Software Defined Networking

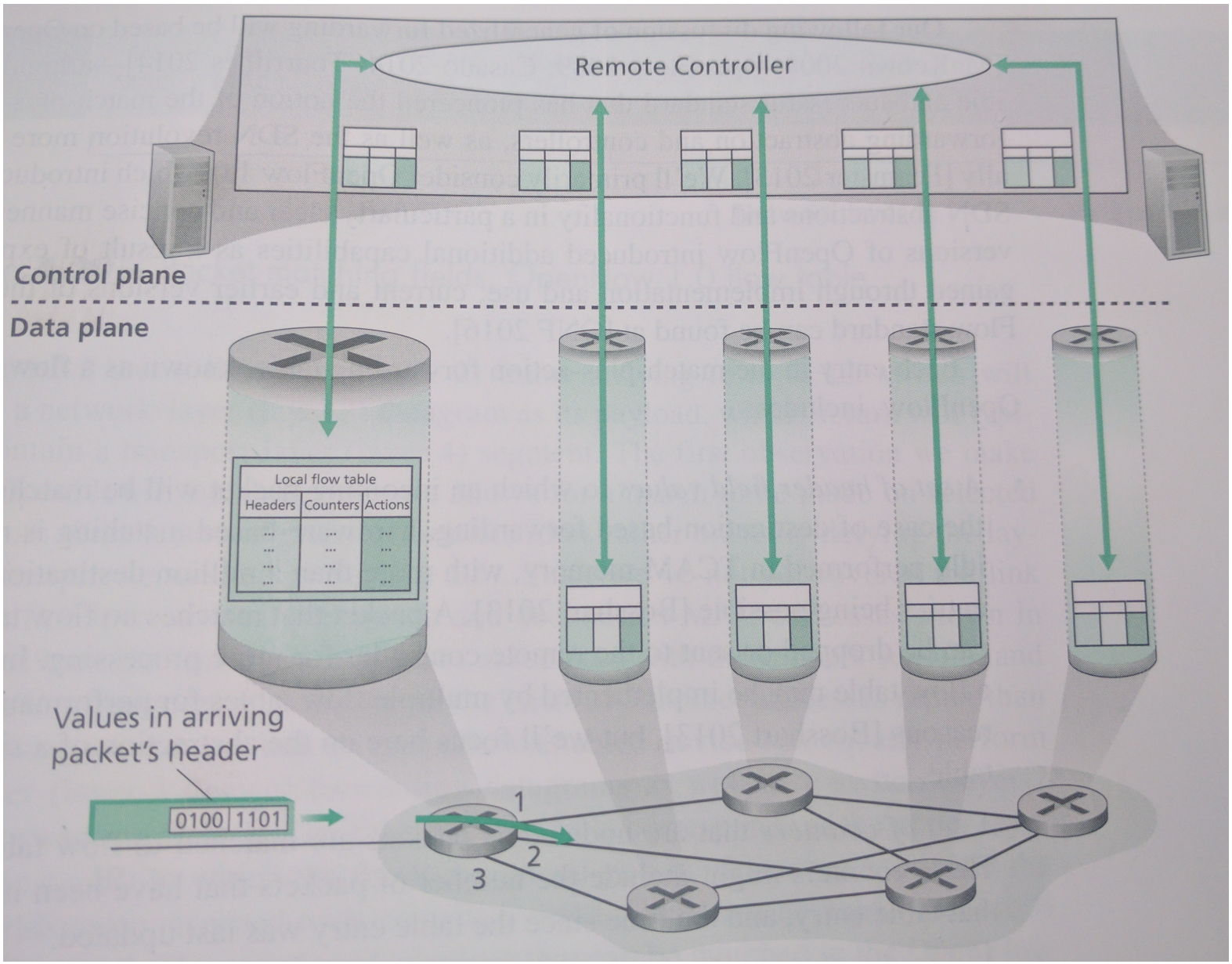
Daniel Zappala

CS 460 Computer Networking
Brigham Young University

Proliferation of Middleboxes

- a router that manipulatees traffic rather than just forwarding it
- NAT
  - rewrite IP address and TCP port fields to allow private addressing
- firewall
  - inspect headers and data (deep packet inspection) to block unwanted traffic
- load balancer
  - inspect headers and reroute packets to a different server to balance load in a cluster
- and more! See RFC 3234
- current networks have a mix of routers (network layer), switches (link layer) and middleboxes (both layers), each with specialized hardware, software, management

