral-sd-wan/2018/05/?c\\_action=related\\_articles](https://www.sdxcentral.com/articles/contributed/supply-chain-management-is-critical-for-carrier-neutral-sd-wan/2018/05/?c_action=related_articles))

Today the supported protocol used to instantiate a flow into vendor switches who support [SDN](https://www.sdxcentral.com/sdn/definitions/why-sdn-software-defined-networking-or-nfv-network-functions-virtualization-now/) (<https://www.sdxcentral.com/sdn/definitions/why-sdn-software-defined-networking-or-nfv-network-functions-virtualization-now/>) is OpenFlow v1.0. Some manufacturers have begun supporting v1.1 and greater, but none of that is shipping in GA. Along that thread, most vendors appear to be skipping v1.1 all together for v1.2 and v1.3.

Generally speaking, we can look at hardware spec sheets with the layers and protocols we have in our industry paradigm today and understand capabilities of hardware and how they are weaved or wedged into our networks. Though we are still left programming essentially firmware on each device it is just the reality of the industry for now.

### Highlights of the OpenFlow Release Specs

- Version 1.1 – Multi-table pipeline processing, MPLS, QinQ.
- Version 1.2 – IPv6 and additional extensibility.
- Version 1.3 – (Release Pending) QOS additions among others.

Simply put, the first packet of a flow checks the OpenFlow forwarding tables for a match, if one is not found, it punts to the controller and the controller inserts a flow entry in the switches forwarding table. *if match; then forward via flow table, else; forward to controller*. With OpenFlow v1.0, early hardware adopters use Ternary Content Addressable Memory (TCAM) to insert a flow from a controller into the forwarding tables or forwarding information base (FIB) for subsequent flow lookups of a flow. That does not give much room for hardware differentiation other than having external TCAM on board for more flow tables in TCAM.

## TCAM and OpenFlow

TCAM naturally lends itself to being great for flow instantiation from an SDN controller to inject forwarding entries with a great deal of flexibility across various Ethernet and IP header fields e.g. 10-tuple OpenFlow v1.0 header fields *Ingress Port, Ethec Src, Ether Dst, Ether Type, Vlan ID, IP Dst, IP Src, TCP Dst, TCP Src, IP Proto*.

Most vendors have TCAM in place for match things such as IP access-lists. That TCAM using software has been modified through either a HAL or some encoder like software algorithm to expand the width of the TCAM for OpenFlow lookups.

Ingress Port	Ethec Src	Ether Dst	Ether Type	Vlan ID	IP Dst	IP Src	TCP Dst	TCP Src	IP Proto
--------------	-----------	-----------	------------	---------	--------	--------	---------	---------	----------

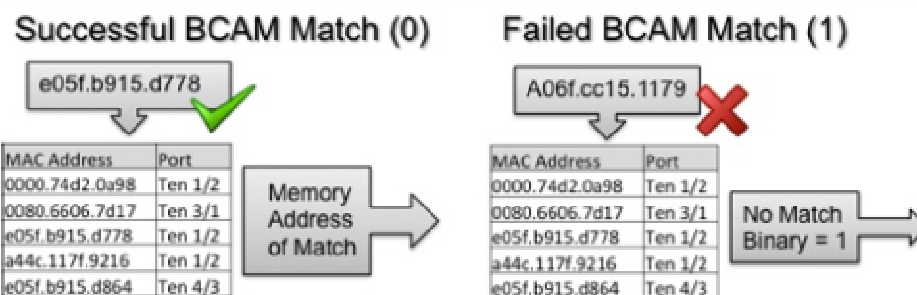
**Figure 1.** 12-Tuples of headers.

## Content Addressable Memory

CAM is a unique memory that can do memory lookups in one clock cycle and in a parallel fashion looking at multiple fields at once in a lookup. This is the most common way to accommodate hardware search ASICs. For programmers, CAM can be described as associative array logic.

- Two type typical hardware lookups are Binary CAM lookups (BCAM) and Ternary CAM lookups (TCAM)

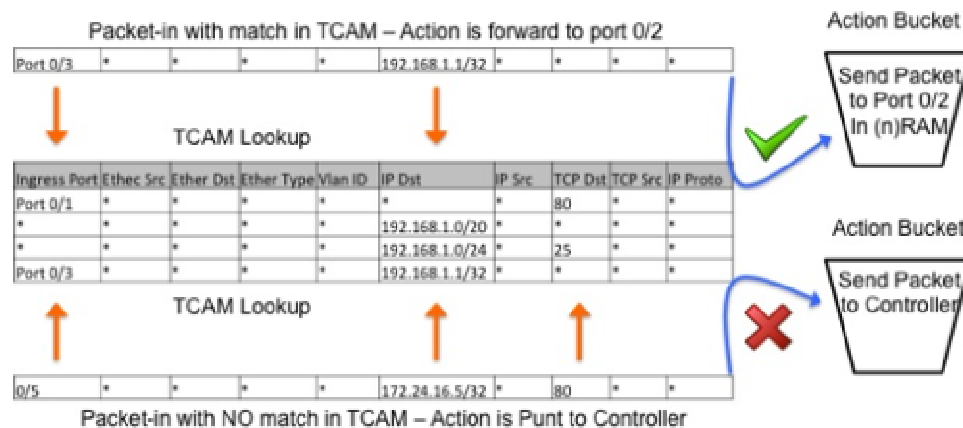
**Binary CAM** – As the word binary implies, is a binary lookup at returns either a 1 or 0. As exemplified in Figure 2, a mac address in an Ethernet frame comes into a switch, the switch looks in its mac address table and either finds that make address or doesn't, 1 or 0.



