

Homework 6: CS 524

1. (10 points) Explain the motivation for developing COPS. What were the goals of COPS framework?

ANS.

COPS (Common Open Policy Service Protocol) is a proposed standard protocol for exchanging network policy information between a policy decision point (PDP) in a network and policy enforcement points (PEPs) as part of overall Quality of Service (QoS) - the allocation of network traffic resources according to desired priorities of service. The policy decision point might be a network server controlled directly by the network administrator who enters policy statements about which kinds of traffic (voice, bulk data, video, teleconferencing, and so forth) should get the highest priority. The policy enforcement points might be routers or layer 3 switches that implement the policy choices as traffic moves through the network. Currently, COPS is designed for use with the Resource Reservation Protocol (RSVP), which lets you allocate traffic priorities in advance for temporary high bandwidth requirements.

MOTIVATION:

The IETF started to address the problem gradually with other protocols like SNMP, strictly on a specific need basis. The first such need was the policy configuration in support of QoS. The proposed model had introduced new challenges - the need to maintain a synchronized state between the network manager and the device. Another challenge came from the potential interference among two or more network managers administering the same device. And then there is a need for policy-based management.

Network providers wanted to have a mechanism that would enable granting a resource based on a set of policy rules. The decision on whether to grant the resource takes into account information about the user, the requested service, and the network itself. Employing SNMP for this purpose was not straightforward, and so the IETF developed a new protocol, for communications between the network element and the Policy Decision Point (PDP)—where the policy-based decisions were made. The protocol is called Common Open Policy Service (COPS).

As an important aside, COPS has greatly influenced the Next-Generation telecommunications Network (NGN) standards, characterized by (1) the prevalent use of IP for end-to-end packet transfer and (2) the drive to convergence between wireline and wireless technologies. As we can see, COPS has solved the problem of policy-based management.

GOAL:

Among the goals of the framework are support for pre-emption, various policy styles, monitoring, and accounting. Pre-emption here means the ability to remove a previously granted resource so as to accommodate a new request.¹² As far as the policy styles go, those to be supported include bi-lateral and multi-lateral service agreements and

policies based on the notion of relative priority, as defined by the provider. Support for monitoring and accounting is effected by gathering the resource use and access data. The framework also states the requirement for fault tolerance and recovery in cases when a PDP fails or cannot be reached.

(reference: [What is COPS \(Common Open Policy Service Protocol\)? - Definition from WhatIs.com \(techtarget.com\)](https://www.techtarget.com/whatis/definition/Common-Open-Policy-Service-Protocol-COPS), Textbook: Cloud Computing: Business Trends and Technologies)

2. (10points) What feature is intrinsically new to COPS (compared to the SNMP)

ANS.

The feature that is intrinsically new to COPS—as compared with SNMP is that COPS employs a stateful client–server model, which is different from that of the remote procedure call. As in any client–server model, the PEP (client) sends requests to the remote PDP (server), and the PDP responds with the decisions. But all the requests from the client PEP are installed and remembered by the remote PDP until they are explicitly deleted by the PEP. The decisions can come in the form of a series of notifications to a single request. This, in fact, introduces a new behavior: two identical requests may result in different responses because the states of the system when the first and second of these requests arrive may be different depending on which states had been installed. Another stateful feature of COPS is that PDP may “push” the configuration information to the client and later remove it.

The COPS stateful model supports two mechanisms of policy control, called respectively the outsourcing model and the configuration model. With the outsourcing mechanism, PEP queries PDP every time it needs a decision; when the configuration mechanism is employed, PDP provisions the policy decision within the PEP.

Unlike SNMP, COPS was designed to leverage self-identifying objects and therefore it is extensible. COPS also runs on TCP, which ensures reliable transport. Although COPS may rely on TLS, it also has its own mechanisms for authentication, protection against replays, and message integrity.

(References: Textbook: Cloud Computing: Business Trends and Technologies)

3. (30points) Motivation for developing NETCONF.

Read RFC 3535, and answer the following questions (you can cite the relevant text of RFC 3535 verbatim). Each correct answer is worth 10 points.

- a. Why the SNMP transactional model and the protocol constraints make it more complex to implement MIBs, as compared to the implementation of commands of a command-line interface interpreter?**

ANS.