Software Defined Networking



Software Defined Networking

- generalize the packet forwarding architecture: match-plus-action
  - previously match only on destination address, action is only forward on an output port
  - now, match on any fields in link layer, network layer, transport layer headers
  - now, action can be forwarding, load balancing, rewrite (NAT), block (firewall), inspect (DPI), etc.
- control plane is logically centralized
- data plane uses a flow table
- pioneering work with OpenFlow

Key characteristics

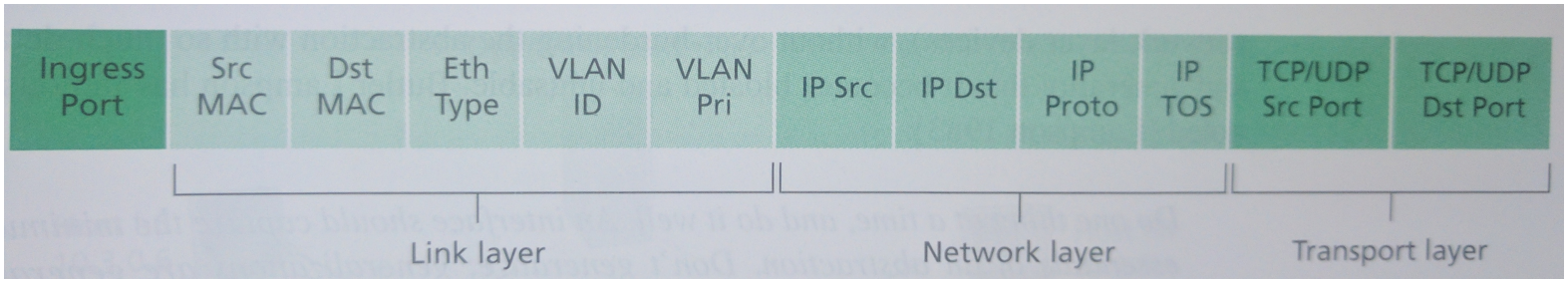
- *flow-based forwarding*: can use any of the fields in any header of any layer
- *separation of data plane and control plane*: data plane packet switches do match-plus-action forwarding, control plane manages the switch flow tables
- *network control functions*: control software runs on machines separate from the packet switches
- *programmable network*: can program the packet switches to execute many different functions: forwarding, load balancing, firewalling, etc.

