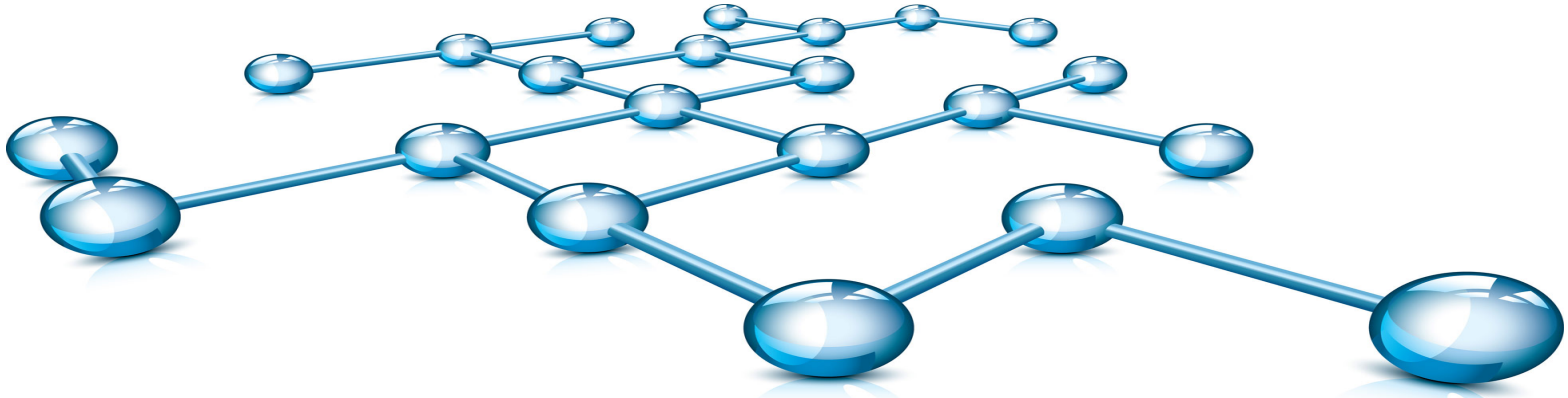


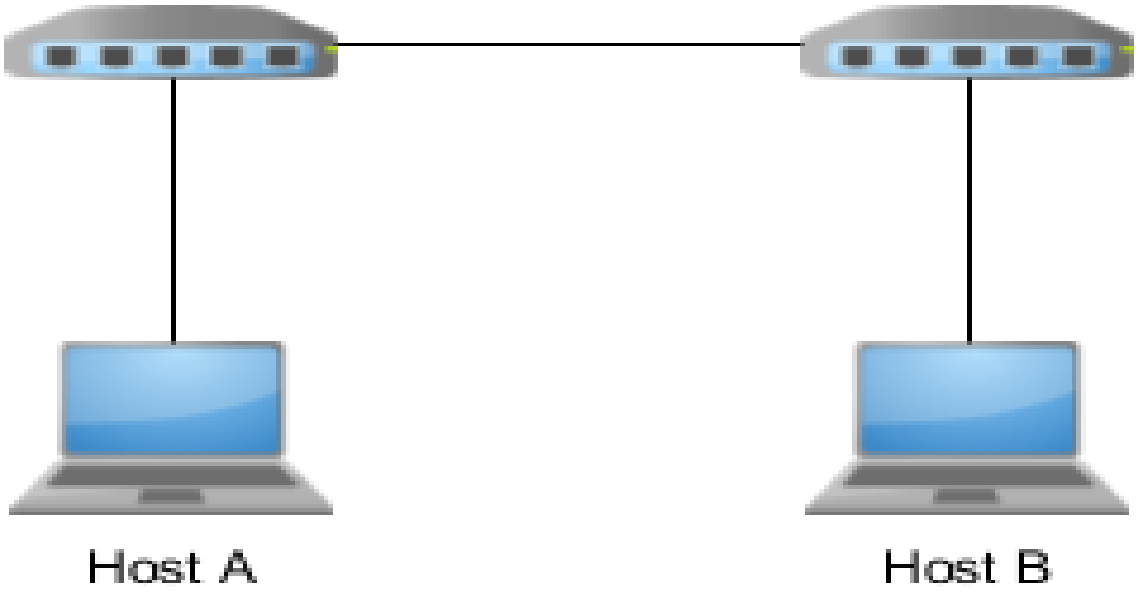
Software-Defined Networking

IBM CorkCon Mar 10th 2016

Dylan Smyth
Nimbus Centre for Embedded Systems Research

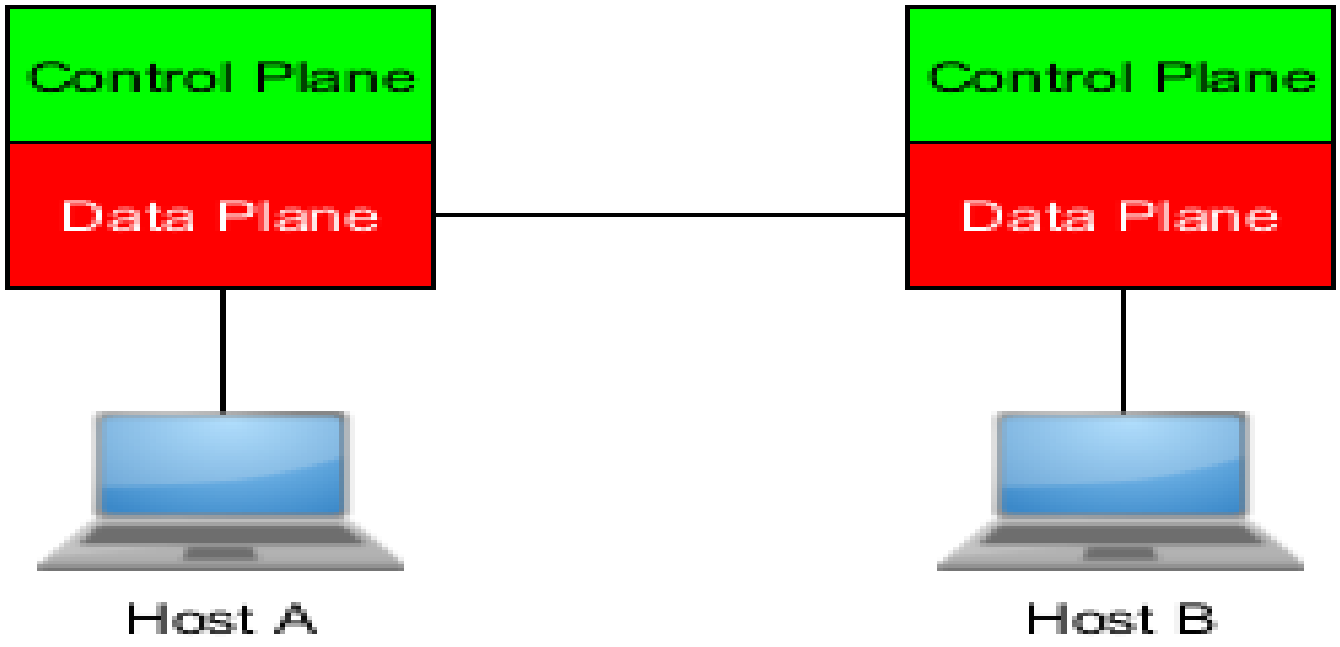
Conventional Networking

- Each device is configured separately
- Each devices makes forwarding decisions



