insight.tech

POWERING IoT AND EMBEDDED INNOVATION

IoT Simulation: The Test for Success

Simulators are used across every aspect of life to test a facsimile of real environments. They are vital to predicting the actions and reactions of complex systems—from flight simulators and movie sets, to weather models and economic forecasts. The IoT is no different.

The complexity of deployments and the number of IoT devices in a typical network are on the rise. This makes the chance of an unanticipated problem that could wreak havoc more likely. The flexibility and scalability that make IoT deployments attractive also make them difficult to test at scale. This is one of the complications that can doom a new project.

A Cisco report states that 60 percent of IoT projects stall at the proof-of-concept stage. Only a fraction are considered a complete success. Reports like this speak to the difficulty of moving from a system that works on paper to one that achieves its full potential. Simulating various components of the network before bringing the entire system online is one of the best ways to ensure a successful project.

Simulation is important when the number of connected devices or their distribution grows beyond a few rooms or handful of products. And it's far more cost-effective to simulate an extensive network than it is to actually build one. Testing rare scenarios, fault tolerance evaluation, or measuring end-to-end latency are additional use cases.

The chance of encountering an unexpected event grows as the number of sensors increases. By simulating these events in advance, operators can ensure they are dealt with appropriately.

MQTT Protocol: IoT Friendly

Simulators come in all shapes and sizes—both commercial software and open-source alternatives. They can be differentiated by a number of factors such as scalability and supported protocols.

One popular protocol is MQTT (Message Queuing Telemetry Transport), a publish-subscribe messaging system. It is prevalent for IoT networks because it's lightweight, with a small code footprint. It also works well in areas where bandwidth is limited.

Applications like the MIMIC MQTT Simulator from Gambit Communications are designed to create a large environment of sensors, actuators, and other MQTT clients.

An IoT simulator can focus on a specific messaging protocol, like MQTT, or support multiple protocols, including CoAP, REST, and AMQP. Picking the right one requires first understanding what kind of simulation is needed and which aspect of the total solution is under test.

Simulating the architectural behavior of individual endpoints is different from that of a client-broker model. The MQTT standard relies on this model for passing messages across a network. Gambit treats the MQTT as an interface to be simulated. This enables it to effectively represent any device that uses the protocol and then scale up the total number to whatever level is required.

Simulation at Scale

In some environments a simulator can help you understand subtle differences in how protocol support has been implemented by various brokers.

When Gambit tested MQTT 5.0 support with a number of services, it found several scenarios that could have caused unexpected behavior in a production environment. Testing the applications against a simulator before deployment helps prevent unwelcome surprises.

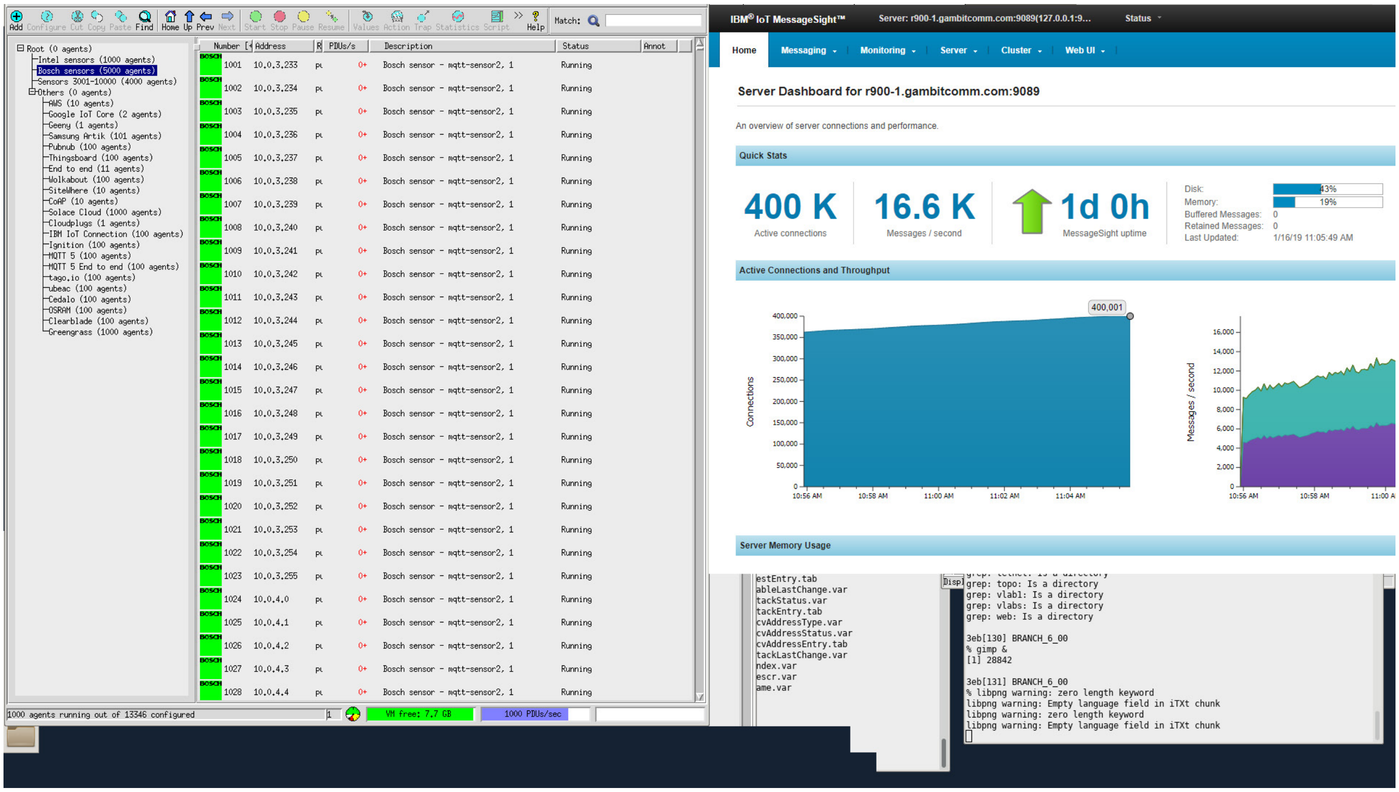
These components can be programmed to interact in different ways, controlled by the end user. Operators can scale simulations up and down, program specific events, and measure the impact of these changes on the platform under test in real time.