The SNMP transactional model and the protocol constraints make it more complex to implement MIBs, as compared to the implementation of commands of a command line interface interpreter. A logical operation on a MIB can turn into a sequence of SNMP interactions where the implementation has to maintain state until the operation is complete, or until a failure has been determined. In case of a failure, a robust implementation must be smart enough to roll the device back into a consistent state.

(References: <https://www.rfc-editor.org/info/rfc3535>)

b. What is the problem with the SNMP lack of support for easy retrieval and playback of Configurations?

ANS.

SNMP does not support easy retrieval and playback of configurations. One part of the problem is that it is not easy to identify configuration objects. Another part of the problem is that the naming system is very specific and physical device reconfigurations can thus break the capability to play back a previous configuration.

(References: <https://www.rfc-editor.org/info/rfc3535>)

c. List the operators' requirements for network management:

ANS.

The following is the list of operator requirements:

- Ease of use is a key requirement for any network management technology from the operators point of view.
- It is necessary to make a clear distinction between configuration data, data that describes operational state and statistics. Some devices make it very hard to determine which parameters were administratively configured and which were obtained via other mechanisms such as routing protocols.
- It is required to be able to fetch separately configuration data, operational state data, and statistics from devices, and to be able to compare these between devices.
- It is necessary to enable operators to concentrate on the configuration of the network as a whole rather than individual devices.
- Support for configuration transactions across a number of devices would significantly simplify network configuration management.
- Given configuration A and configuration B, it should be possible to generate the operations necessary to get from A to B with minimal state changes and effects on network and systems. It is important to minimize the impact caused by configuration changes.

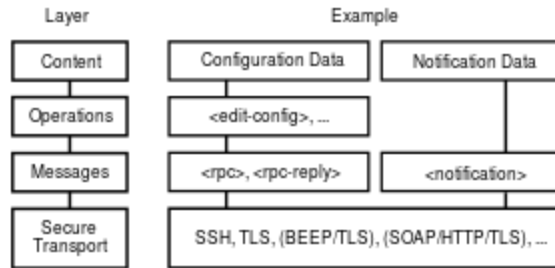
- A mechanism to dump and restore configurations is a primitive operation needed by operators. Standards for pulling and pushing configurations from/to devices are desirable.
- It must be easy to do consistency checks of configurations over time and between the ends of a link in order to determine the changes between two configurations and whether those configurations are consistent.
- Network wide configurations are typically stored in central master databases and transformed into formats that can be pushed to devices, either by generating sequences of CLI commands or complete configuration files that are pushed to devices. There is no common database schema for network configuration, although the models used by various operators are probably very similar. It is desirable to extract, document, and standardize the common parts of these network wide configuration database schemas.
- It is highly desirable that text processing tools such as diff, and version management tools such as RCS or CVS, can be used to process configurations, which implies that devices should not arbitrarily reorder data such as access control lists.
- The granularity of access control needed on management interface needs to match operational needs. Typical requirements are a role-based access control model and the principle of least privilege, where a user can be given only the minimum access necessary to perform a required task.
- It must be possible to do consistency checks of access control lists across devices.
- It is important to distinguish between the distribution of configurations and the activation of a certain configuration. Devices should be able to hold multiple configurations.
- SNMP access control is data-oriented, while CLI access control is usually command (task) oriented. Depending on the management function, sometimes data-oriented or task-oriented access control makes more sense. As such, it is a requirement to support both data-oriented and task-oriented access control.

(References: <https://www.rfc-editor.org/info/rfc3535>)

4. (10 points) Does NETCONF use REST API in its Messages layer?

ANS.


NETCONF does not use REST API in its Messages layer and instead its operations are realized on top of a simple Remote Procedure Call (RPC) layer. The NETCONF protocol uses an Extensible Markup Language (XML) based data encoding for the configuration data as well as the protocol messages. The protocol messages are exchanged on top of a secure transport protocol.



The NETCONF Messages Layer provides a simple, transport-independent framing mechanism for encoding

- RPC invocations (<rpc> messages),
- RPC results (<rpc-reply> messages), and
- event notifications (<notification> messages).

Every NETCONF message is a well-formed XML document. An RPC result is linked to an RPC invocation by a message-id attribute. NETCONF messages can be pipelined, i.e., a client can invoke multiple RPCs without having to wait for RPC result messages first. RPC messages are defined in RFC 6241 and notification messages are defined in RFC 5277.