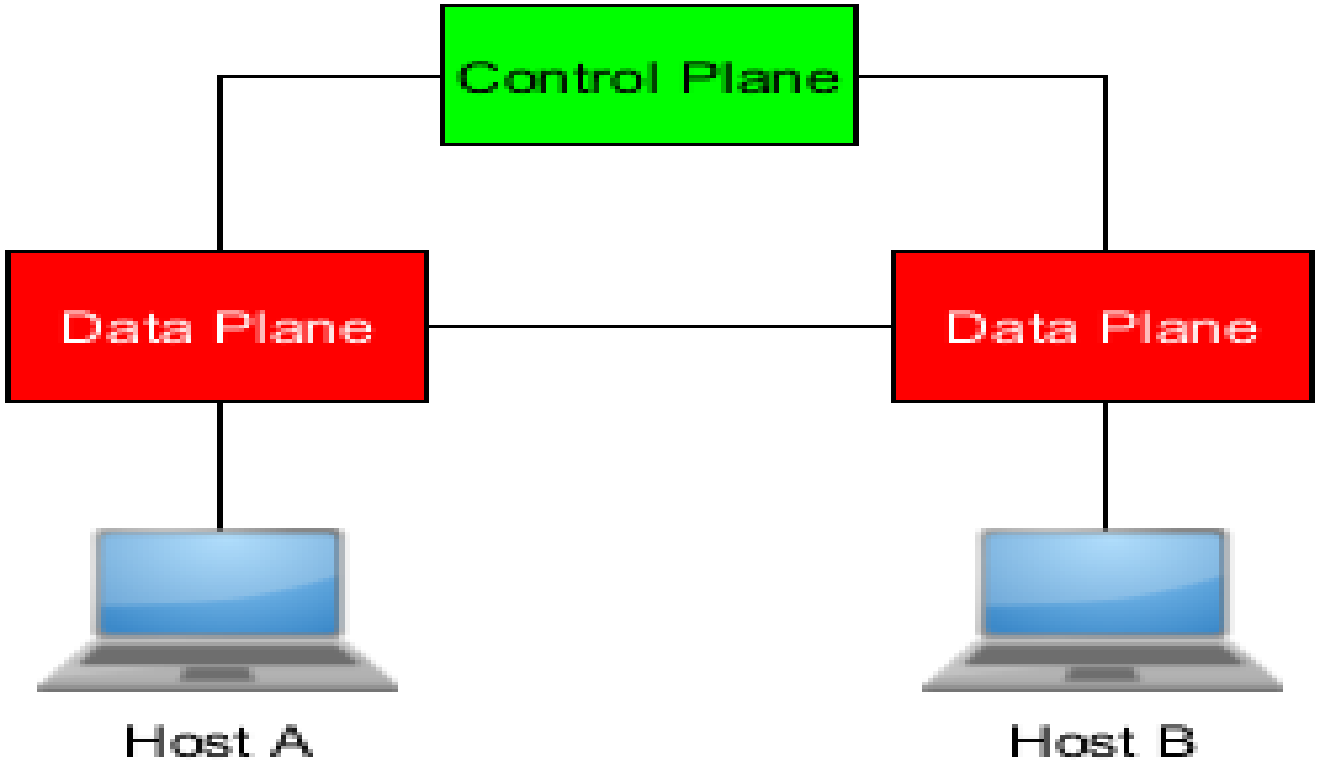
Conventional Networking

- A Control and Data plane exists on each device



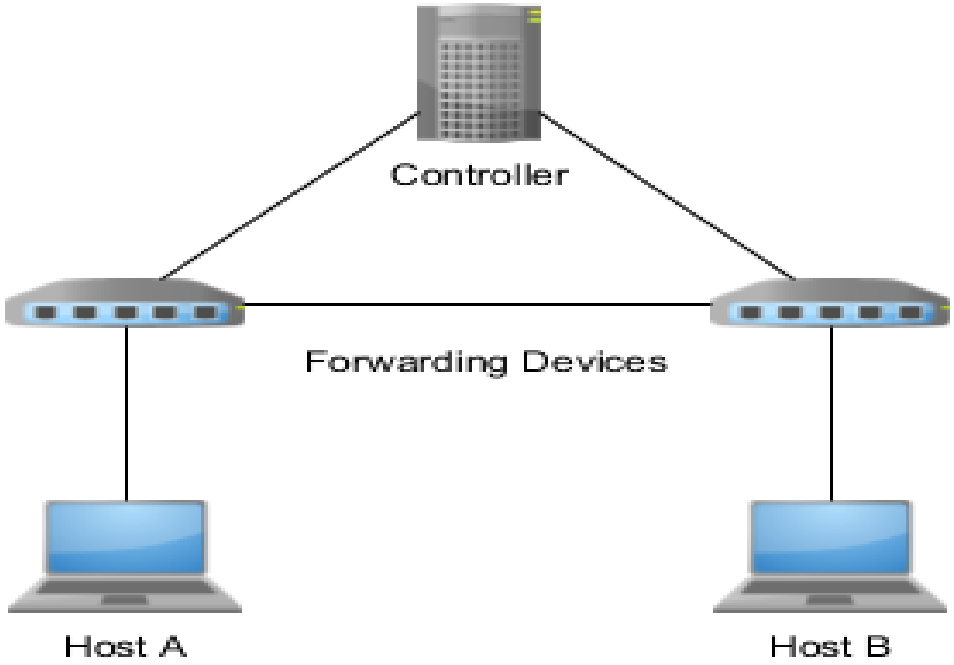
Software-Defined Networking

- Control and Data planes are decoupled



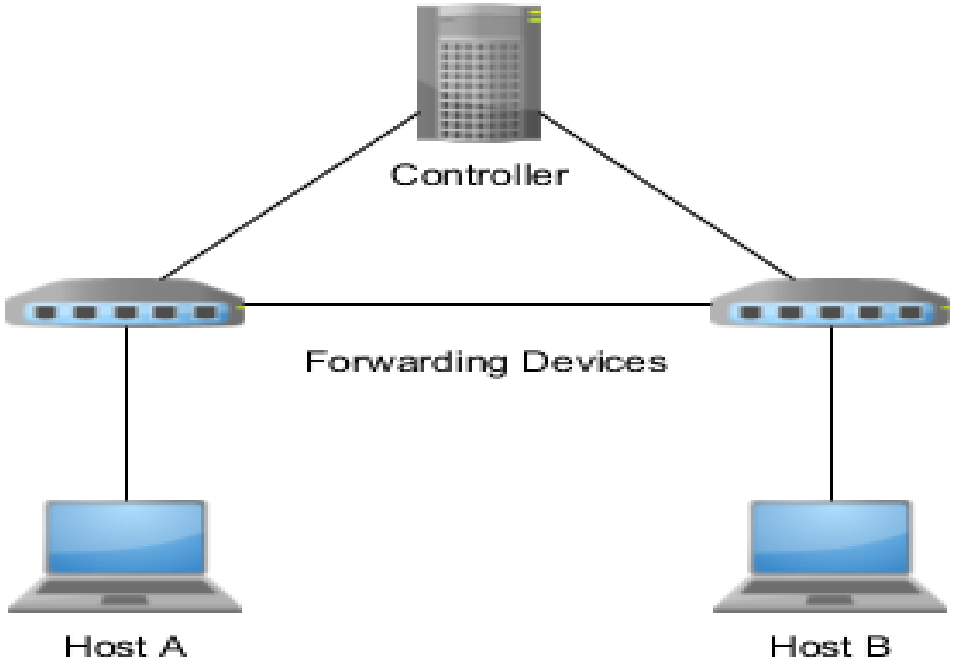
Software-Defined Networking

