CCNA Day 59

Network Automation

6.0 Automation and Programmability   10% ∧

6.1 Explain how automation impacts network management
6.2 Compare traditional networks with controller-based networking
6.3 Describe controller-based and software defined architectures (overlay, underlay, and fabric)
    • 6.3.a Separation of control plane and data plane
    • 6.3.b North-bound and south-bound APIs
6.4 Compare traditional campus device management with Cisco DNA Center enabled device management
6.5 Describe characteristics of REST-based APIs (CRUD, HTTP verbs, and data encoding)
6.6 Recognize the capabilities of configuration management mechanisms Puppet, Chef, and Ansible
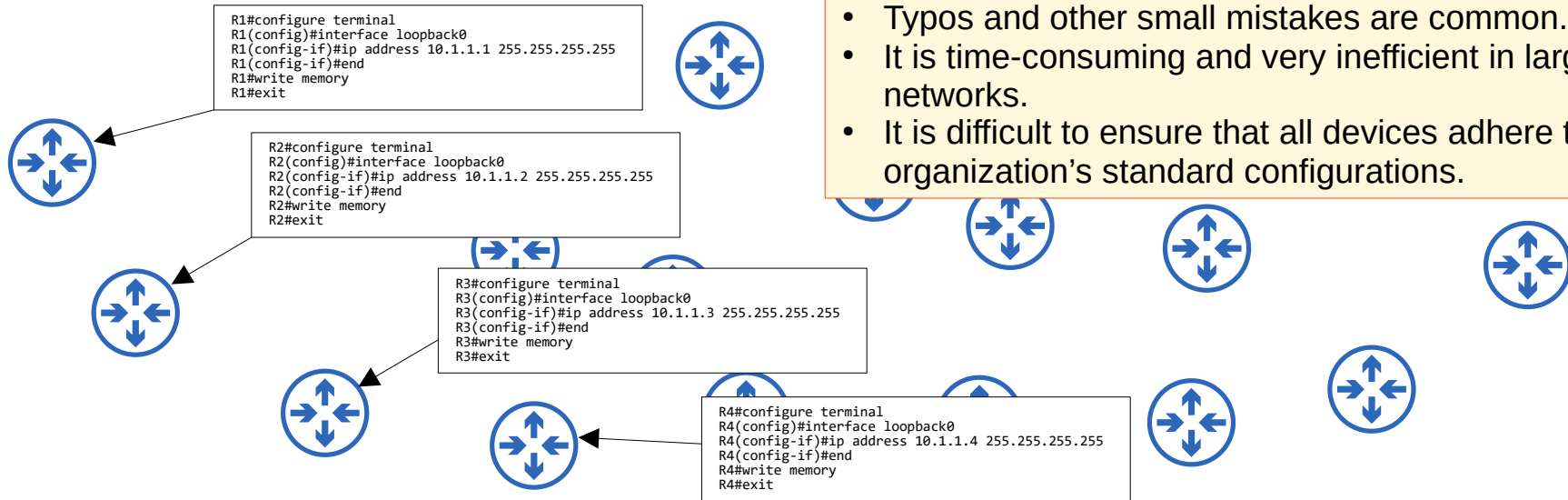6.7 Interpret JSON encoded data

- Why network automation?

- Benefits of network automation

- Logical 'planes' of network functions

- Software-defined networking (SDN)

- APIs

- Data serialization

- Previous versions of the CCNA focused on the traditional model of managing/controlling networks.

- The current version focuses on the traditional model as well, but CCNA candidates are expected to have a basic understanding of various topics related to network automation.

