Lumen IoD Bandwidth Scheduler

Demo Tool

Jacob Johansen

1

Lumen IoD Bandwidth Scheduler overview

To drive automated updates to Lumen Internet On-Demand connections, ***leverage the power of an existing scheduling engine*** and similarly ***meet customers in a familiar environment*** they already use.

In this demo tool, a Microsoft 365 calendar is used to drive the IoD bandwidth events. However, the concept could be extended to any third-party calendar solution that has a public API endpoint.

While this demo app uses a *pull* model to retrieve calendar events on a periodic basis, customers could build their own solutions to integrate more deeply into third-party solutions, potentially on a event-based push model.

3

Lumen IoD Bandwidth Scheduler setup

Dependences:

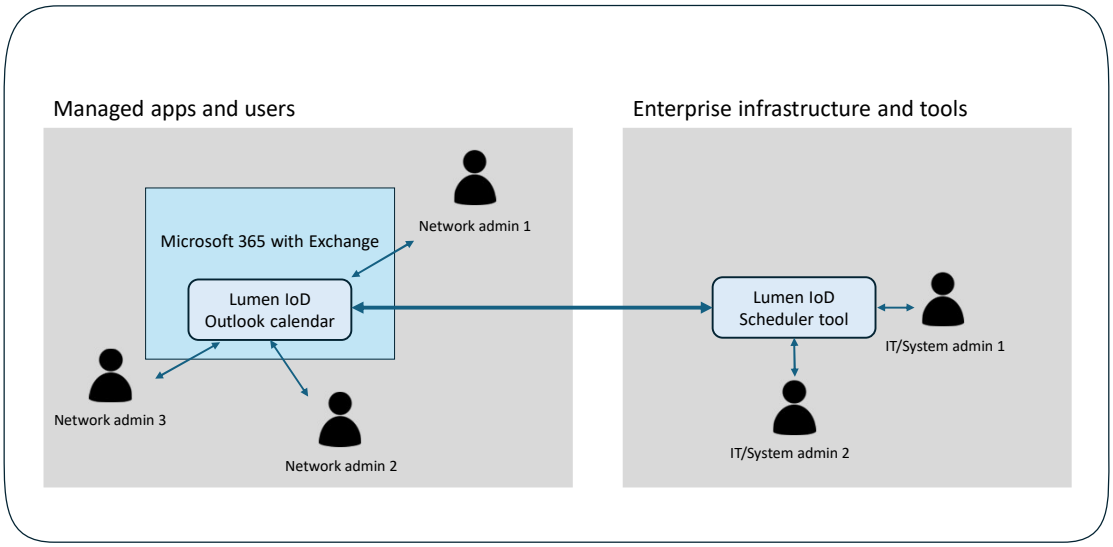
- Microsoft 365 license for work/school (Personal account can be used for testing)

Configuration:

1. Microsoft 365 calendar created and provisioned to specified users
2. Registered application in Azure Entra ID to allow calendar access through Microsoft Graph APIs:
   - Including client secret and Graph API permissions (Calendars.Read and User.Read.All)
3. Registered application in Lumen Developer Center for IoD
   - Including client secret
4. Configured process to run Lumen IoD Scheduler on recurring basis (such as through Windows Task Scheduler, Windows services, Kubernetes cronjob, etc.)

4

Organization boundary



The demo app is a Windows Console app, but could be adopted to be run in a container, run on a VM, etc.

5

Lumen IoD Bandwidth Scheduler examples

6

Calendar event format

For this tool, the following syntax is required for calendar events:

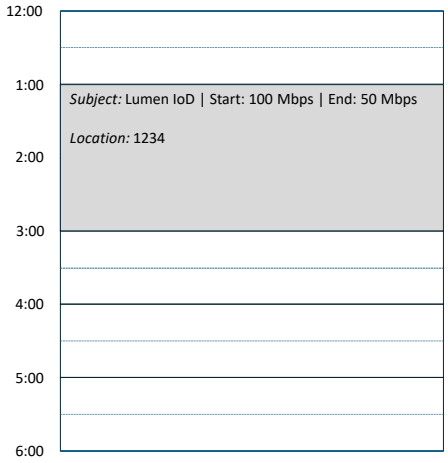
- Subject line of a (configurable) format. Default value:
  - "Lumen IoD | Start: <#> Mbps | End: <#> Mbps" where # is the value such as 100, 50, 1000
- Location:
  - Value represents the Service Id of the IoD connection
- Category tagging:
  - The *count* of the Categories set on a calendar event reflects the priority of the event; The more categories, the higher the priority