- Decision making is centralized to a “Controller”
- Data forwarding is left to the “Forwarding Devices”



Software-Defined Networking

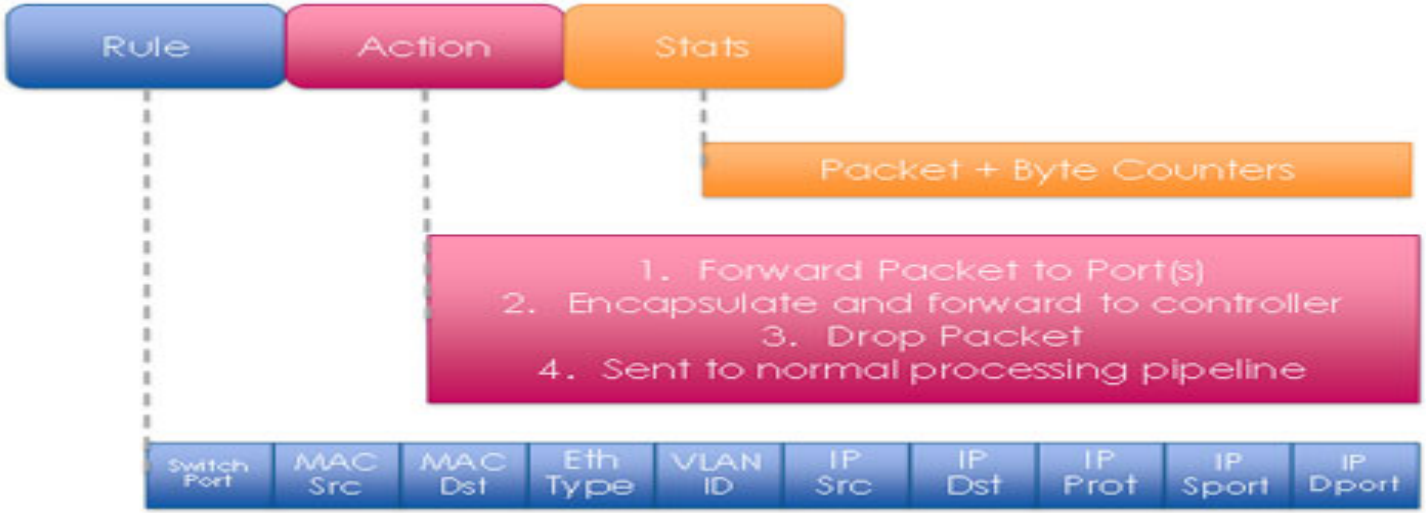
- Controller has a centralized view of the network topology



Software-Defined Networking

- Forwarding is defined by 'flow entries' in 'flow tables'