- In the traditional model, engineers manage devices one at a time by connecting to their CLI via SSH.

```
R1#configure terminal
R1(config)#interface loopback0
R1(config-if)#ip address 10.1.1.1 255.255.255.255
R1(config-if)#end
R1#write memory
R1#exit
```

```
R2#configure terminal
R2(config)#interface loopback0
R2(config-if)#ip address 10.1.1.2 255.255.255.255
R2(config-if)#end
R2#write memory
R2#exit
```

```
R3#configure terminal
R3(config)#interface loopback0
R3(config-if)#ip address 10.1.1.3 255.255.255.255
R3(config-if)#end
R3#write memory
R3#exit
```

```
R4#configure terminal
R4(config)#interface loopback0
R4(config-if)#ip address 10.1.1.4 255.255.255.255
R4(config-if)#end
R4#write memory
R4#exit
```
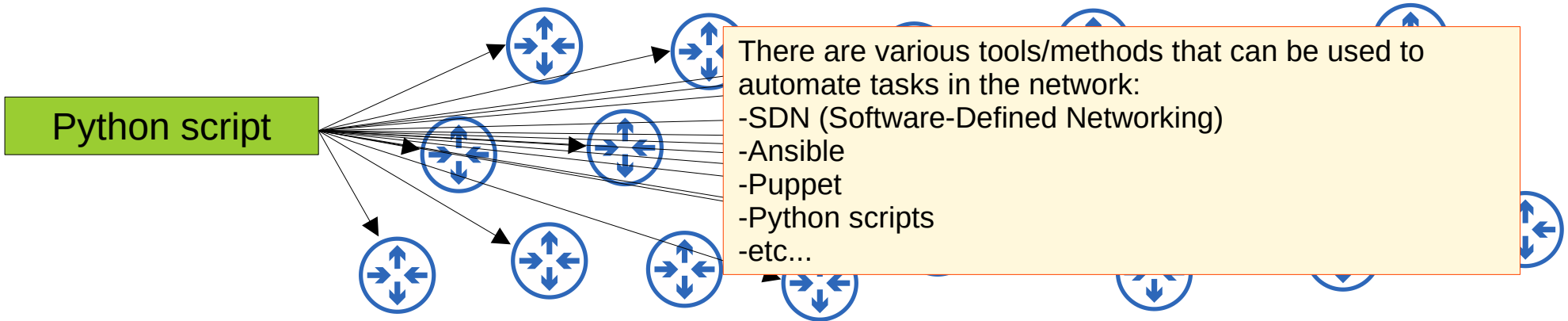
Some downsides of configuring devices one-by-one:
- Typos and other small mistakes are common.
- It is time-consuming and very inefficient in large-scale networks.
- It is difficult to ensure that all devices adhere to the organization's standard configurations.

Network automation provides many key benefits:

- Human error (typos etc.) is reduced.

- Networks become much more scalable. New deployments, network-wide changes, and troubleshooting can be implemented in a fraction of the time.

- Network-wide policy compliance can be assured (standard configurations, software versions, etc).

- The improved efficiency of network operations reduces the opex (operating expenses) of the network. Each task requires fewer man-hours.

Python script

There are various tools/methods that can be used to automate tasks in the network:
-SDN (Software-Defined Networking)
-Ansible
-Puppet
-Python scripts
-etc...

What does a router do?
- It forwards messages between networks by examining information in the Layer 3 header.
- It uses a routing protocol like OSPF to share route information with other routers and build a routing table.
- It uses ARP to build an ARP table, mapping IP addresses to MAC addresses.
- It uses Syslog to keep logs of events that occur.
- It allows a user to connect to it via SSH and manage it.

What does a switch do?
- It forwards messages within a LAN by examining information in the Layer 2 header.
- It uses STP to ensure there are no Layer 2 loops in the network.
- It builds a MAC address table by examining the source MAC address of frames.
- It uses Syslog to keep logs of events that occur.
- It allows a user to connect to it via SSH and manage it.

The various functions of network devices can be logically divided up (categorized) into *planes*:
- Data plane
- Control plane
- Management plane

- All tasks involved in forwarding user data/traffic from one interface to another are part of the **data plane**.

- A router receives a message, looks for the most specific matching route in its routing table, and forwards it out of the appropriate interface to the next hop.
  → It also de-encapsulates the original Layer 2 header, and re-encapsulates with a new header destined for the next hop's MAC address.

- A switch receives a message, looks at the destination MAC address, and forwards it out of the appropriate interface (or floods it).
  → This includes functions like adding or removing 802.1q VLAN tags.

- NAT (changing the src/dst addresses before forwarding) is part of the data plane.

- Deciding to forward or discard messages due to ACLs, port security, etc. is part of the data plane.

