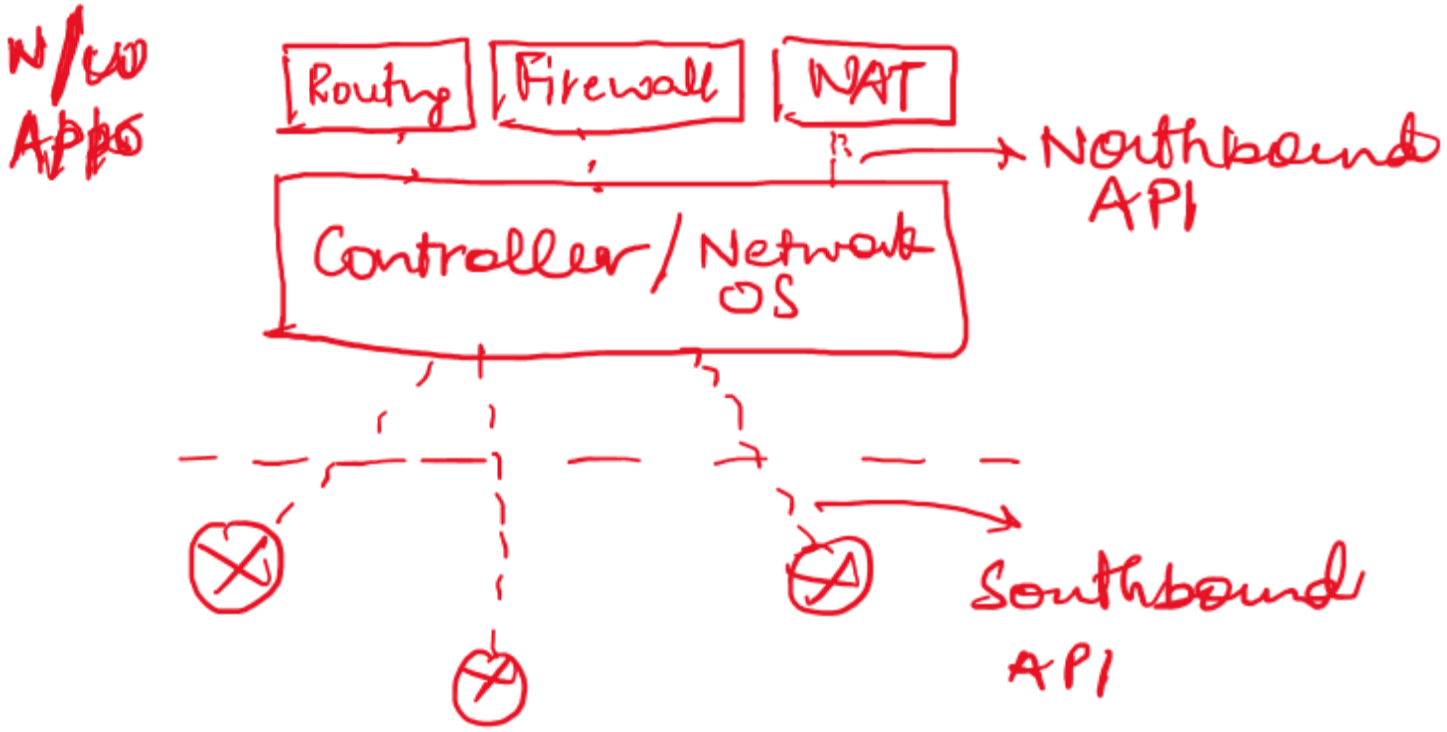
Computer Networks
COL 334/672

Software-defined Networking

Sem 1, 2025-26

Recap



OpenFlow (Southbound API)