(<https://www.sdxcentral.com/wp-content/uploads/2012/06/image004.png>)

**Figure 2.** Binary CAM operation example

**Ternary CAM** – TCAM adds a 3<sup>rd</sup> additional component as the name ‘ternary’ suggest. TCAM lookups match explicit 1 and 0 but also have a ‘don’t care’ bit. You will also hear this referred to with regard to OpenFlow as a wildcard bit often depicted to in diagrams with a ‘\*’. TCAM can have multiple matches and determine a best match. This is necessary since CIDR lookups need to support a longest prefix match. Example 192.168.1.7/32 would match on 192.168.1.0/24 and 192.168.1.0/25. The closest match to 192.168.1.7/32 is 192.168.1.0/25 and would be chosen.



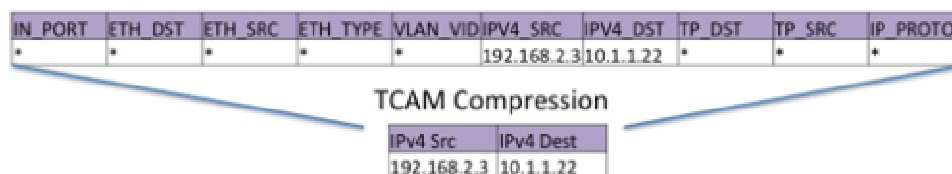
(<https://www.sdxcentral.com/wp-content/uploads/2012/06/image006.png>)

**Figure 3.** Ternary CAM OpenFlow operation example

### TCAM Limitations

TCAM is not the cure all for hardware search lookups. TCAM is power hungry, expensive and takes up quite a bit of silicon space. It would not be unusual for it to be the most expensive component on commodity switches. Many vendors use a blend of BCAM memory, SRAM, NPU and software algorithms to perform ternary lookups (see MX trio chipset and EzChip NPUs in ASRs, Trie lookups).

There are techniques depending on how many explicit matches are made across n-tuples to mask wildcard fields in a TCAM entry. You may hear from vendors that flow table capacity will depend on your desire to match a given number of tuples. If you are looking for exact matches on all tuples in every flow it may affect the total number of supported flows while matching a couple tuples in flows with the ability to mask more fields. That said, there are incredibly clever software algorithms that can even overcome full flow match limitations to some degree.

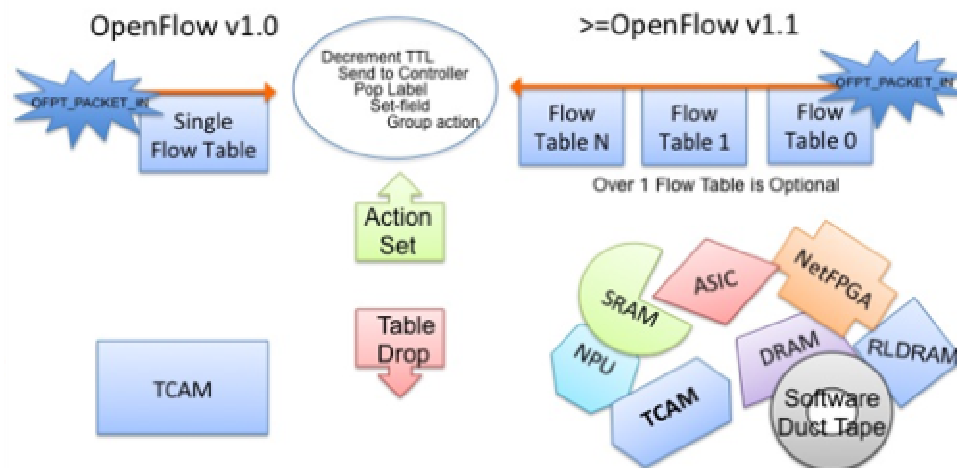


(<https://www.sdxcentral.com/wp-content/uploads/2012/06/image008.png>)

**Figure 4.** Example of TCAM compression to preserve TCAM space.

### Looking Ahead

Starting with OpenFlow v1.1 and beyond rather than having one flow table there is a pipeline made up of multiple flow tables. While this does add complexity to the standard it also allows for more flexibility in leveraging different hardware components rather than just raw TCAM to do associative array logic. Brocade has a slick implementation of protected and unprotected modes that can optimize for hybrid deployments or migration to balance the normal FIB and SDN tables. This is important for my personal [use cases \(https://www.sdxcentral.com/sdn-nfv-use-cases/\)](https://www.sdxcentral.com/sdn-nfv-use-cases/) in research and education networks, to allow researchers to have research networks consolidated on the same network substrate as traditional production services.