7

Calendar event examples

Actions:

- None

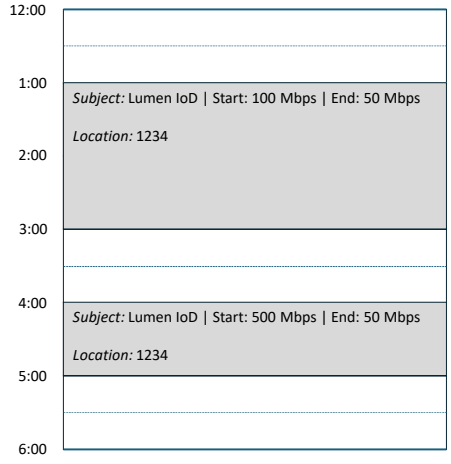


Actions:

- Bandwidth set at 1 pm to 100 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 50 Mbps for Service Id 1234

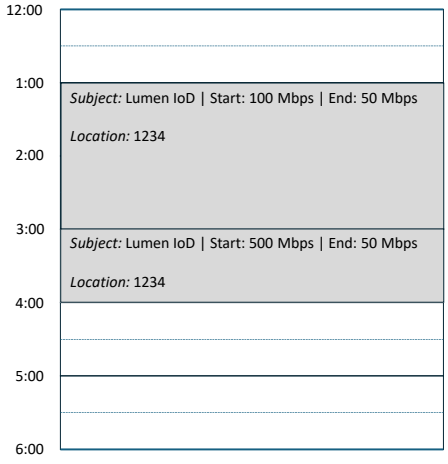
8

Calendar event examples



Actions:

- Bandwidth set at 1 pm to 100 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 50 Mbps for Service Id 1234
- Bandwidth set at 4 pm to 500 Mbps for Service Id 1234
- Bandwidth set at 4 pm to 50 Mbps for Service Id 1234

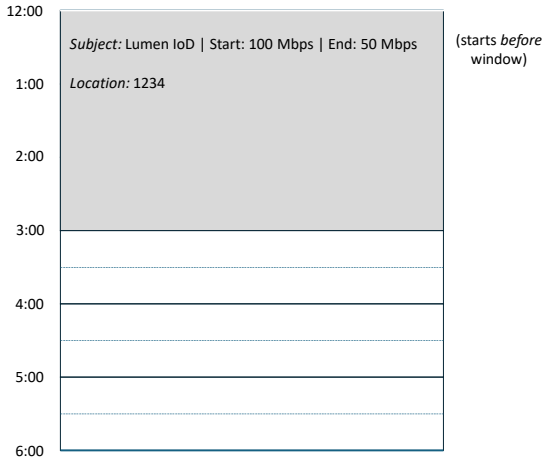


Actions:

- Bandwidth set at 1 pm to 100 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 500 Mbps for Service Id 1234
- Bandwidth set at 4 pm to 50 Mbps for Service Id 1234

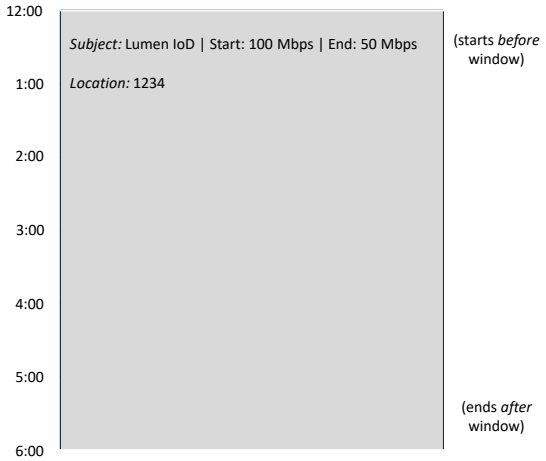
9

Calendar event examples



Actions:

- Bandwidth set at 3 pm to 50 Mbps for Service Id 1234

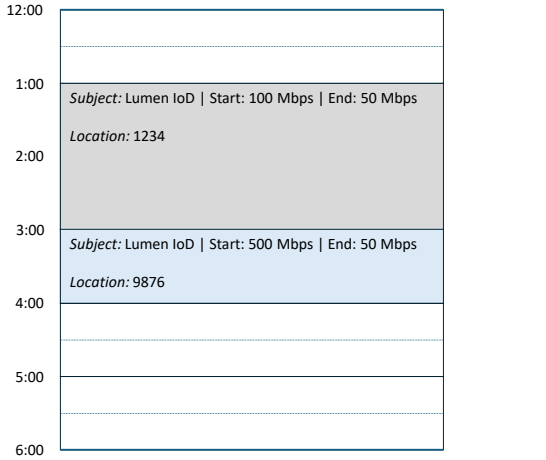


Actions:

- None

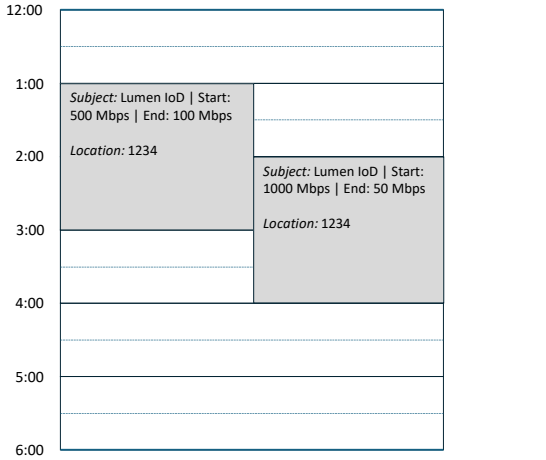
10

Calendar event examples



Actions:

- Bandwidth set at 1 pm to 100 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 50 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 500 Mbps for Service Id 9876
- Bandwidth set at 4 pm to 50 Mbps for Service Id 9876



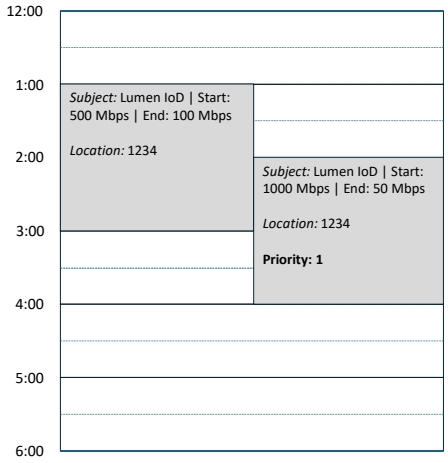
Actions:

- Bandwidth set at 1 pm to 500 Mbps for Service Id 1234
- Bandwidth set at 3 pm to 100 Mbps for Service Id 1234
- Bandwidth set at 4 pm to 50 Mbps for Service Id 1234

(Ignores conflicting start time for same Service Id when the priority is the same.)

11

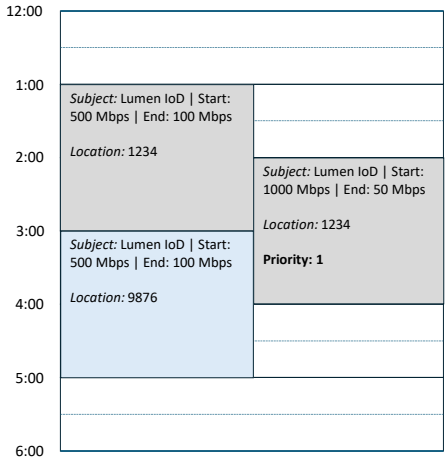
Calendar event examples



Actions:

- Bandwidth set at 1 pm to 500 Mbps for Service Id 1234
- Bandwidth set at 2 pm to **1000 Mbps** for Service Id 1234
- Bandwidth set at 4 pm to 50 Mbps for Service Id 1234

(Higher priority conflict on same Service Id will win.)



Actions:

- Bandwidth set at 1 pm to 500 Mbps for Service Id 1234
- Bandwidth set at 2 pm to **1000 Mbps** for Service Id 1234
- Bandwidth set at 3 pm to 500 Mbps for Service Id 9876
- Bandwidth set at 4 pm to 50 Mbps for Service Id 1234
- Bandwidth set at 5 pm to 100 Mbps for Service Id 9876

(Higher priority conflict on same Service Id will win.)

12