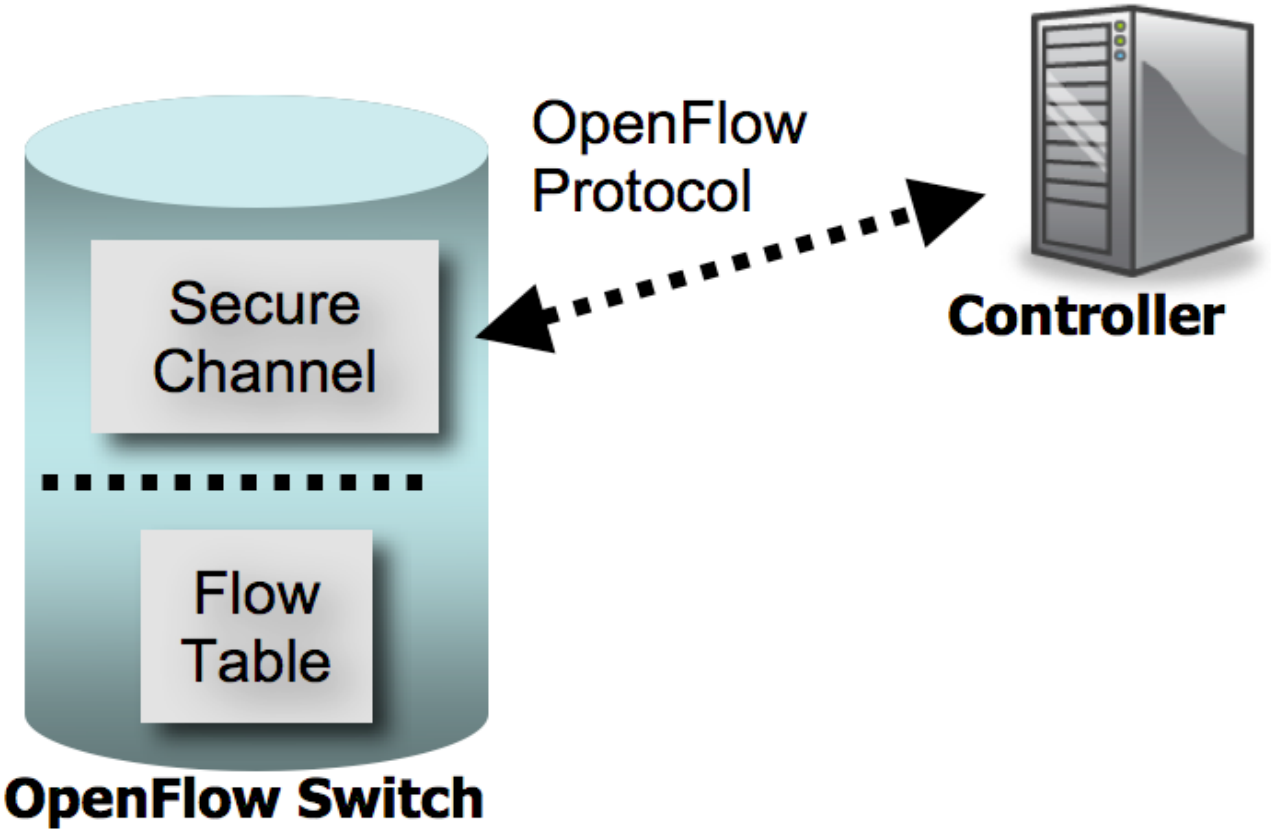
- Flow entries contain:
  - Packet matching information (Port, src/dst IP, etc.)
  - Actions to perform (Forward, drop, etc.)
  - Statistical Information for flows



[1]

Software-Defined Networking

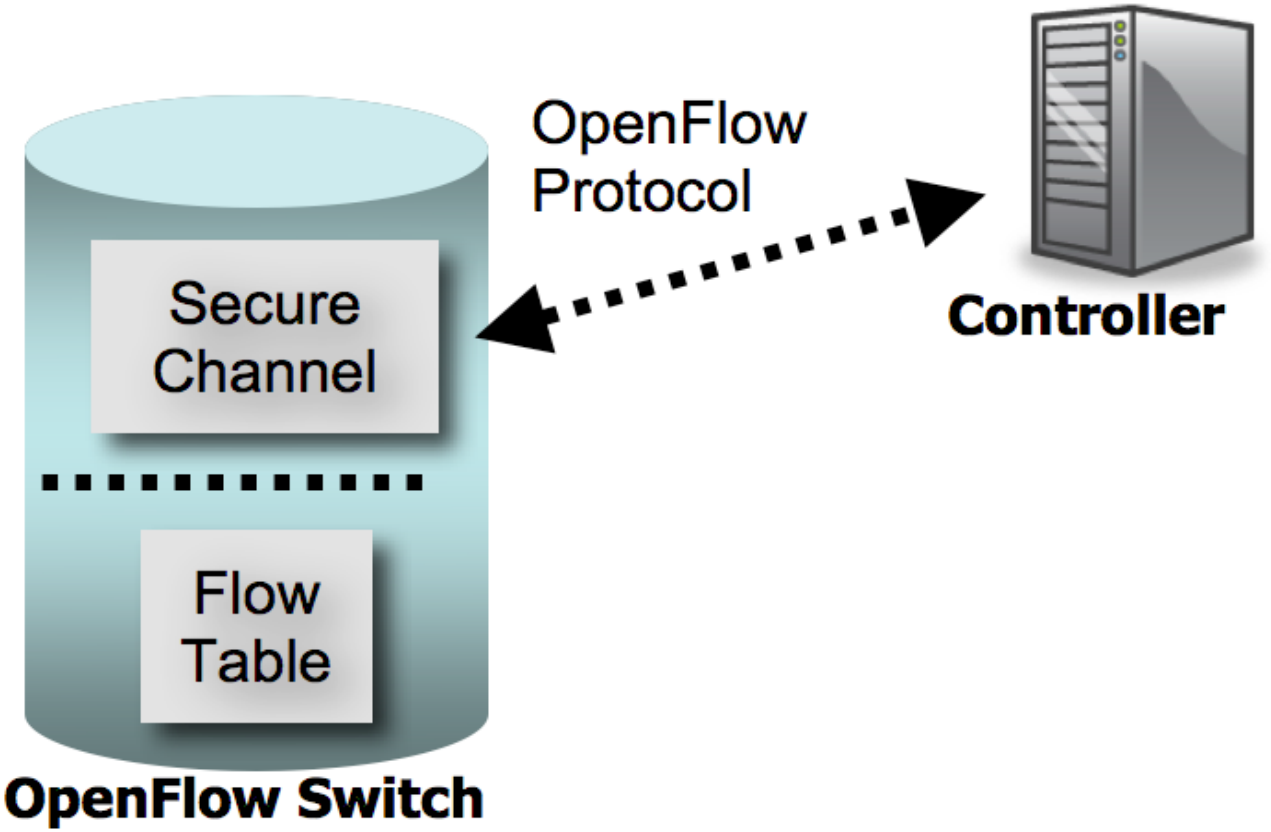
- Controllers insert, update, and delete flow table entries