① Generalized data plane abstraction

MATCH, ACTION

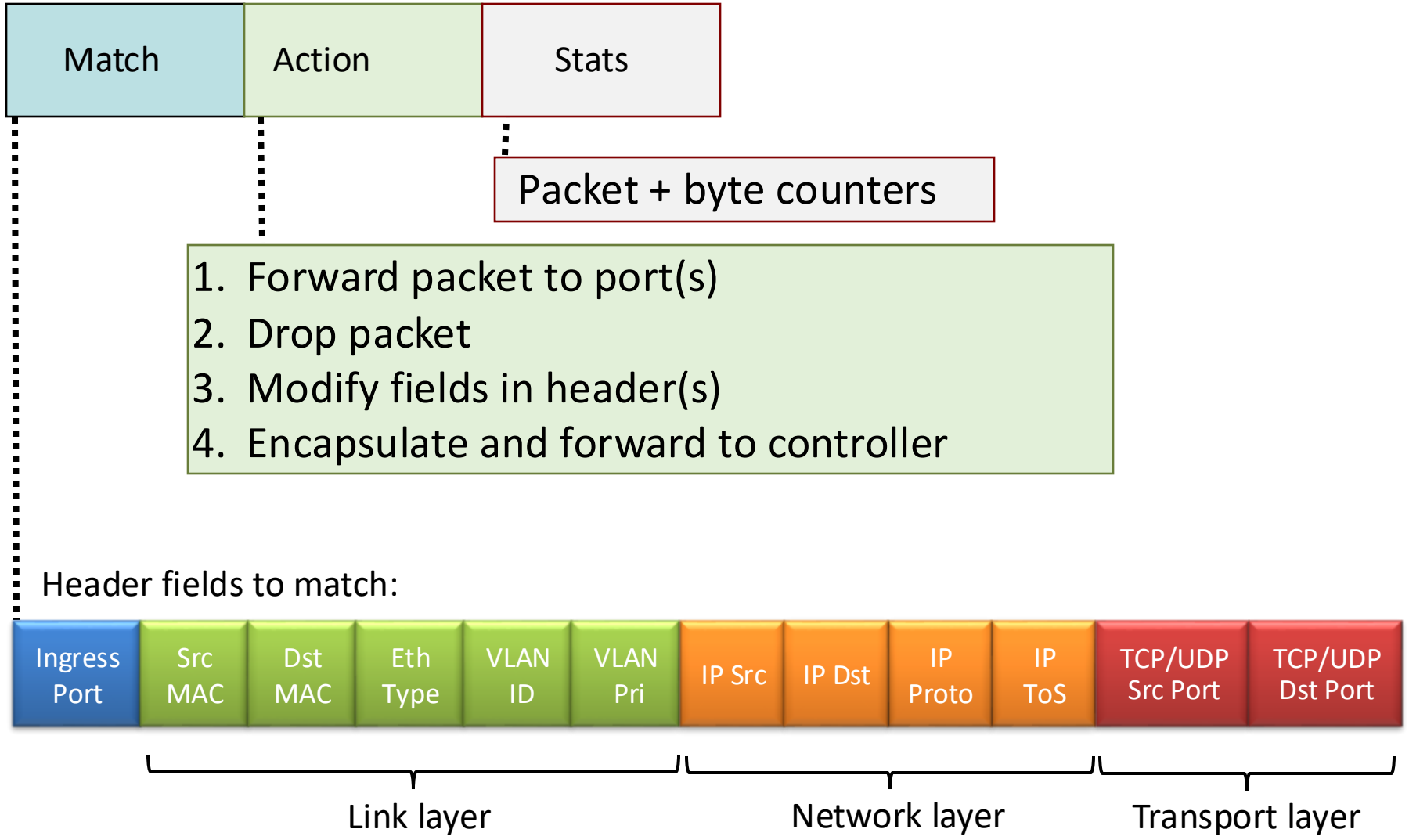
②. OpenFlow Messages

Why Generalized Abstractions?

①. Flexible routing

②. More than routing (NAT / Access control list)