- The data plane is also called the 'forwarding plane'.
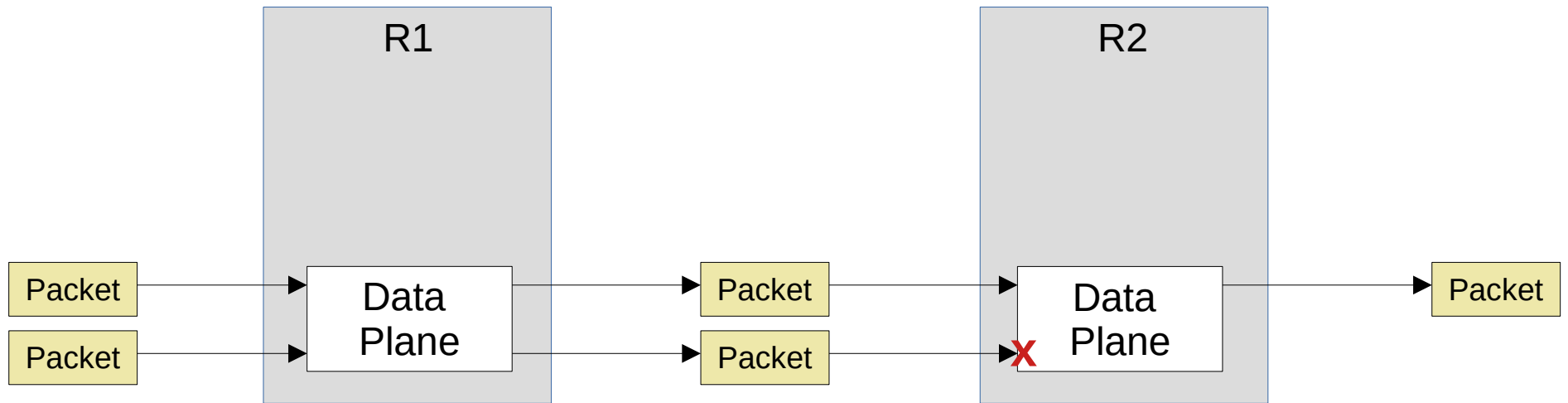
Data Plane

- How does a device's data plane make its forwarding decisions?
  → routing table, MAC address table, ARP table, STP, etc.

- Functions that build these tables (and other functions that influence the data plane) are part of the **control plane**.

- The control plane *controls* what the data plane does, for example by building the router's routing table.

- The control plane performs *overhead* work.
  → OSPF itself doesn't forward user data packets, but it informs the data plane about how packets should be forwarded.
  → STP itself isn't directly involved in the process of forwarding frames, but it informs the data plane about which interfaces should and shouldn't be used to forward frames.
  → ARP messages aren't user data, but they are used to build an ARP table which is used in the process of forwarding data.