[2]

Software-Defined Networking

- Communicates via a “Southbound” API



[2]

OpenFlow

- A control protocol used for communication between the Data Plane and the Control Plane.

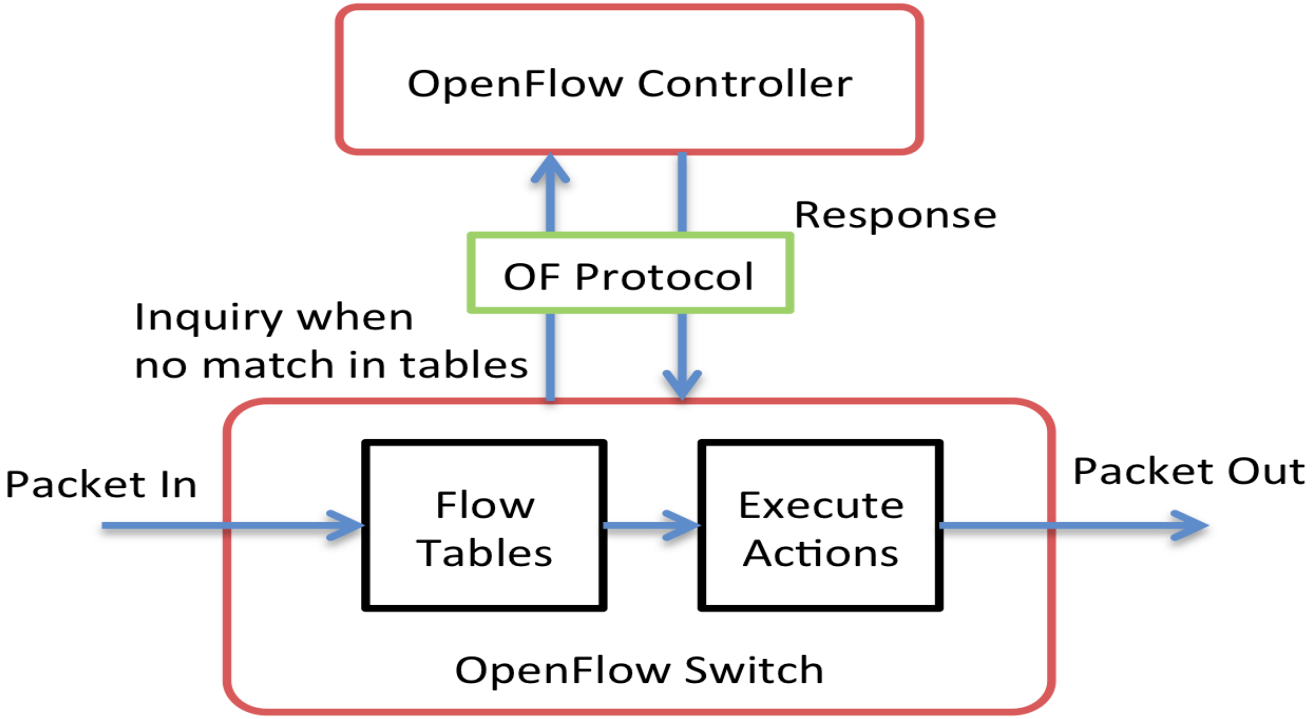
[3]

Software-Defined Networking

- SDN networks can be “Proactive” or “Reactive”
- “Table Miss” - When a packet does not match any flow table entries

Software-Defined Networking

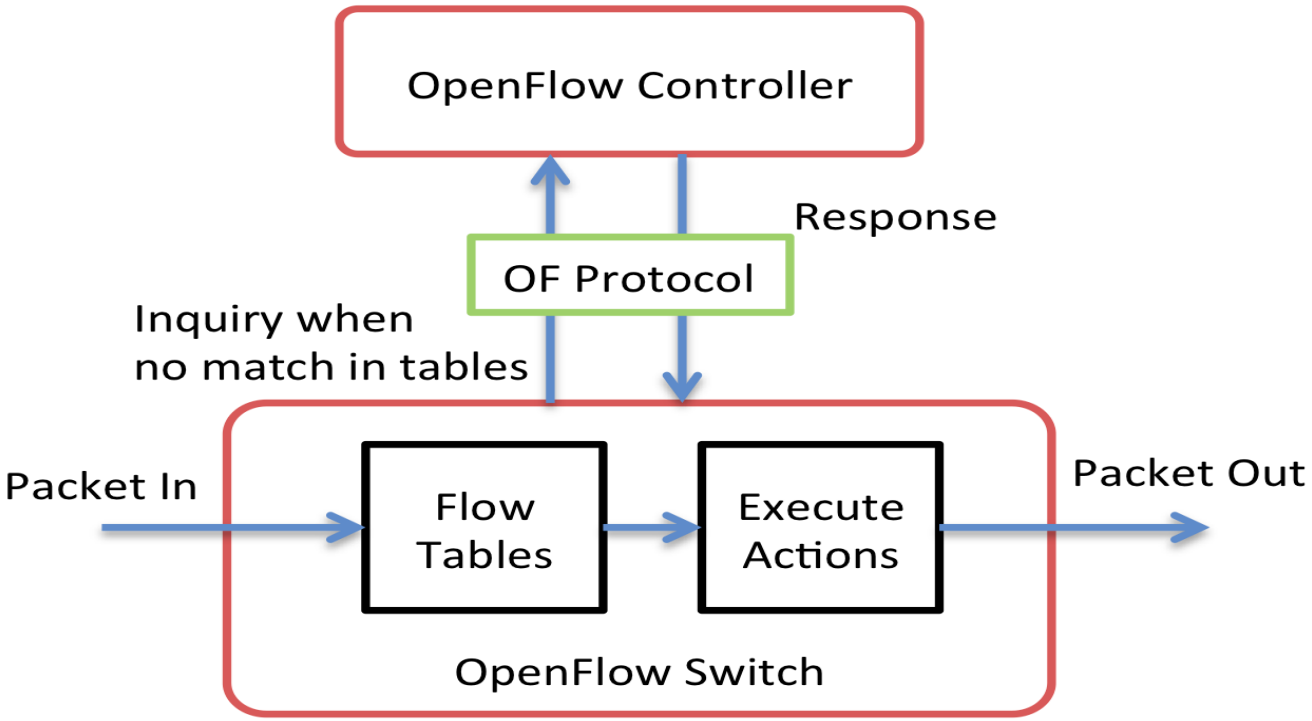
- When a Table Miss occurs:
  - Switch sends data to controller.
  - Controller determines what to do
  - Updates switches in the network accordingly