OpenFlow: flow table entries



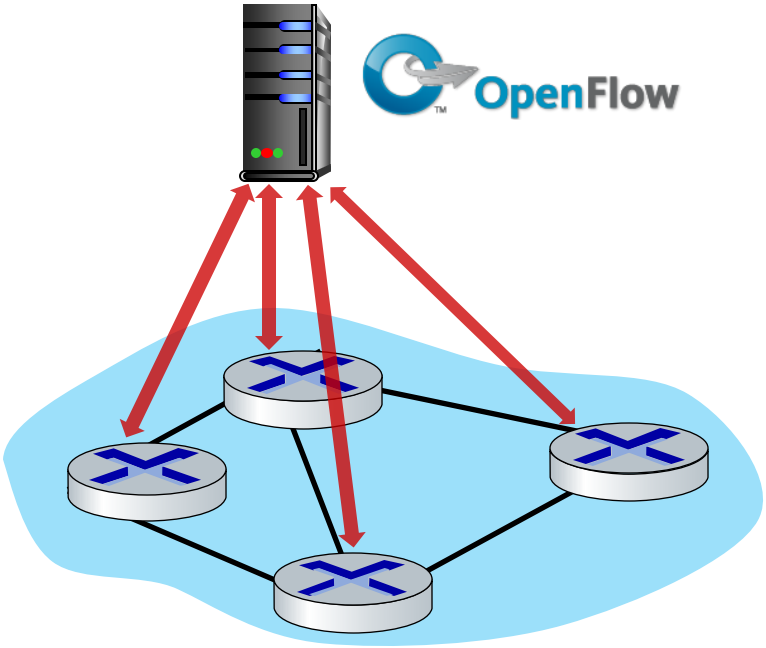
Two main piece of information

1. Information about the data plane (switch configuration, link cost, updates etc)
2. Dissemenate flow tables to the switches (also configures switch)

OpenFlow Messages

- TCP used to exchange messages
  - optional encryption
- Three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc.)
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions

OpenFlow Controller

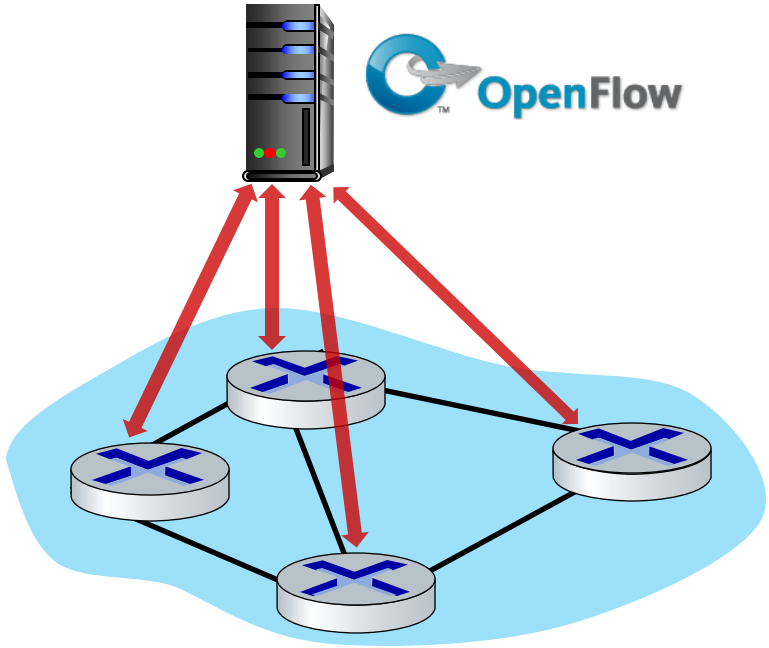


OpenFlow: controller-to-switch messages

Key controller-to-switch messages

- *features:* controller queries switch features, switch replies
- *configure:* controller queries/sets switch configuration parameters
- *modify-state:* add, delete, modify flow entries in the OpenFlow tables
- *packet-out:* controller can send this packet out of specific switch port