R1

Control Plane

OSPF

R2

Control Plane

Packet

Packet

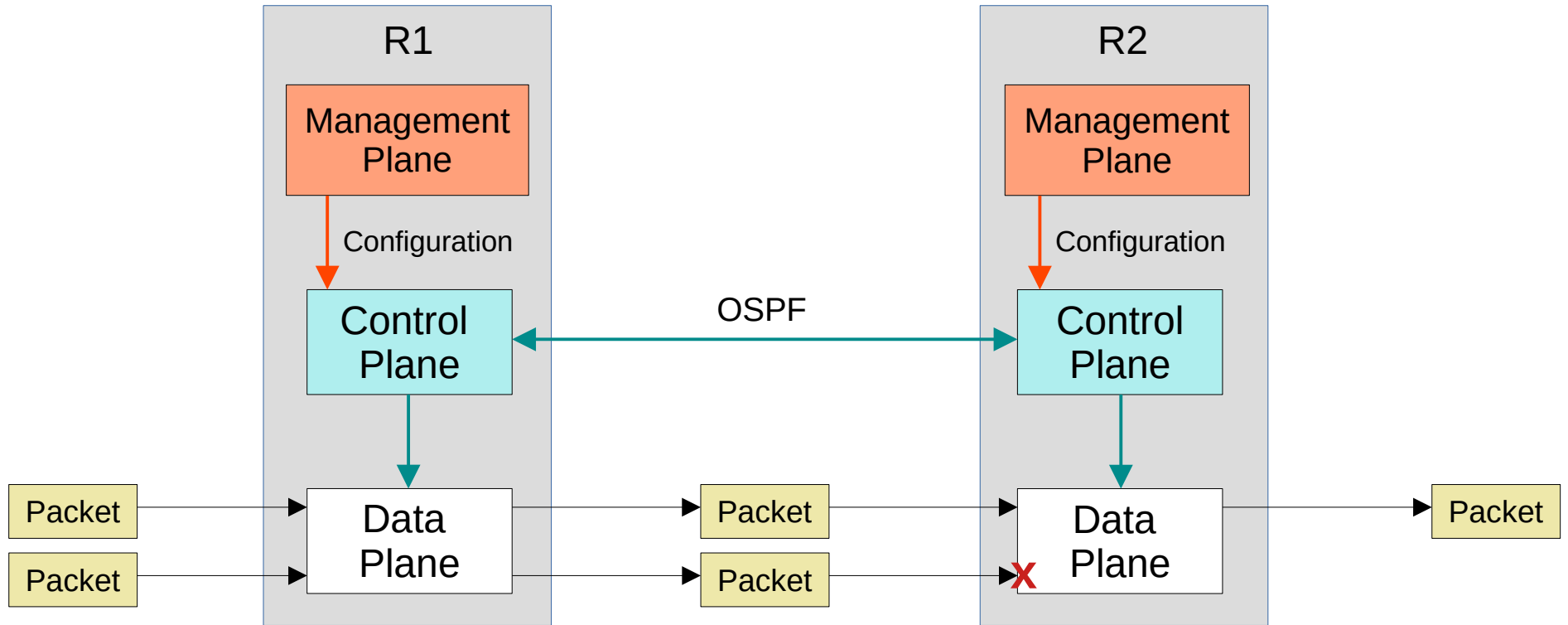Data Plane

Packet

Packet

Data Plane

Packet
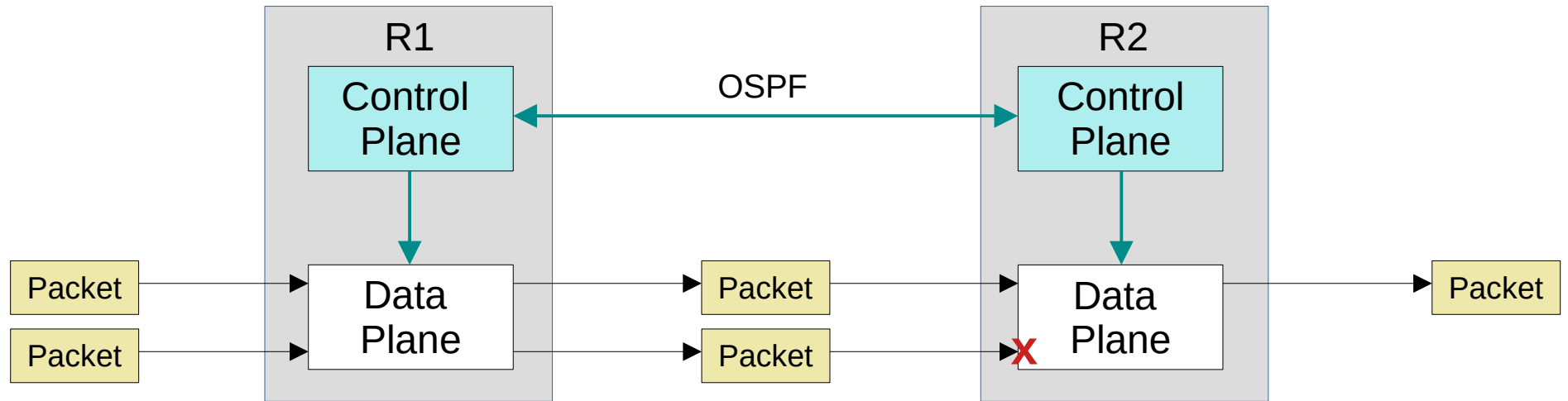
In traditional networking, the data plane and control plane are both distributed. Each device has its own data plane and its own control plane. The planes are 'distributed' throughout the network.

- Like the control plane, the **management plane** performs overhead work.
  → However, the management plane doesn't directly affect the forwarding of messages in the data plane.

- The management plane consists of protocols that are used to manage devices.
  → SSH/Telnet, used to connect to the CLI of a device to configure/manage it.
  → Syslog, used to keep logs of events that occur on the device.
  → SNMP, used to monitor the operations of the device.
 → NTP, used to maintain accurate time on the device.

Management Plane

R1

Management Plane

Configuration

Control Plane

OSPF

Data Plane

Packet

Packet

Packet

Packet

R2

Management Plane

Configuration

Control Plane

Data Plane

Packet

Packet ✗

Packet

The Data plane is the reason we buy routers and switches (and network infrastructure in general), to forward messages. However, the Control plane and Management plane are both necessary to enable the data plane to do its job.

- The operations of the Management plane and Control plane are usually managed by the CPU.

- However, this is not desirable for data plane operations because CPU processing is slow (relatively speaking).

- Instead, a specialized hardware ASIC (Application-Specific Integrated Circuit) is used. ASICs are chips built for specific purposes.