Two Most Common Network API Protocols

NETCONF (SSH Based)

- IETF RFC standard in 2006
- Uses only XML encoded data
- Transported over SSH
- Connection oriented, transactional
- Use NETCONF RPCs to CRUD
- Supports Candidate Configuration

RESTCONF and REST APIs (HTTP/S Based APIs)

- RESTCONF IETF Standard 2017
- Uses XML or JSON encoded data
- Transported over HTTP/HTTPS
- Stateless, each request separate from next
- Use HTTP Verbs to CRUD
- Able to reuse common tooling for REST APIs from rest of industry



interop.com | [#interop](https://twitter.com/interop) | [@interop](https://twitter.com/interop)

(References: [NETCONF - Wikipedia](https://en.wikipedia.org/wiki/NETCONF))

5. (10 points) Name a de-facto modeling language for NETCONF. What document is it specified in? What is the name of this language's XML-based representation?

ANS.

YANG is the de-facto NETCONF modeling language. It is well structured, so following a module one can find both its high-level view and the ultimate encoding in NETCONF operations. By design, YANG has also been made extensible, thus allowing other SDOs to develop its extensions and individual programmers to produce plug-and-play modules.

REST vs Other Protocols



	REST	SNMP	NETCONF	SOAP
Data models		SNMP MIBs	YANG Models	
Data Modeling Language		SMI	YANG	WSDL
Management Operations	HTTP Verbs	SNMP Operations	NETCONF Operations	N/A
RPC Protocol Encoding	HTTP/XML/JSON	BER	XML	XML
Transport Stack	SSL/HTTP/TCP	UDP	SSH/TCP	SSL/HTTP/TCP

Confidential Information | December 18, 2012

8

RFC 6020 (<https://tools.ietf.org/html/rfc6020>) specifies YIN as the XML-based representation of YANG (a data-modeling language): “YANG modules can be translated into an equivalent XML syntax called YANG Independent Notation (YIN), allowing applications using XML parsers and Extensible Stylesheet Language Transformations (XSLT) scripts to operate on the models. The conversion from YANG to YIN is lossless, so content in YIN can be round-tripped back into YANG.”

(References: Textbook: Cloud Computing: Business Trends and Technologies)

6. (10 points) List the steps involved in onboarding of an application.

ANS.

Onboarding is where a service developer needs to specify which applications run on which virtual machines, what kinds of events an orchestrator needs to handle (and what exactly to do when such an event occurs), and what information to collect. The actual process of onboarding ('forklifting' workloads) has seven relatively straightforward steps:

1. Define the workload: The number and type of virtual machines required for migration will depend on the nature and scale of the workload, and the way it interacts with software and services not being migrated.

2. Provision cloud resources: Service providers will have a self-service interface for the creation of accounts and purchase of the services that you need (e.g., servers, storage, network).
3. Establish a connectivity bridge: Secure and transparent bi-directional connectivity, usually through an internet VPN, is required between your data centre and the cloud, both for the migration itself and for cross-platform application interactions after migration.
4. Deploy the workload: With connectivity in place, virtual machines can be configured and connected to services remaining behind (such as Active Directory), followed by the transfer of the application and any associated databases, software and services being migrated.
5. Ensure seamless two-way access: Smooth integration is required between the cloud workload and services not migrated, and you need to be able to monitor and manage the application as well as the cloud infrastructure.
6. Test and validate: However well you have prepared and tested prior to deployment, there may be surprises. Has everything been transferred correctly.
7. Discontinue the old service: When you are certain that everything is working well, you can give access to users and decommission the enterprise service.

(References: Textbook: Cloud Computing: Business Trends and Technologies,
http://www.interxion.com/Documents/Whitepapers%20and%20PDFs/Interxion_CloudOnboarding_Whitepaper_EN_online.pdf)

7. (10points) List the actors involved in the service life cycle and its stages. What constitutes an offering?

ANS.

The three entities involved here are the Cloud service provider, the Cloud service developer, and the Cloud service consumer.

First, suppose the instances for a load balancer and two servers have been created successfully, but creating the virtual machine for the third server has failed. What should the user program do? Deleting all other instances and restarting again is hardly an efficient course of action for the following reasons. From the service developer's point of view, this would greatly complicate the program (which is supposed to be fairly simple). From the service provider's point of view, this would result in wasting the resources which were first allocated and then released but never used.