OpenFlow Controller

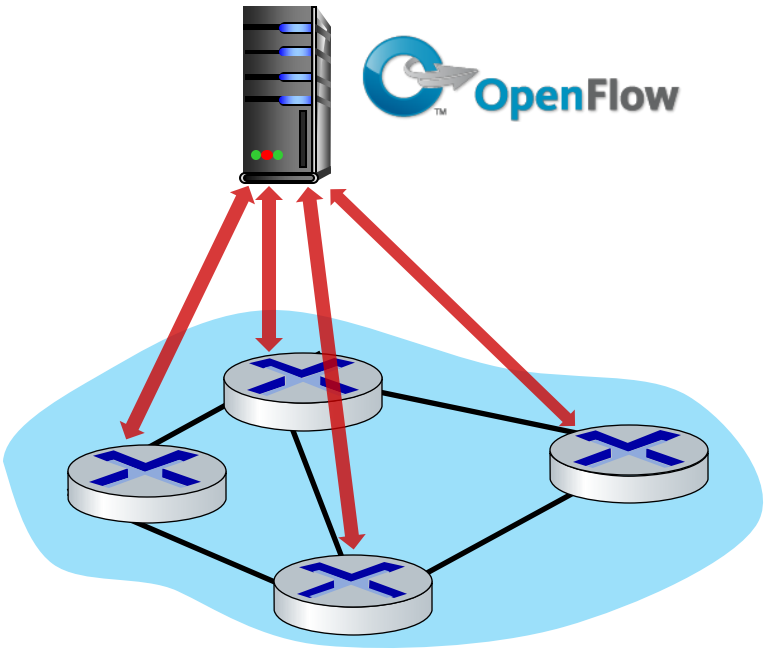


OpenFlow: switch-to-controller messages

Key switch-to-controller messages

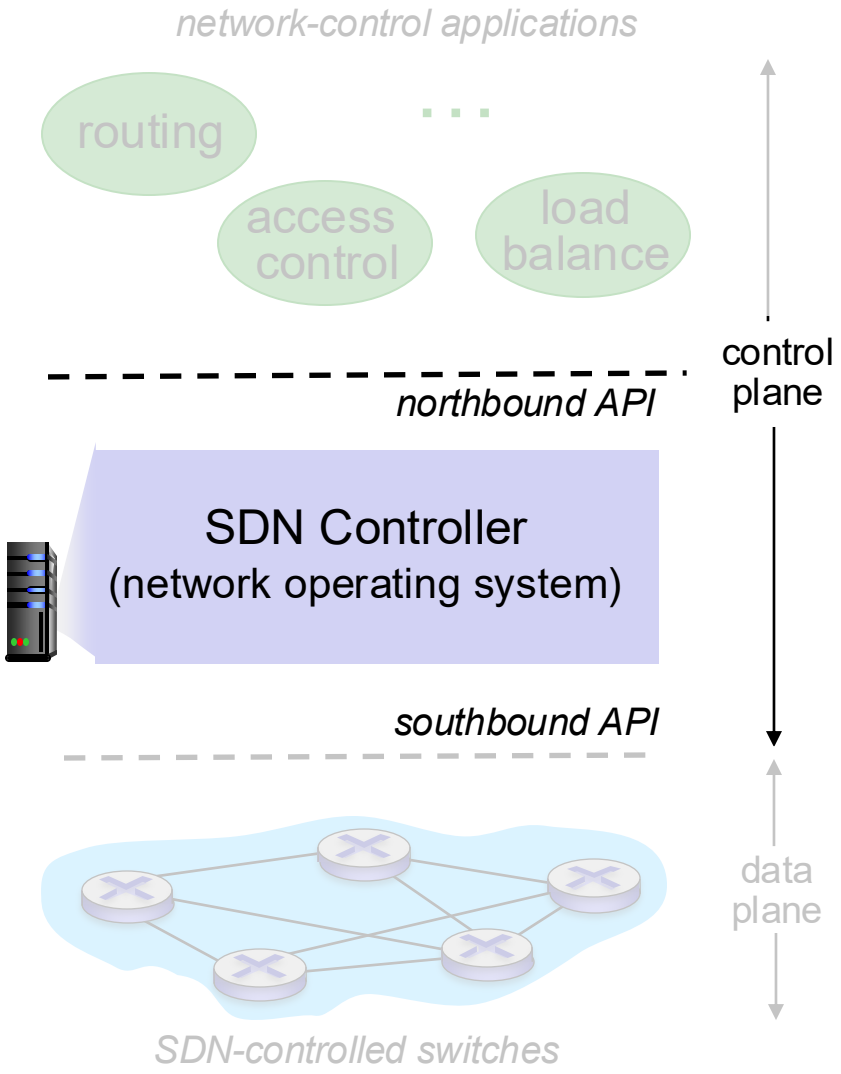
- *packet-in:* transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed:* flow table entry deleted at switch
- *port status:* inform controller of a change on a port.

OpenFlow Controller



SDN Controller

- maintains network state information
- interacts with network control applications “above” via northbound API
- interacts with network switches “below” via southbound API
- implemented as distributed system for performance, scalability, fault-tolerance, robustness