- Using a switch as an example:
  → When a frame is received, the ASIC (not the CPU) is responsible for the switching logic.
  → The MAC address table is stored in a kind of memory called TCAM (Ternary Content-Addressable Memory).
  *Another common name for the MAC address table is *CAM table*.
  → The ASIC feeds the destination MAC address of the frame into the TCAM, which returns the matching MAC address table entry.
  → The frame is then forwarded out of the appropriate interface.

- Modern routers also use a similar hardware data plane: an ASIC designed for forwarding logic, and tables stored in TCAM.
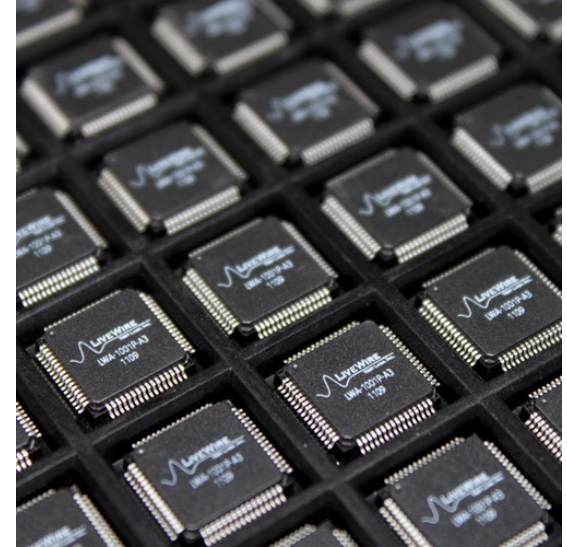

https://en.wikipedia.org/wiki/Application-specific_integrated_circuit

A simple summary:
- When a device receives control/ management traffic (destined for itself), it will be processed in the CPU.
- When a device receives data traffic which should pass through the device, it is processed by the ASIC for maximum speed.

- **Software-Defined Networking (SDN)** is an approach to networking that centralizes the control plane into an application called a *controller.*
  →You are already familiar with this concept from learning about Wireless LAN Controllers.

- SDN is also called **Software-Defined Architecture (SDA)** or **Controller-Based Networking**.

- Traditional control planes use a distributed architecture.
  →For example, each router in the network runs OSPF and the routers share routing information and then calculate their preferred routes to each destination.

- An SDN controller centralizes control plane functions like calculating routes.
  →That is just an example, and how much of the control plane is centralized varies greatly.

