November 2022

ADAPTIVE SERVICE TESTING FOR OBSERVABILITY

John Monaghan

ADAPTIVE SERVICE TESTING FOR OBSERVABILITY

AUTHORS:
John Monaghan

ABSTRACT

Described herein are techniques for optimizing network and application tests based on a prioritization scheme. These techniques may minimize costs and network overhead while reducing complexity for operations teams.

DETAILED DESCRIPTION

Techniques exist to allow application and network services to be tested from various vantage points. For example, software agents can be deployed on hosts such as user devices, network nodes, virtual machines, and cloud Points of Presence (PoPs). They inject synthetic traffic to measure various performance metrics such as packet loss and latency.

However, there is no mechanism to adjust these tests to cater for those that may already be running from the local network, from a colleague's machine in the same building, or the nearest PoP. This is inefficient both commercially and technically.

Accordingly, techniques are provided for optimizing network and application tests based on a prioritization scheme. These techniques may involve a Management Platform (MP) which is aware of and controls all testing and agent deployments.

As described herein, application and network service tests may be implemented using software agents deployed on hosts. These hosts may include user devices such as laptops; network resources such as switches and routers; and compute resources such as Virtual Machines (VMs).

The MP uses a Location Service (LS) which can return the approximate physical location of hosts. The LS may derive the location from the hosts directly (via their operating systems) or from a dedicated positioning system. The MP can also access the network topology where hosts are located. This may come from a separate tool.

Agents may be classified as either Endpoint or Network. The former run on end user devices, and the latter can run anywhere in the network (e.g., dedicated VM, hosted on a switch, etc.). There is a hierarchical relationship in terms of agent priority. This may be determined by a combination of agent type and position in the network topology. In terms of agent priority, Network may be higher than Endpoint. Meanwhile, network position will be determined based on where in the logical and physical topology the Agent is hosted. For example, a design for a Site (e.g., a building) may follow the classic topology of customer premises equipment (e.g., a Wide Area Network (WAN) router) followed by core, distribution, access, and then endpoint layers. Thus, a Network agent deployed on a core switch has a higher priority than one deployed on an Access switch and an Endpoint agent deployed on a laptop. But, it would have a lower priority than a Network agent deployed on a CPE router. For clarity, the same test running on a Network Agent will supersede that running on an Endpoint Agent.

Hosts are located at a Site. A site may be a private location (e.g., a home), a mobile location (e.g., a cafe or a hotel) or a fixed corporate location (e.g., an office or a campus). Sites may be created and managed in the MP.

The association between Agent and Site is dynamic since Hosts can change locations. Via the LS, the MP is updated regarding the association between the Host and the Site.

The Agents are configured by the MP to test a variety of application and network services. There are many possible test types. For example, Endpoint Agents on user laptops may be configured to test a standard application every thirty minutes for latency, whereas the main site router might be configured to test custom applications, or network services such as Domain Name System (DNS) and Border Gateway Protocol (BGP) route availability.

The combination of Host, Test, Type, and Frequency is called a Regime, and is managed in the MP via a Graphical User Interface (GUI) or Application Programming Interface (API).

It is possible, and indeed quite likely, that Agents running on different Hosts may in fact be testing the same service. This should be avoided because tests incur penalties both economically and in terms of overhead.

The likelihood of overlapping test regimes is increased because of the mobility and flexible working patterns of employees. Consider the scenario of employees that have laptops running Endpoint Agents configured to test an application who come into one of their company offices to work. If the main gateway router is already configured to test that same application, then there is some overlapping testing occurring and the Administrator may want to remove the Endpoint Agent Application tests, or reduce the number of those tests.

The techniques described herein may be implemented using the following algorithm:

- Agents are deployed by the MP or dedicated software management tools such as Group Policy Objects (GPO), Mobile Device Management (MDM), etc.
- All agent types register and check-in to the MP at configurable intervals.
- At check-in, the MP extracts location data from the LS and uses it to associate the Agent with its current Site.
- The MP also calculates and updates each Agent priority using Agent Type and position in the network (which may be derived from a network management tool).
- Because of the periodic check-in, new locations are detected and re-associated to Sites if necessary and Agent priorities may be continually updated.

When an agent checks in to the MP, the associated Site is identified. The MP then collects all other agents associated to the same site and determines which tests are being run from all these Agents. Because there is an Agent Priority, the MP may automatically switch off all duplicate tests running from agents with a lower priority or advise the administrator of the inefficiency via a GUI or API.

Techniques described herein may address the dynamic nature of networking systems and the network on which it is based (e.g., the influx/efflux of active Agents caused by mobile workers coming into/out of an office, where tests are already running, e.g., on switches or on colleagues' machines). Similarly, tests may be introduced manually via other management systems or applications.

Inefficiencies may be captured in active Agents within a network, not just profiles that may or may not be deployed. For example, when two or more Agents are testing the same service from the same network, these techniques may detect such duplication and expose it to an administrator or via external messaging.

Figure 1 below illustrates an example system architecture.

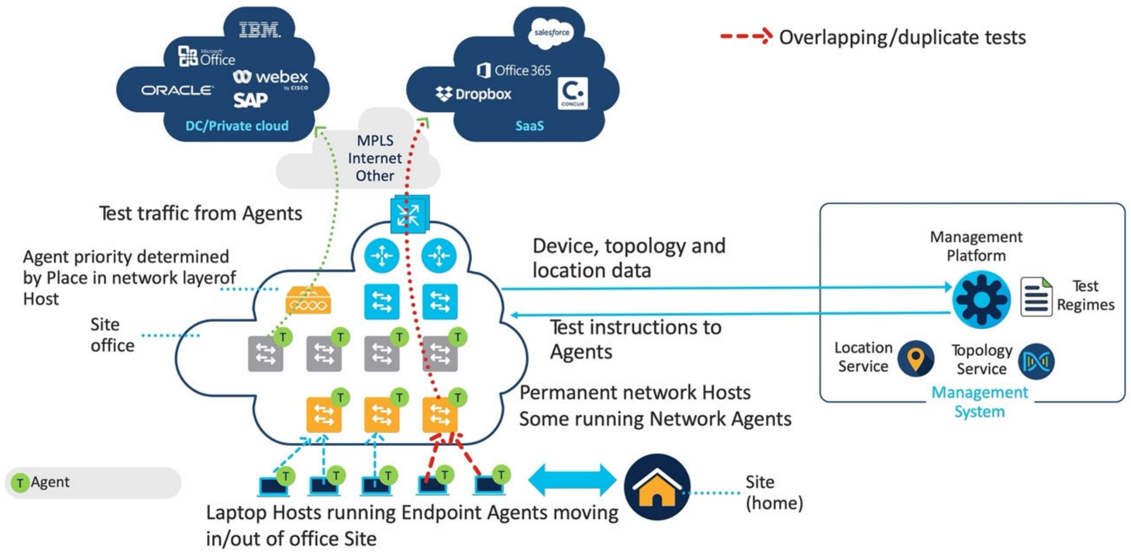

Figure 1

In summary, described herein are techniques for optimizing network and application tests based on a prioritization scheme. These techniques may minimize costs and network overhead while reducing complexity for operations teams.