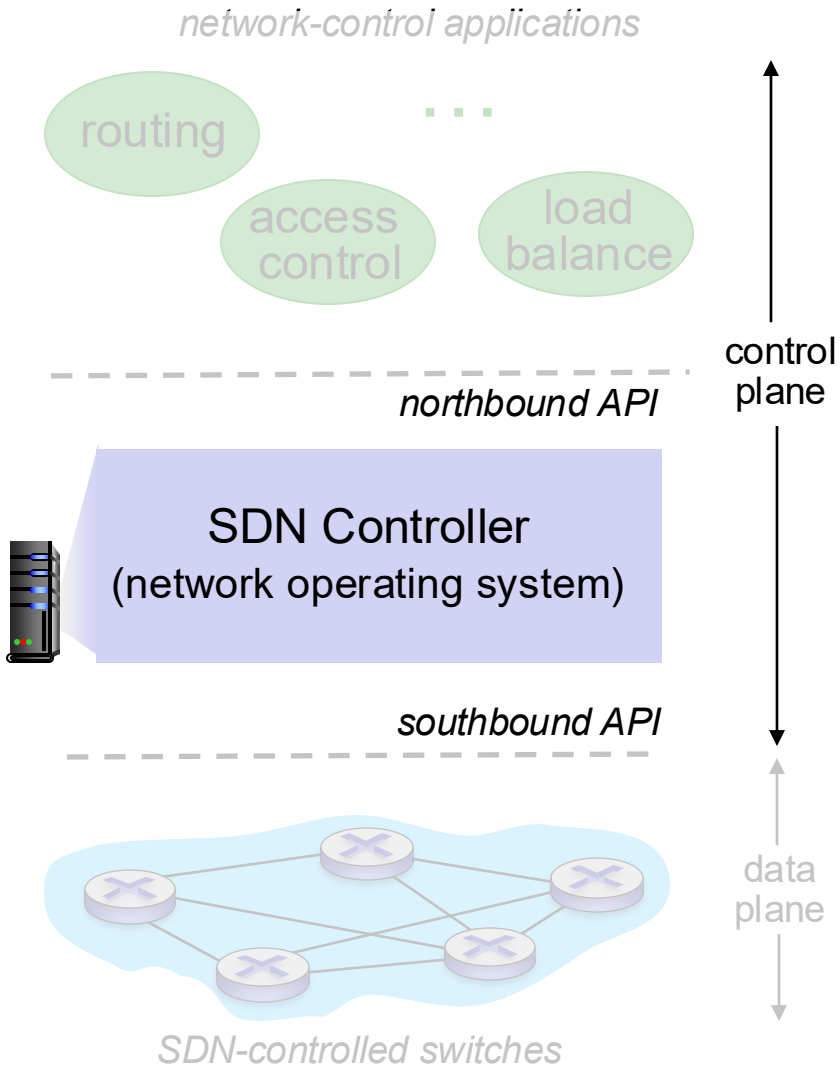


Example of Controllers

- **Ryu**
- **Floodlight**
- **NOX/POX**
- **OpenDayLight (ODL)**
- **Open Network Operating System (ONOS)**
- Pyretic
- Frenetic
- Procera

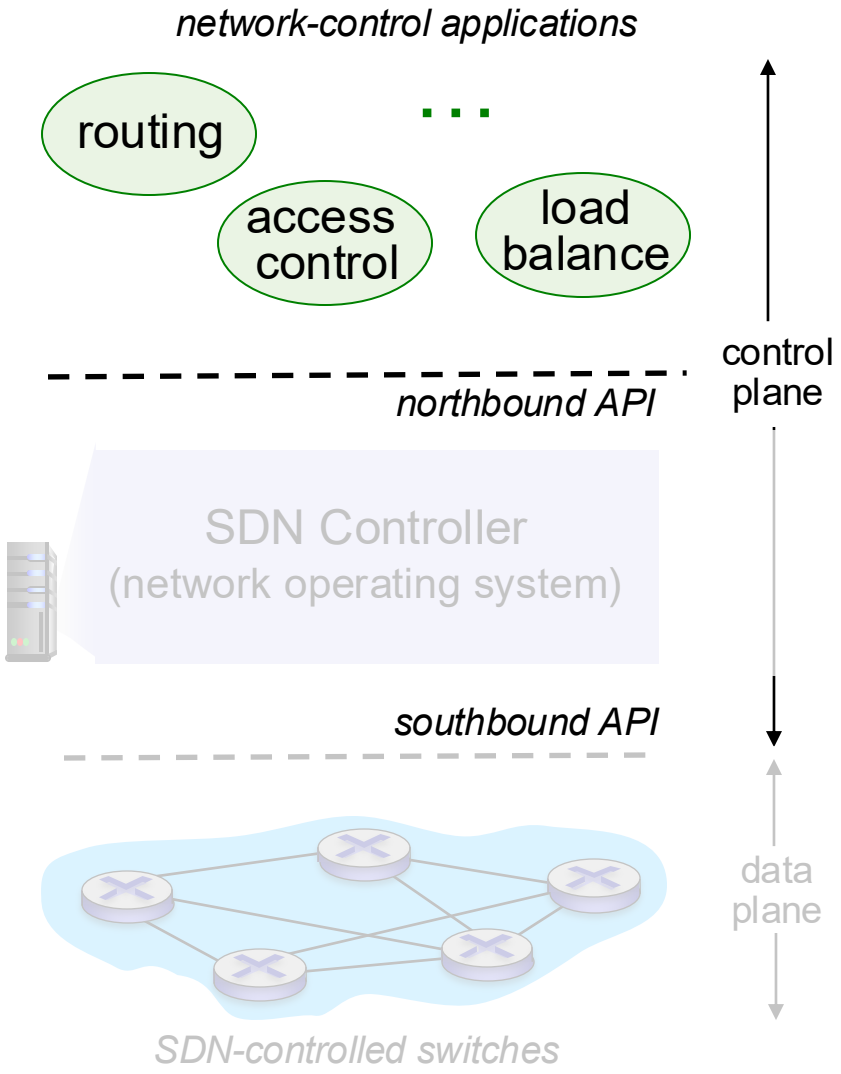
Choosing the right SDN controller?