- The controller can interact programmatically with the network devices using APIs (Application Programming Interface).
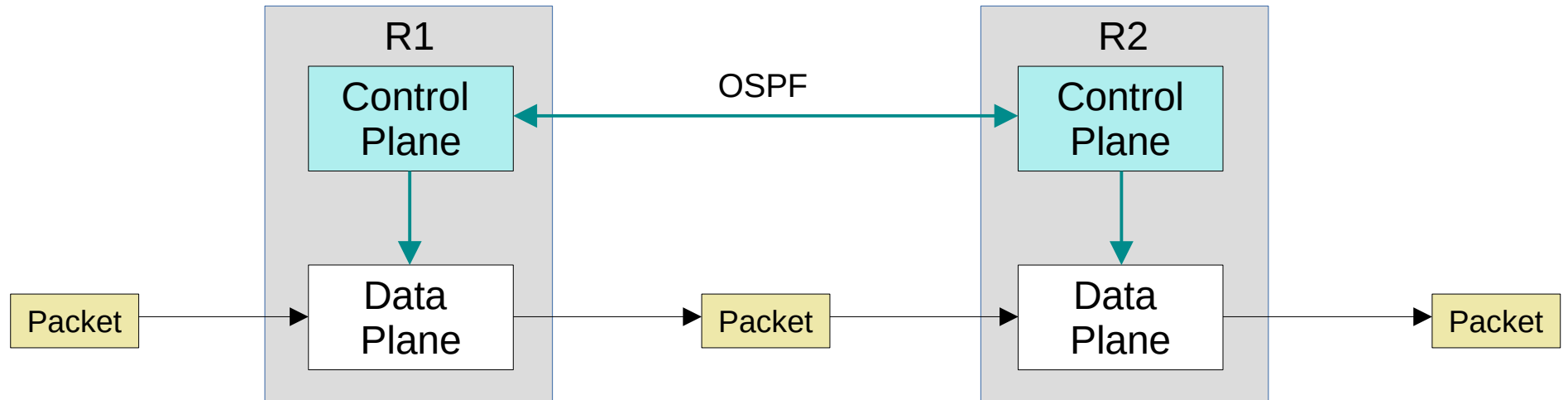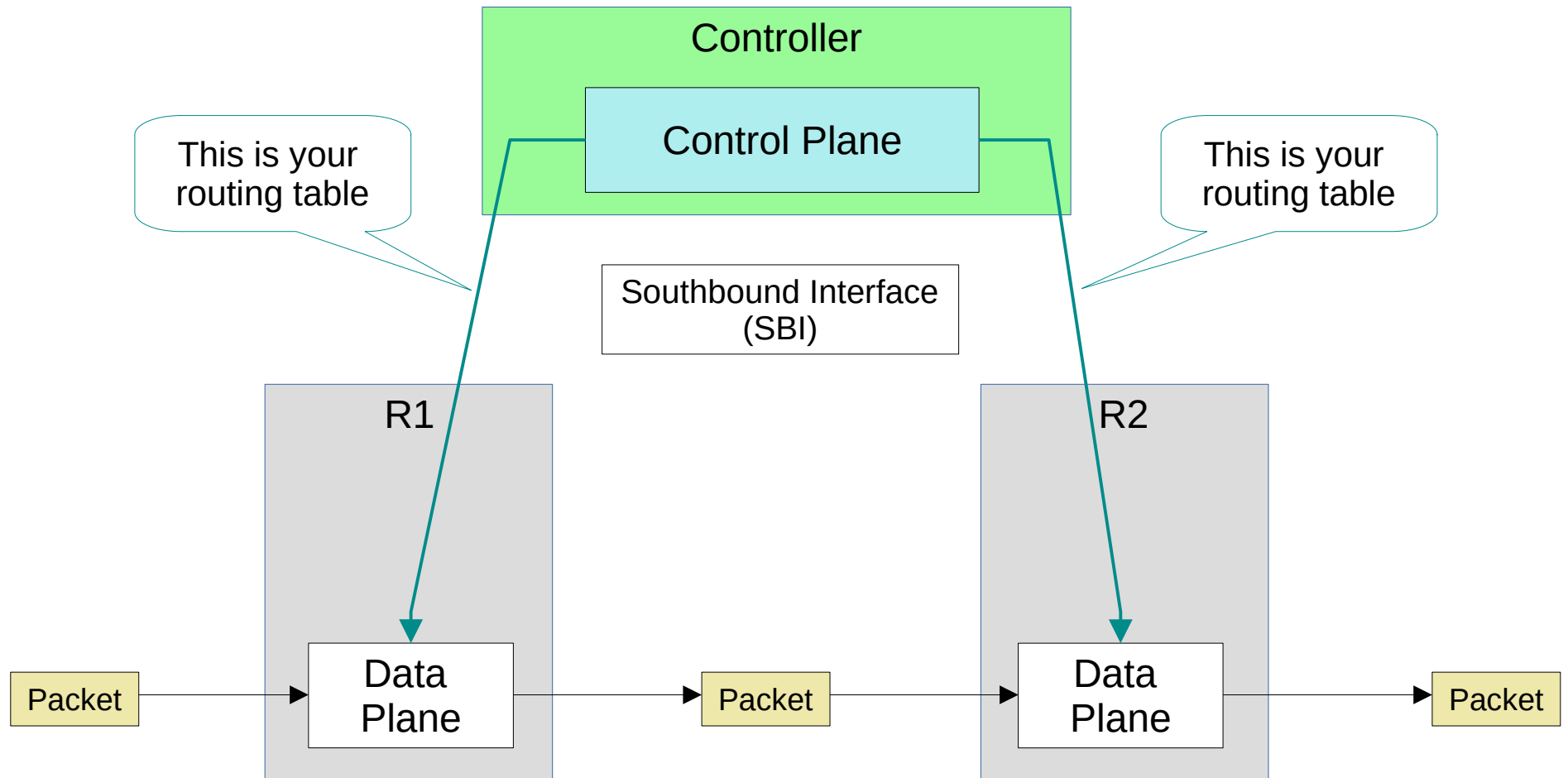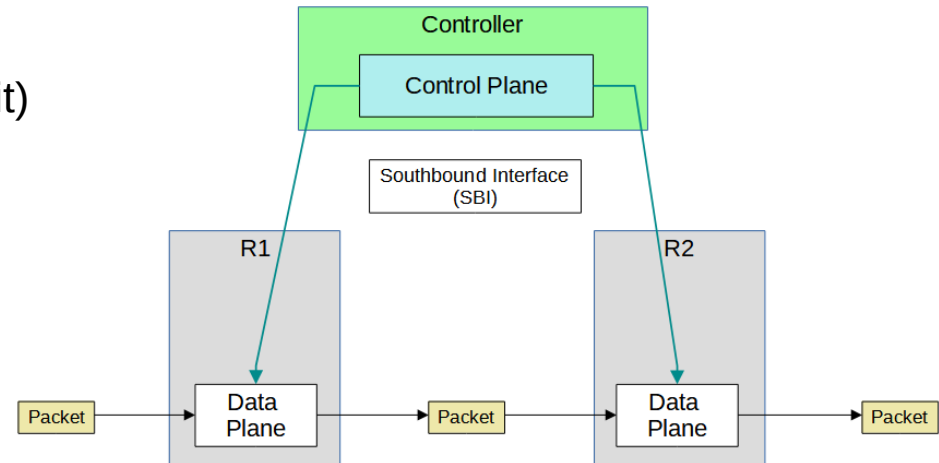
Southbound Interface (SBI)