Data Plane

Flow table

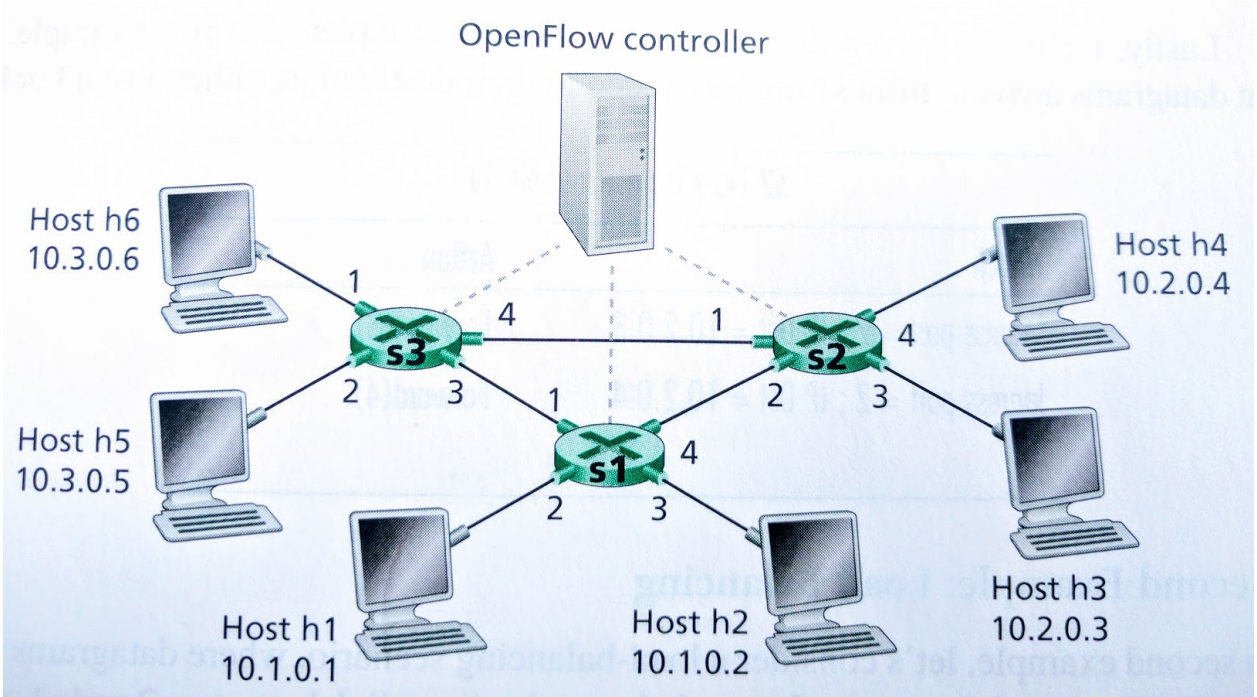
- header field values: match incoming packets
- counters: count packets that have been matched
- actions: actions based on result of match
- essentially a programmable packet switch
- can be implemented efficiently with multiple flow tables, in hardware

Match-Plus-Action



- match
  - OpenFlow 1.0 packet matching fields cover three layers
  - newest OpenFlow spec provides 41 values for matching
- action
  - forwarding
  - dropping
  - modify field
- can act as a switch or a router or a middlebox