- Use case
- Learning curve, programming language
- Focus (Southbound API, Northbound API)
- Community support

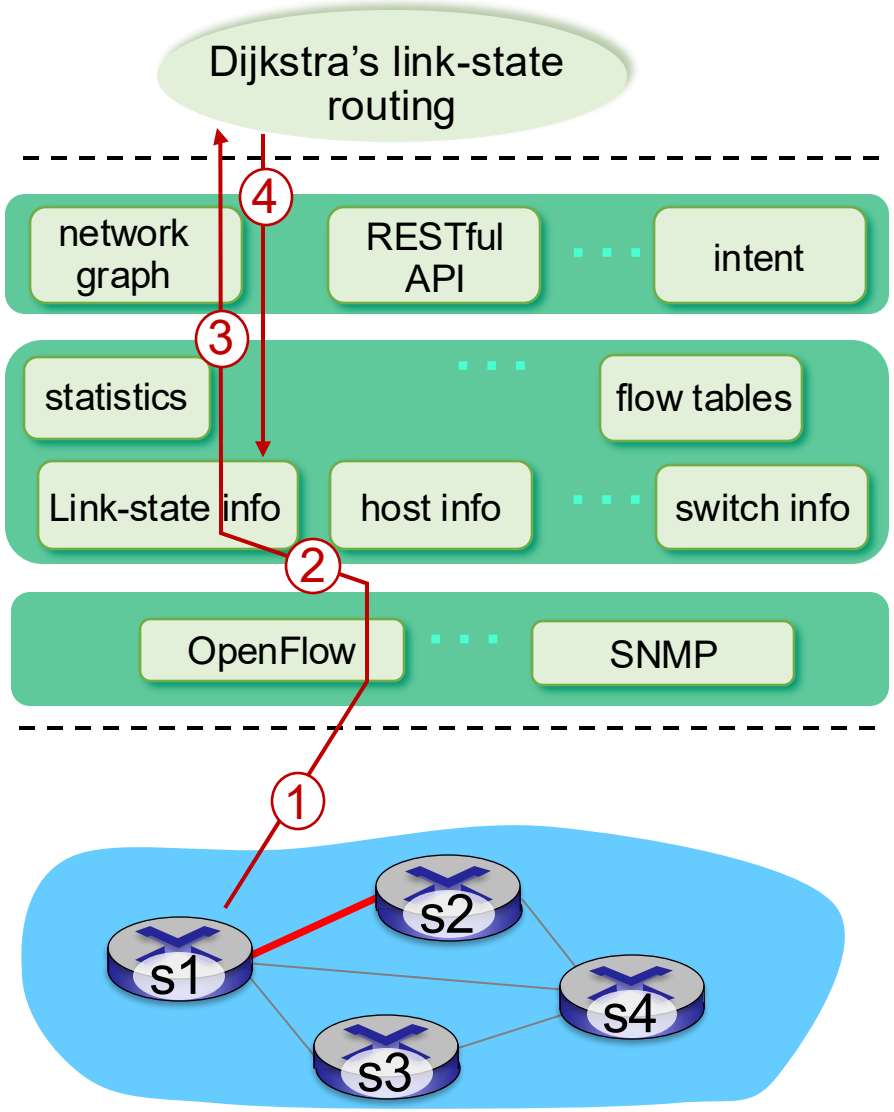


Software defined networking (SDN)