- The SBI is used for communications between the controller and the network devices it controls.

- It typically consists of a communication protocol and API (Application Programming Interface).

- APIs facilitate data exchanges between programs.
  - → Data is exchanged between the controller and the network devices.
  - → An API on the network devices allows the controller to access information on the devices, control their data plane tables, etc.

- Some examples of SBIs:
  - → OpenFlow
  - → Cisco OpFlex
  - → Cisco onePK (Open Network Environment Platform Kit)
  - → NETCONF

- Using the SBI, the controller communicates with the managed devices and gathers information about them:
  - → The devices in the network
  - → The topology (how the devices are connected together)
  - → The available interfaces on each device
  - → Their configurations

- The **Northbound Interface (NBI)** is what allows us to interact with the controller, access the data it gathers about the network, program it, and make changes in the network via the SBI.

- A *REST API* is used on the controller as an interface for apps to interact with it.
  - → REST = Representational State Transfer

- Data is sent in a structured (*serialized*) format such as JSON or XML.
  - → This makes it much easier for programs to use the data.
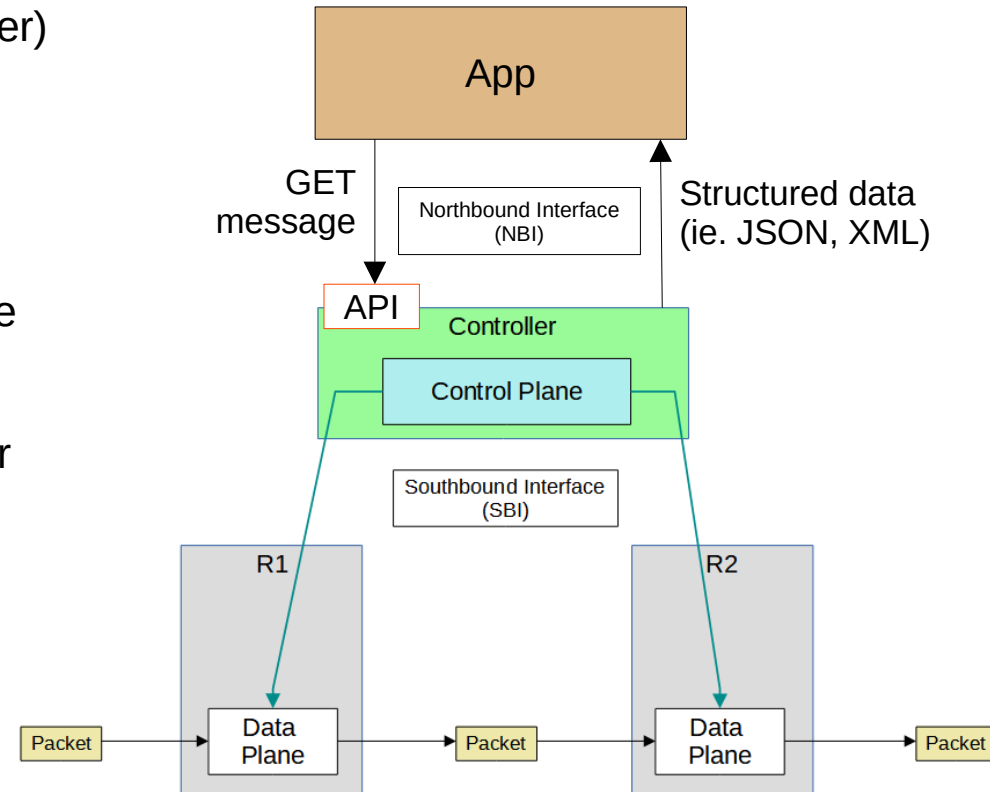
- Networking tasks can be automated in traditional network architectures too:
  → Scripts can be written (ie. using Python) to push commands to many devices at once.
  → Python with good use of Regular Expressions can parse through **show** commands to gather information about the network devices.

```
Switch#show interfaces
GigabitEthernet0/0 is up, line protocol is up (connected)
  Hardware is iGbE, address is 0cb0.28f6.5500 (bia 0cb0.28f6.5500)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Unknown, Unknown, link type is auto, media type is unknown media type
  output flow-control is unsupported, input flow-control is unsupported
  Auto-duplex, Auto-speed, link type is auto, media type is unknown
  input flow-control is off, output flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last input 00:00:00, output 00:00:09, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     83 packets input, 8576 bytes, 0 no buffer
     Received 82 broadcasts (82 multicasts)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 82 multicast, 0 pause input
     54 packets output, 7221 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 unknown protocol drops
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier, 0 pause output
     0 output buffer failures, 0 output buffers swapped out
```