[4]

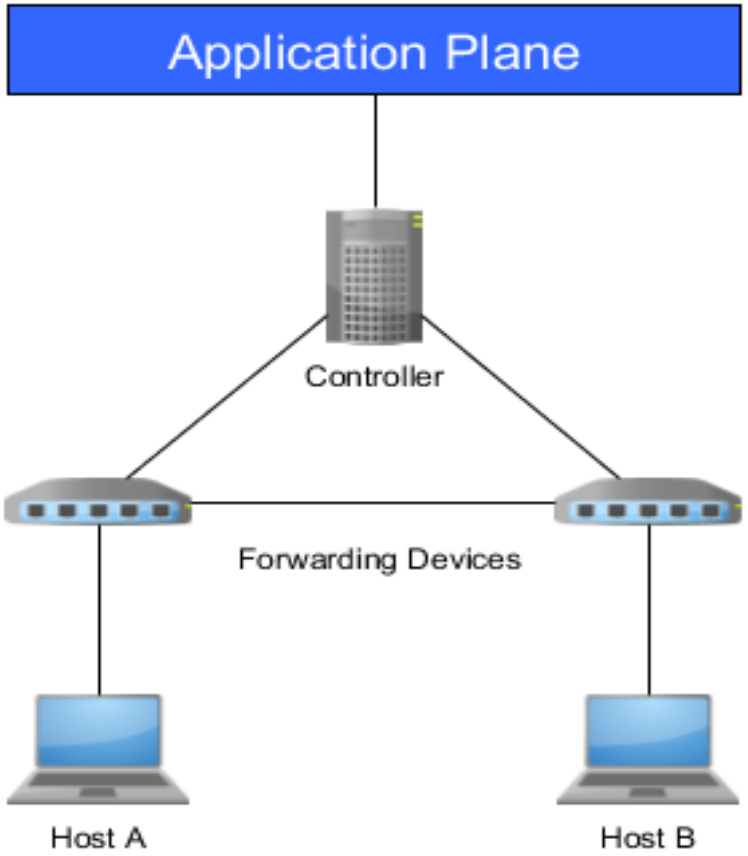
Software-Defined Networking

- Controllers decision is influenced by the requirements of Applications



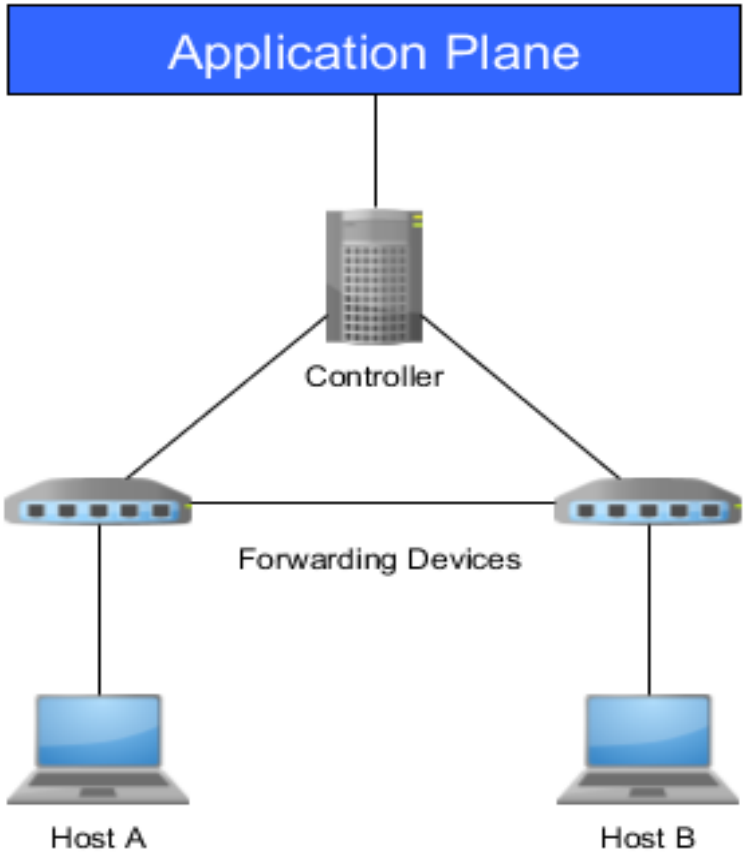
[4]