- operators don't "program" switches by creating/sending OpenFlow messages directly.
- instead use higher-level abstraction at controller
- "brains" of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled:* can be provided by 3rd party: distinct from routing vendor, or SDN controller



SDN: control/data plane interaction example



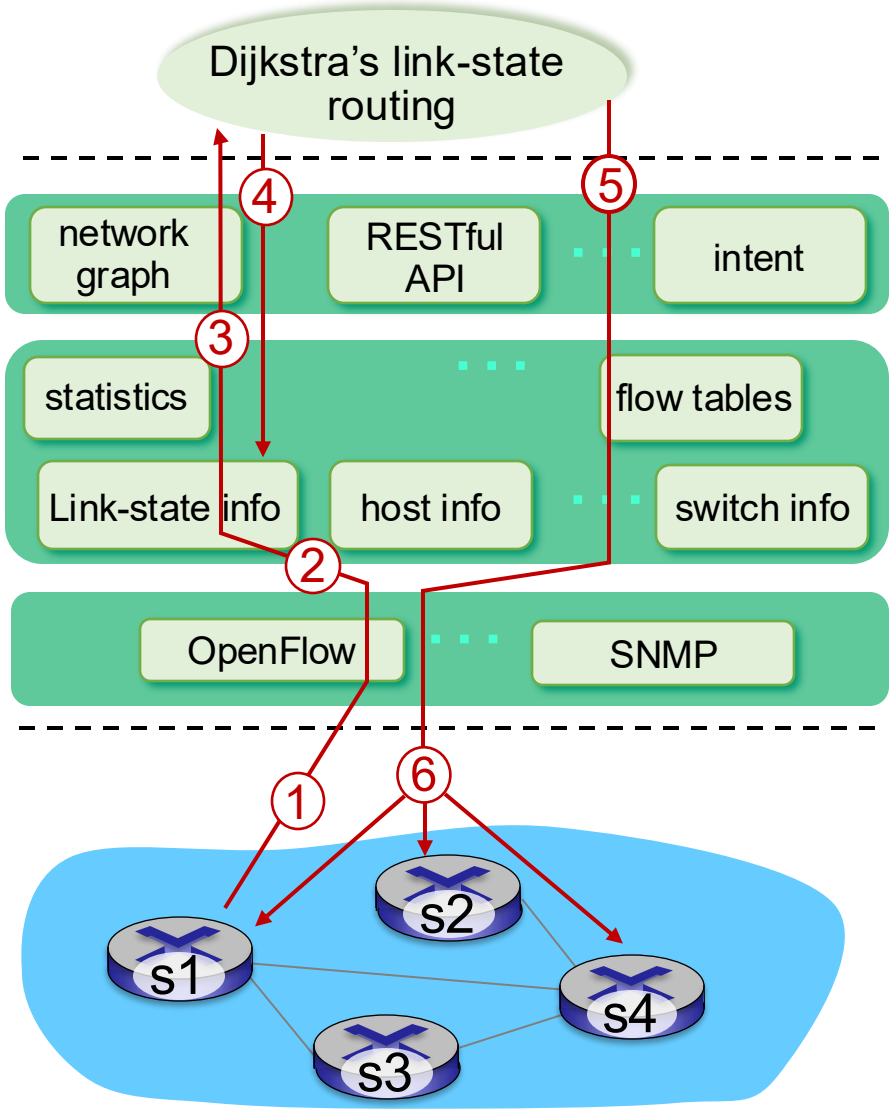
① S1, experiencing link failure uses OpenFlow port status message to notify controller

② SDN controller receives OpenFlow message, updates link status info

③ Dijkstra's routing algorithm application has previously registered to be called whenever link status changes. It is called.

④ Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes

SDN: control/data plane interaction example



⑤ link state routing app interacts with flow-table-computation component in SDN controller, which computes new flow tables needed

⑥ controller uses OpenFlow to install new tables in switches that need updating