- Networking tasks can be automated in traditional networks:
  - → Scripts can be written (ie. using Python) to push commands to many devices at once.
  - → Python with good use of Regular Expressions can parse through **show** commands to gather information about the network devices.

- However, the robust and centralized data collected by SDN controllers greatly facilitates these functions.
  - → The controller collects information about all devices in the network.
  - → Northbound APIs allow apps to access information in a format that is easy for programs to understand (ie. JSON, XML).
  - → The centralized data facilitates network-wide analytics.

- SDN tools can provide the benefits of automation without the requirement of third-party scripts & apps.
  - → You don't need expertise in automation to make use of SDN tools.
  - → However, APIs allow third-party applications to interact with the controller, which can be very powerful.

> Although SDN and automation aren't the same thing, the SDN architecture greatly facilitates the automation of various tasks in the network via the SDN controller and APIs.

- Why network automation?

- Benefits of network automation

- Logical 'planes' of network functions

- Software-defined networking (SDN)

- APIs (REST)

- Data serialization (JSON, XML)

Which of the following are benefits of network automation? (select two)

a) Reduced human error.

b) Reduced OpEx.

c) Reduced hardware costs.

d) Centralized data plane.

Which of the following are SBIs? (select two)

a) REST

b) API

c) OpenFlow

d) OpFlex

Which of the following network functions would you expect to be centralized in SDN?

a) Calculating routes

b) Forwarding packets

c) Translating source IP addresses of packets

d) Denying packets via an ACL

What is the purpose of the SBI in SDN architecture?

a) To facilitate data exchange between users.

b) To facilitate data exchange between the controller and apps.

c) To facilitate data exchange between the controller and network devices.

d) To facilitate data exchange between controllers.
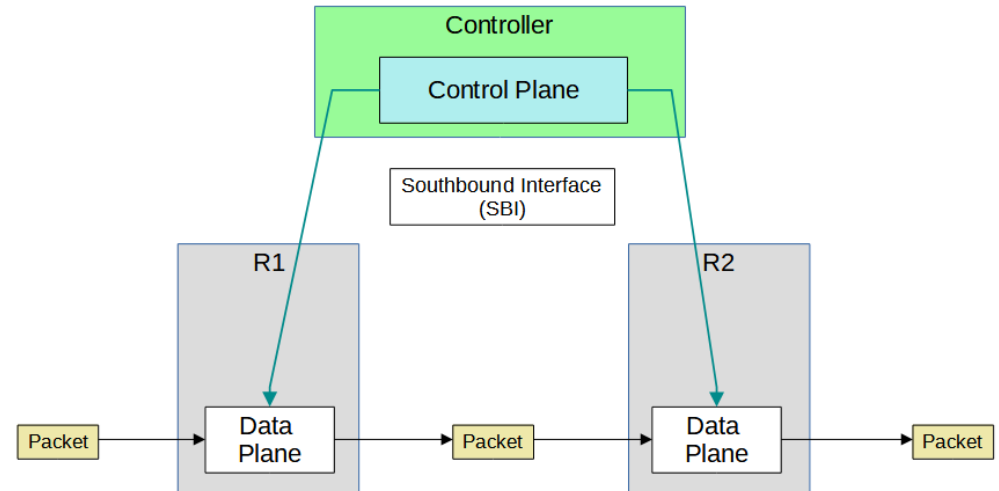
Which of the logical planes of networking does NTP's function fit in?

a) Control plane

b) Management plane

c) Data plane

d) Forwarding plane