OpenFlow Examples: Simple Forwarding

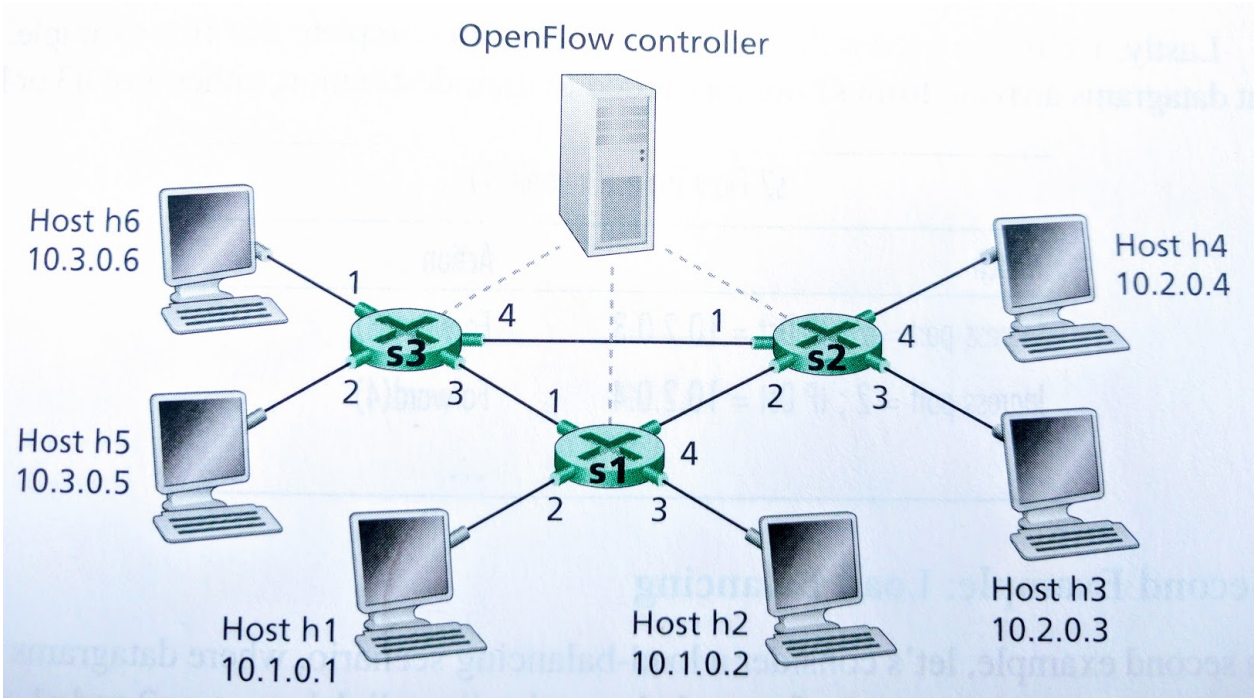


s1 Flow Table

Match	Action
Ingress Port = 1; IP Src = 10.3.*.*; IP Dst = 10.2.*.*	Forward(4)
...	

Will need similar tables for each packet switch

OpenFlow Examples: Load Balancing

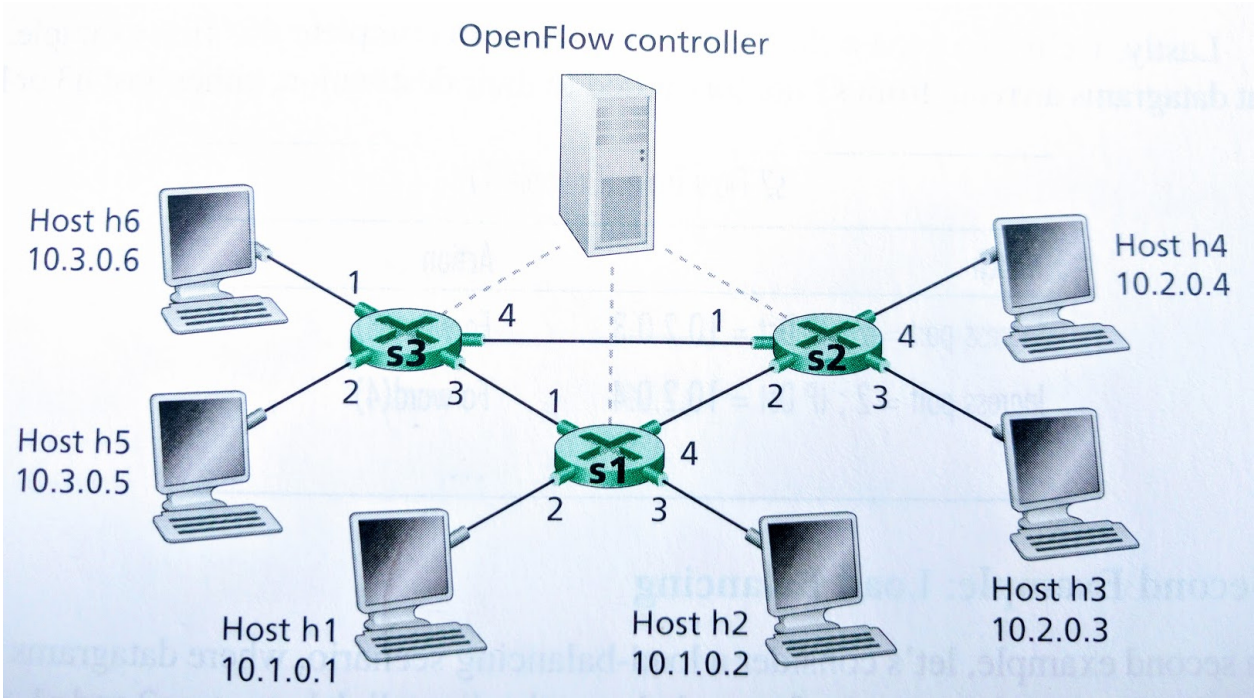


s2 Flow Table

Match	Action
Ingress Port = 3; IP Dst = 10.1.*.*	Forward(2)
Ingress Port = 4; IP Dst = 10.1.*.*	Forward(1)
...	

