T8: SDN in practice: CISCO's Application Policy Infrastructure

Critique

Birkner, Rüdiger

1. The paper "Application Centric Infrastructure Overview" is a pure marketing paper using a lot of nice words describing how good there architecture is while omitting the details (e.g. support emerging industry demands, exceptional in the industry, ACI is built for the needs of both today's workloads and tomorrow's changing demands)
2. It is stated that in case of a controller failure, no loss in existing datacenter functionality will result. They do not
3. The APIC provides a policy-driven configuration and can be configured via several interfaces (RESTful, CLI, GUI). It is not mentioned how the configuration works. In my opinion, it is important to have a simple interface with high usability. It is not clear if they have really implemented what they are writing about or whether those are just some vague ideas.
4. They do not mention that their controller is based on OpenDayLight.

Lee, Tae-Ho

1. The white papers only describe their product without providing any evidence.
2. It is very confusing to read the paper. The paper would be better off if it has a high-level example with workflows that show how various components that they describe in the paper come into the picture.

Chothia, Zaheer

1. SDN is an overloaded term, but if we interpret it as the decoupling of control and data plane then Cisco's ACI is a realisation of such an architecture. The network fabric is designed in such a way that the policy controller remains entirely outside the data path.
2. The article mentions how this enables high performance (but gives no experimental support) and fails to mention this also reduces flexibility. Whilst OpenFlow can support reactive decisions, in ACI the controller never observes the data and is constrained to just pre-emptive policies. Example: an in-network firewall seems possible but not a load balancer. Despite how Cisco present their new design, I view ACI as a distributed database with a glorified management interface bolted on.
3. What differentiates this proposal, then, is the focus on provisioning and having a repository describing the infrastructure (state, events, faults). They realise this by having built-in support for network overlays and a mechanism to define application connectivity directly, rather than via low-level primitives ("subnets, VLAN stitching, and ACLs").
4. Even in this sense, though, ACI falls short. A major selling point is the separation of logical model and concrete realisation which they describe at length ("Promise Theory"). The whitepaper does not contain any compelling examples and looking through 'acitoolkit' it really is just imperative scripting rather than a new declarative language: https://github.com/datacenter/acitoolkit/blob/master/samples/aci_demo_contract.py
5. It is laudable of Cisco to offer an alternative to the CLI but the claim of open APIs is only true in the narrow technical sense. The innards of the implementation are not exposed;
you are limited to whatever the designers have conceived and still unable to extend at will. This goes largely against the spirit of the early OpenFlow idea, namely to enable innovation.

6. I should probably stop at this point but still have many unanswered questions. Here are two as examples:
   a) Unclear to what extent they mean "logical separation" (e.g. between dev/prod). Is this purely from the management standpoint or do they also consider performance isolation or failures?
   b) From what I understand, the management interface runs in-band. How do the monitoring or topology subsystems work when there is a network partition?

Defense

Miladinovic, Djordje

CISCO ACI is a network framework, managed by SDN-like central controller and in the paper we can find many positive points.

1. ACI is a framework which nicely integrates all the desired properties that we want to see in our datacenter: Security and isolation of tenants, traffic monitoring and observing, reliability with redundancy of controller, custom policies enforcement...
2. ACI has a standardized northbound interface and a variety of ways to talk to user applications. This makes it easily deployable and used.
3. Has an enhanced scalability and reliability by using some high-level concepts like "Sharding".

Pappas, Chris

Cisco APIC&ACI white paper sounds like a complicated way to describe basic SDN concepts, although there are some differences with the common realization of SDN.

1. The proposal focuses especially on realizing network policies and specifically application oriented policies.
2. A leaf-spine fabric topology is defined in order to achieve robustness. Traditional SDN proposals do not assume any underlying topology.
3. A different management model is proposed for the control plane that is hardware independent and defines logical configuration policy abstractions of the policy state.

Shinde, Pravin

**** Strengths

1. Goal/Aim:
   a) Data-model based declarative provisioning
   b) management automation, programmable policies to support dynamic workload
   c) Network configuration and logical topology should be based on the application configuration and not the other way round.
2. The grouping is done with endpoints of the granularity of NICs, VNICs or higher
3. Multiple endpoints can be grouped together to define higher-level/aggregated policies
4. The policy enforcement is done by "policy elements" which are monitoring the current configuration to ensure that policies are valid.
5. The APICs can discover each other and the topologies by using variant of LLDP leading to zero configuration (plug and play) setup.

**** Criticism

1. As per my understanding, the APICs are doing network management at the granularity of NICs and VNICs, but not at the granularity of flows.
2. The paper stresses that APICs are not in data path, and hence better than open-flow, but I am not sure if they are doing the same task.
3. It is not clear to me how long does the APIC convergence takes place with the custom LLDP protocol. I guess this is not very critical as the APIC does not operate at flow level, hence does not need very high accuracy with flows.
4. It is not clear how policy enforcement with policy element happens. Is it just reprogramming the switches whenever an issue is detected? The paper also mention that actual reality may drift away from the the policies, but implications of such a drift from correctness and performance point of view are not clear.
5. It is not clear what happens when there are conflicting policies. Who detects and resolves such conflicts, and how?

Schmid, Stefan

1. Cisco presents their approach of an SDN implementation. Their system consists of a controller (or controllers) (Application Policy Infrastructure Controller, APIC), spine switches and leaf switches. The spine switches are not interconnected but connect to all leaf switches. Traffic is either forwarded locally or over one spine switch to another leaf switch. The APIC can be accessed via API through XML and JSON and also provides a command-line interface and GUI. The APIC can be used to enforce policies or manage the network but it is not part of the data path. This means even if the APIC fails, the switches will still forward packets according to the last applied policy.
2. The papers read like an advertisement for their system. They use a lot of superlatives and only say what their system can do but not how it is achieved. They also do not provide any evaluation or measurement results. It seems that the system will do its job for customers looking for an out of the box solution, but as soon as more customized network applications are needed, it will reach its limitations soon.

Yu, Xinyuan

1. APIC is completely removed from the data path. It is separated from data control. In this way, application connectivity and policy at the core. The fabric can still forward traffic even when communication with the APIC is lost. This makes the system more stable.
2. Sharding is used in APIC. This makes replication across multiple severs easy.
3. APICs a scalable and flexible system. APIC cluster are fully extensible to computing and storage managemant.