Software-Defined Networking



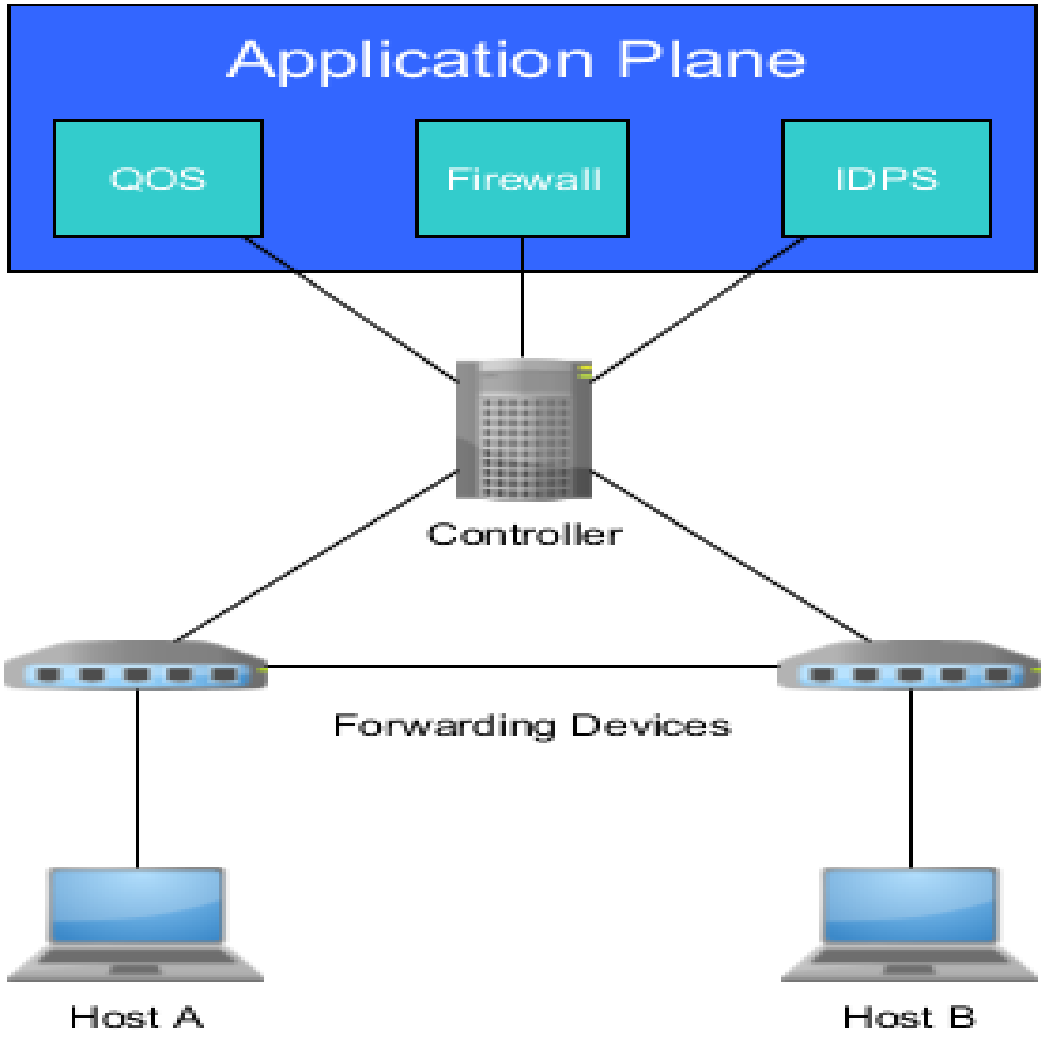
- Applications communicate requirements to the controller

Software-Defined Networking



- Communicates via a “Northbound” API (e.g. REST)

Software-Defined Networking



Security Opportunities

- Security can be implemented as Applications
- Security requirements can be maintained centrally
- Flow entries can be inserted to block connections or assist monitoring