(<https://www.sdxcentral.com/wp-content/uploads/2012/06/image010.png>)

**Figure 5.** There will be new combinations of hardware and software for SDN lookups coming.

Nick McKeown put it quite eloquently in a meeting in February of 2010 in the weekly OpenFlow meeting notes. The topic multiple flow tables was being discussed. “We want to create a dynamic where we have a very good base set of vendor-agnostic instructions. On the other hand, we need to give room for switch/chip vendors to differentiate.”

This ability to distinguish hardware platforms may be the balancing act to squeeze performance out of the hardware market for the next few generations of network devices as the industry dips its toes into SDN while juggling normal forwarding tables, and even more complicated, interactions between the two pipelines. When choosing hardware for proof of future proof of concepts or reading new spec sheets with application towards SDN the flow tables and ACL like tuple matching will be relatively important when trying to pre-determine performance and scale. Software will be just as important on the southbound interface as it is on the north, but hopefully over the next few years, we can get back to being blissfully unaware of what is happening below the UI when implementing policy and architectures.

#### CONTRIBUTED ARTICLE DISCLAIMER

*Statements and opinions expressed in articles, reviews and other materials herein are those of the authors; not the editors and publishers.*

*While every care has been taken in the selection of this information and reasonable attempts are made to present up-to-date and accurate information, SDxCentral, LLC cannot guarantee that inaccuracies will not occur. SDxCentral will not be held responsible for any claim, loss, damage or inconvenience caused as a result of any information within this site, or any information accessed through this site.*

*The content of any third party web site which you link to from the SDxCentral site are entirely out of the control of SDxCentral, and you proceed at your own risk. These links are provided purely for your convenience. They do not imply SDxCentral's endorsement or association. The copyright and any other intellectual property right any third party content belongs to the author and/or other applicable third party.*

## About Brent Salisbury

Brent Salisbury is a Network Architect at the University of Kentucky. He is a CCIE 11972. He blogs at <http://networkstatic.net/> and can be reached at <https://twitter.com/networkstatic>



## Comments



**David Lenrow** says  
[July 2, 2012 at 9:00 am](#)

Greatpost Brent. You've made the totally abstract completely concrete for folks who aren't fluent in switch architecture and design. The world needs more of this kind of good, practical advice to enable folks to summon the courage to jump into SDN or at least dip a toe into the pool.

However I have to take issue with your last sentence. I don't think we've ever been able to implement policy without understanding everything about the low level plumbing. I guess you mean that previously we didn't need to understand how the protocols get implemented on the hardware, but I don't even want to know what protocols are being used. My dream is for SDN to take us way up the abstraction ladder to where all we need to think about is how the network should behave, not how to hook the plumbing together. You're probably going to suggest we take this one step at a time, and that's absolutely the only way we'll get there. But I don't want to lose sight of the summit or we may lose the trail

[Log in to Reply](#)



**Brent Salisbury** says  
[July 2, 2012 at 10:13 am](#)

Hi David. You nailed it TCAM is thrown around a lot these days and until the past year, I was quite delinquent on some of the hardware nuances until I have been forced to understand. Since the community is debating evolving the construct, it seems to be requiring some re-visiting of the whole stack.



I think that abstraction needs to be the end game to start stacking meaningful, Northbound applications on top of one another to integrate network resources as just another resource floating in the pool. It certainly makes the hardcore cringe and I would be lying if I said otherwise at times. There is not much more I love in life, than taking a pile of autonomous gear that have zero awareness of one another and making them sing. The problem is doing that to scale efficiently. Urs Hoelzle analysis at the ONF had a great statement when he pointed out that most infrastructure elements, compute, storage etc become cheaper when we grow, but networking seems to keep a linear or even worse in some cases exponential climb in cost.

Could not agree with you more in keeping the eye on the end game. I am really getting ancy that we aren't discussing the Northbound API more, but to your point it is probably being tackled in a sequential layered approach. As a VC where do I put R D money without an articulate roadmap.

We are stuck thinking about the details for a while I reckon, but I imagine it will be more hardware focused than speccing out an x86 server today but much much less than the details to try and figure specs out today (opening the door for a good bashing). I think commoditization in general is arguably leading the horse to water, a large slow lumbering horse.

Thanks and nice to meet you, great insight



**Art Fewell** says

July 5, 2012 at 12:13 pm

Really excellent post Brent

[Log in to Reply](#)



**Brent Salisbury** says

July 5, 2012 at 5:18 pm

Thanks Art appreciate it pal

[Log in to Reply](#)



**Jay Flora** says

March 17, 2014 at 4:42 pm

Excellent post Brent

[Log in to Reply](#)

## Leave a Reply

You must be [logged in](#) to post a comment.