Second, assuming that all instances have been created, a service provider needs to support elasticity. The question is: How can this be (a) specified and (b) effected? Suppose each of the three servers has reached its threshold CPU utilization. Then a straightforward solution is to create yet another instance (which can be deleted once the burst of activity is over), but how can all this be done automatically? To this end, perhaps, maybe not three but only two instances should have been created in the first place.

The solution adopted by the industry is to define a service in more general terms (we will clarify this with examples), so that the creation of a service is an atomic operation performed by the service provider— this is where orchestration first comes into the picture. And once the service is deployed, the orchestrator itself will then add and delete instances (or other resources) as specified in the service definition.

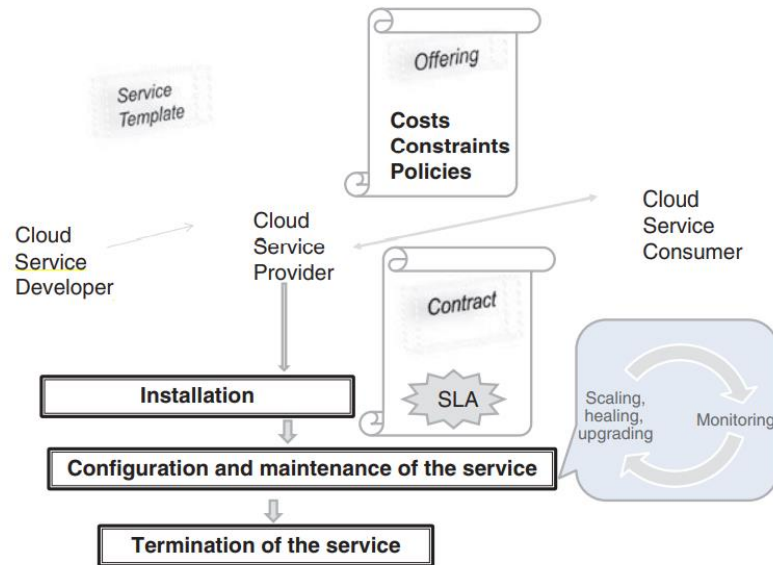


Figure 7.8 The service life cycle.

(References: Textbook: Cloud Computing: Business Trends and Technologies)

8. (10 points) What is the name of the protocol used for communications among the OpenStack daemons?

ANS.

OpenStack services use advanced message queuing protocol (AMQP), an open standard for messaging middleware. This messaging middleware enables the OpenStack services that run on multiple servers to talk to each other. OpenStack Oslo RPC supports two implementations of AMQP: RabbitMQ and ZeroMQ. Communications among daemons are carried out via the Advanced Message Queuing Protocol (AMQP). The message queue of Figure 7.11 is the structure that enables this.

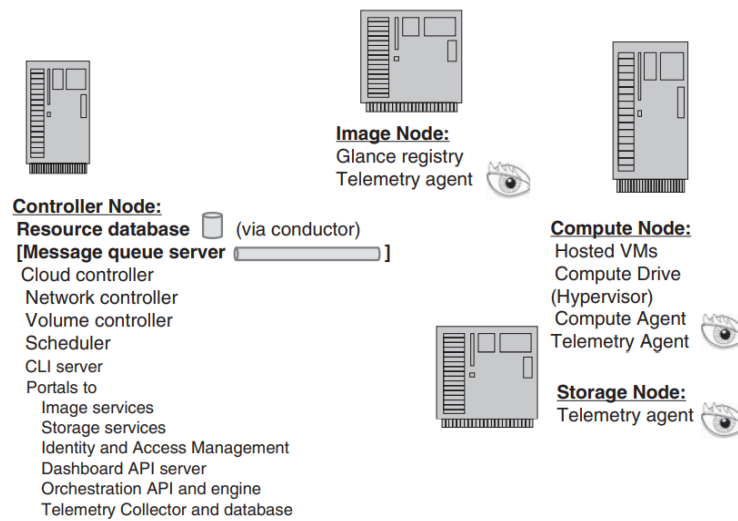


Figure 7.11 Mapping the OpenStack components into a physical architecture: an example.

The Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing (including point-to-point and publish-and-subscribe), reliability and security. AMQP can be initiated from either end of the pipe. On the contrary, an HTTP transaction can be initiated only by the client because HTTP is a pure client/server protocol.

(References: Textbook: Cloud Computing: Business Trends and Technologies, [Advanced Message Queuing Protocol - Wikipedia](#))