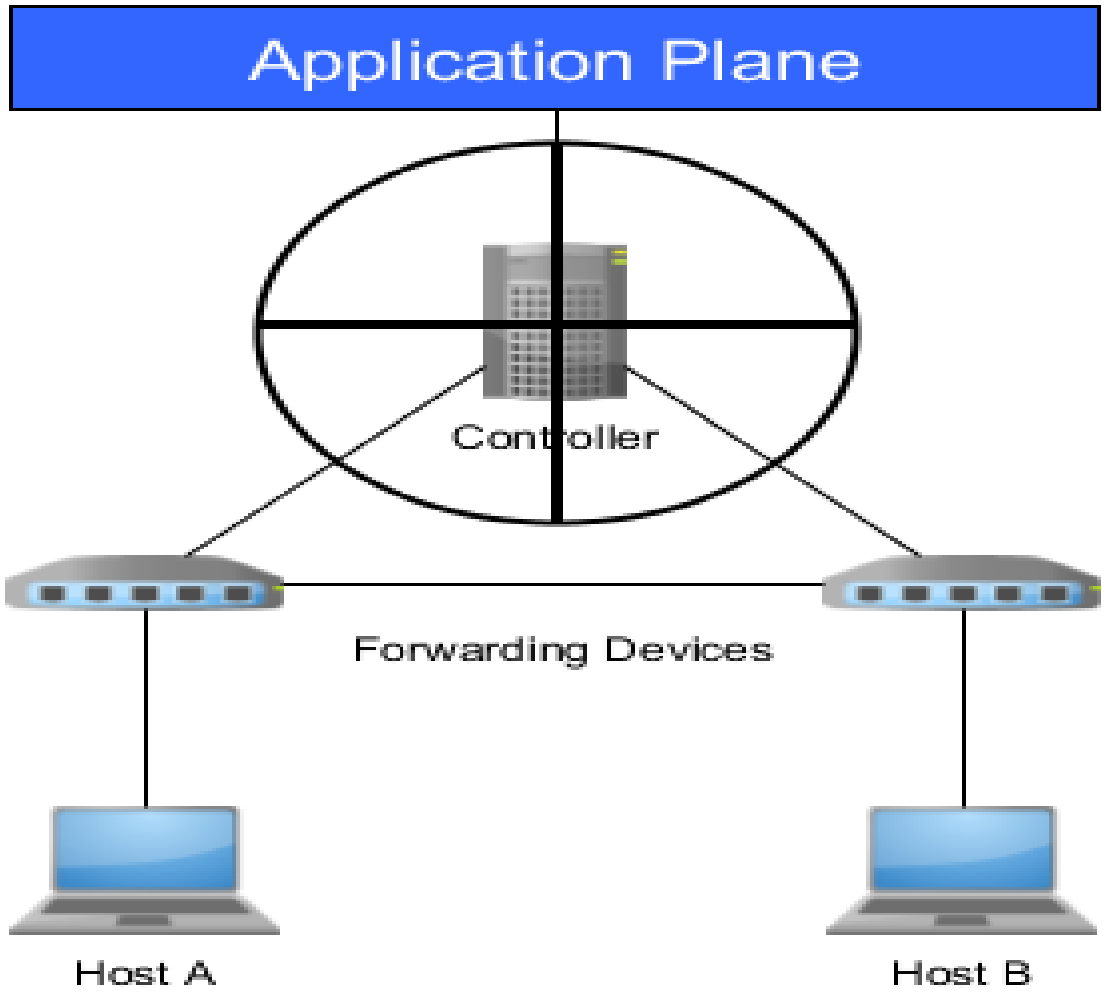
Software-Defined Networking

Security Opportunities

- Detection mechanisms possible for DOS, Worm propagation, etc.

- Ability to actively update rules in the network presents opportunities (Self Defending Network)

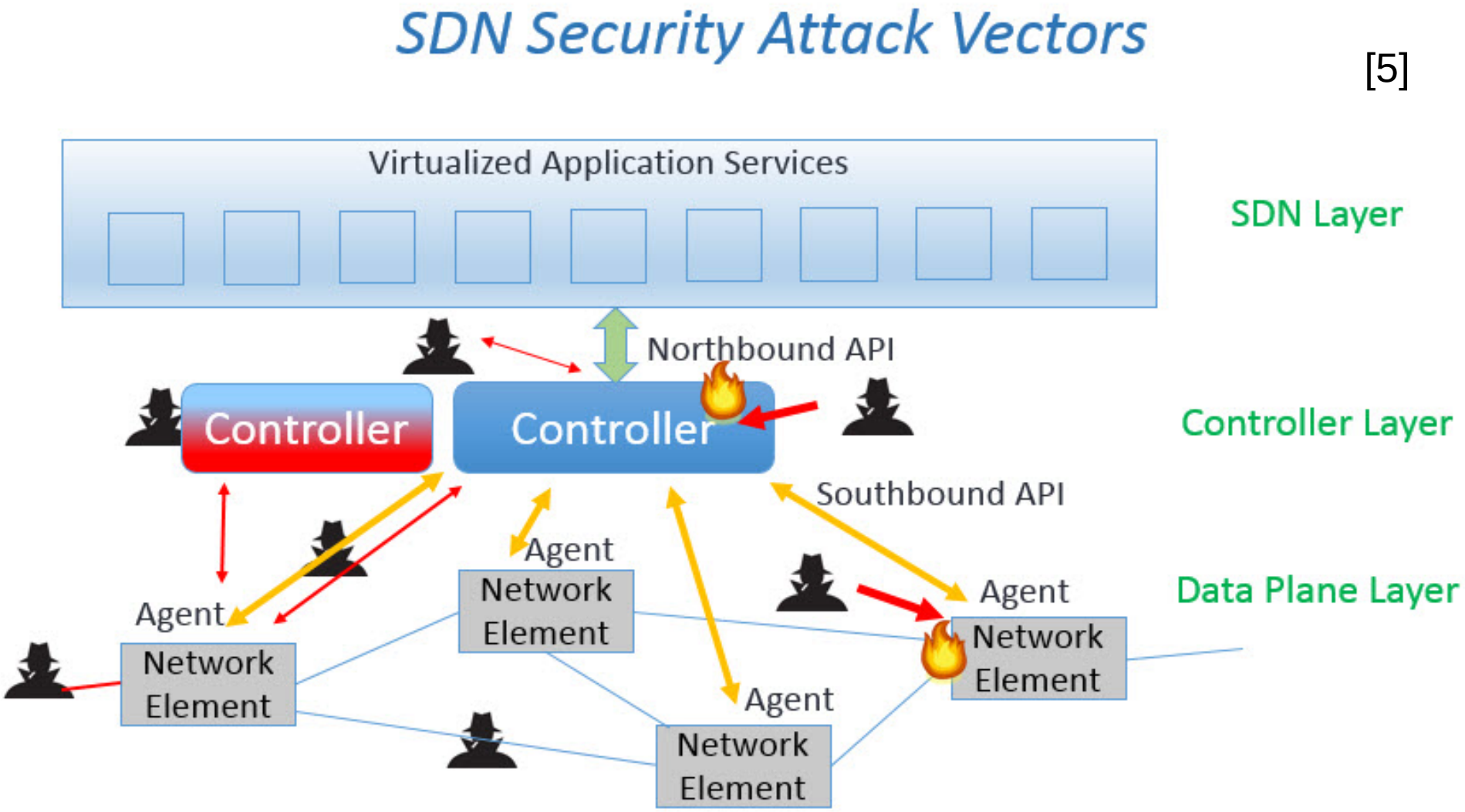
Security Challenges



Security Challenges

- Controller is a single point of failure
- TLS usage is not required for OpenFlow
- Malicious/Vulnerable Applications
- Flows could be engineered to bypass access control
- Side-Channel attacks to determine flow rules

Security Challenges



[5]

Questions

Image Sources

[1]
https://www.sdxcentral.com/wp-content/uploads/2013/08/manipulated-openflow-switch.jpg

[2]
https://tele-4642-group-4.wikispaces.com/file/view/openflow%201.jpg/512890232/openflow%201.jpg

[3]
https://www.opennetworking.org

[4]
https://s3f.iti.illinois.edu/usrman/openflow.html

[5]
http://core0.staticworld.net/images/article/2014/10/sdn-sec-1d-100527554-large.idge.jpg