Will need additional tables for each packet switch

OpenFlow Examples: Firewalling



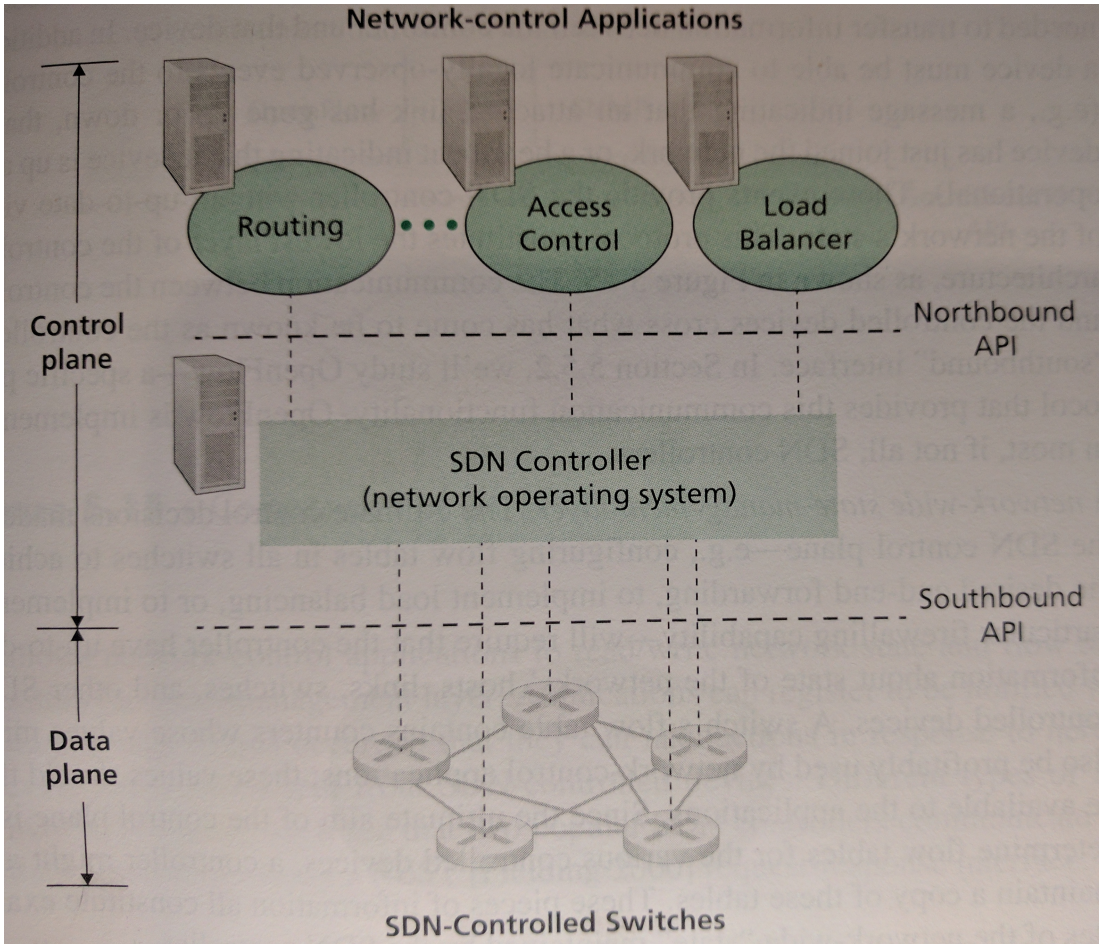
s2 Flow Table

Match	Action
IP Src = 10.3.*.* IP Dst = 10.2.0.3	Forward(3)
IP Src = 10.3.*.* IP Dst = 10.2.0.4	Forward(4)
...	

In absence of other entries, no other traffic forwarded

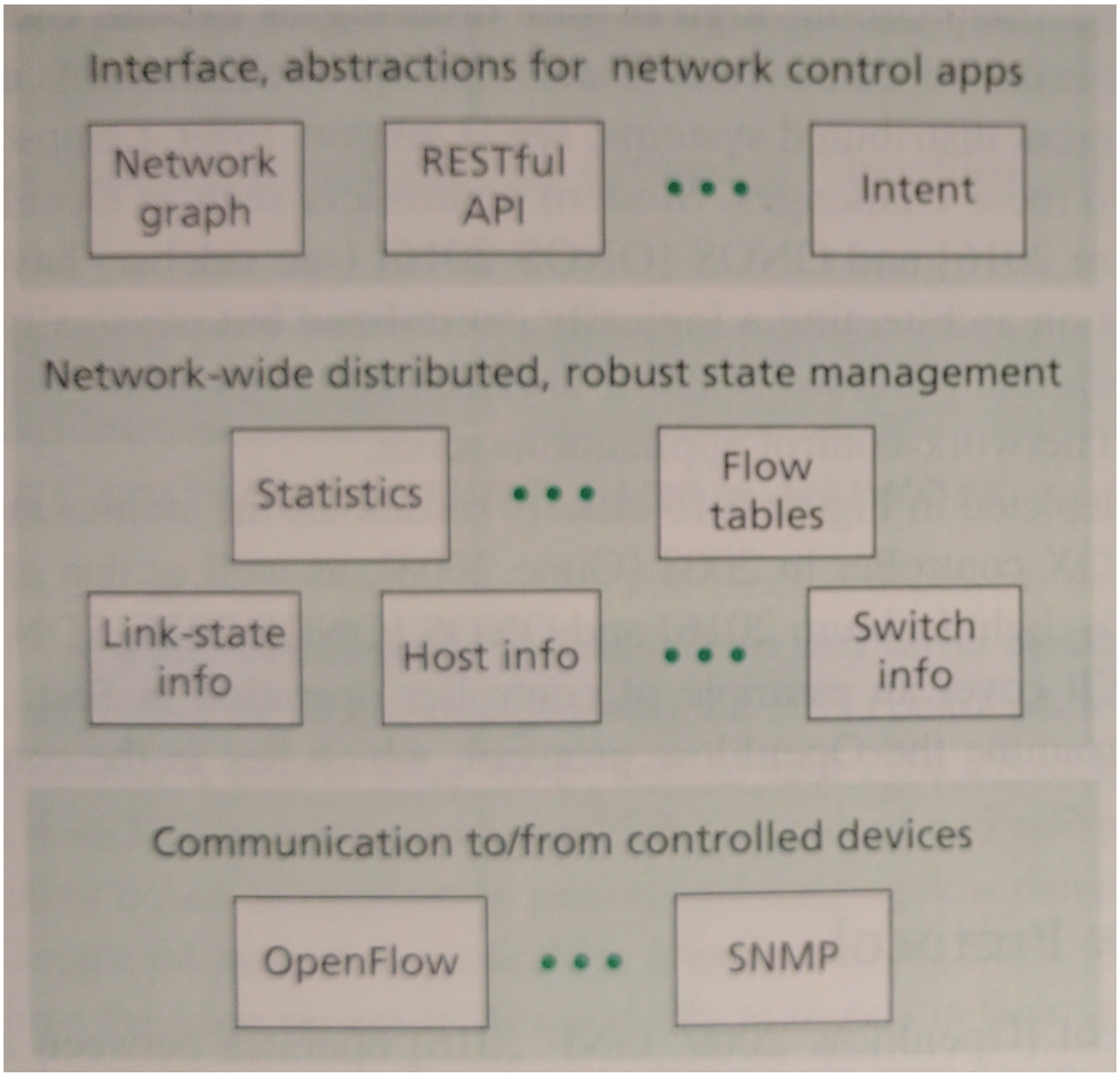
Control Plane

SDN Architecture



- controller maintains state of links, switches, hosts
- network control applications program network functions

SDN Controller



- logically centralized, but physically distributed among a set of

OpenFlow Protocol

- messages from the controller to a switch
  - *configuration*: query and set paramters
  - *modify-state*: add/delete/modify nentries in flow table
  - *read-state*: collect statistics and and counters
  - *send-packet*: send a packet on a specified port
- messages from a switch to the controller
  - *flow-removed*: flow table entry removed, due to timeout or *modify-state* message
  - *port-status*: change in port status (e.g. up/down)
  - *packet-in*: send packet that doesn't match any flow table entry to controller for processing
- and more...

Link State Change Example