The MIMIC Simulator is capable of positioning hardware based on the Intel® IoT Gateway platform as a Device Under Test (DUT), and can create thousands of virtual sensors. Each sensor can be defined as a different type, such as temperature, pressure, and location. They can also be programmed to predictably generate unique traffic, or to simulate various failure modes.

MIMIC supports dynamic rules that can adjust a simulation mid-run and introduce new parameters. This is important for evaluating how an analytics engine behaves under real-world conditions when confronted with unexpected behavior. This could be malfunctioning sensors, sudden failure of a group of sensors, or radio interference. Tools that support only static rules can't be adjusted to test these kinds of scenarios **(Figure 1)**.

Performance Testing and Edge-Case Evaluation

The Gambit solution can be used to examine how other IoT components perform as the number of connected devices grows. Can the customer's analytics engine scale effectively to meet current or future demand? Can it handle unexpected corner cases in an appropriate way?



```
⊟ Root (0 agents)
  ├Intel sensors (1000 agents)
  ├Bosch sensors (5000 agents)
  ├Sensors 3001-10000 (4000 agents)
  ⊟Others (0 agents)
    ├AWS (10 agents)
    ├Google IoT Core (2 agents)
    ├Geeny (1 agents)
    ├Samsung Artik (101 agents)
    ├Pubnub (100 agents)
    ├Thingsboard (100 agents)
    ├End to end (11 agents)
    ├Wolkabout (100 agents)
    ├SiteWhere (10 agents)
    ├CoAP (10 agents)
    ├Solace Cloud (1000 agents)
    ├Cloudplugs (1 agents)
    ├IBM IoT Connection (100 agents)
    ├Ignition (100 agents)
    ├MQTT 5 (100 agents)
    ├MQTT 5 End to end (100 agents)
    ├tago.io (100 agents)
    ├ubeac (100 agents)
    ├Cedalo (100 agents)
    ├OSRAM (100 agents)
    ├Clearblade (100 agents)
    └Greengrass (1000 agents)
```

Number [	Address	R	PDUs/s	Description	Status	Annot
BOSCH 1001	10.0.3.233	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1002	10.0.3.234	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1003	10.0.3.235	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1004	10.0.3.236	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1005	10.0.3.237	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1006	10.0.3.238	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1007	10.0.3.239	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1008	10.0.3.240	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1009	10.0.3.241	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1010	10.0.3.242	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1011	10.0.3.243	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1012	10.0.3.244	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1013	10.0.3.245	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1014	10.0.3.246	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1015	10.0.3.247	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1016	10.0.3.248	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1017	10.0.3.249	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1018	10.0.3.250	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1019	10.0.3.251	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1020	10.0.3.252	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1021	10.0.3.253	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1022	10.0.3.254	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1023	10.0.3.255	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1024	10.0.4.0	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1025	10.0.4.1	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1026	10.0.4.2	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1027	10.0.4.3	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	
BOSCH 1028	10.0.4.4	pι	0+	Bosch sensor - mqtt-sensor2, 1	Running	

1000 agents running out of 13346 configured | 1 | VM free: 7.7 GB | 1000 PDUs/sec

Server Dashboard for r900-1.gambitcomm.com:9089

An overview of server connections and performance.

Quick Stats

400 K Active connections

16.6 K Messages / second

1d 0h MessageSight uptime

Disk: 43%
Memory: 19%
Buffered Messages: 0
Retained Messages: 0
Last Updated: 1/16/19 11:05:49 AM

Active Connections and Throughput

Server Memory Usage

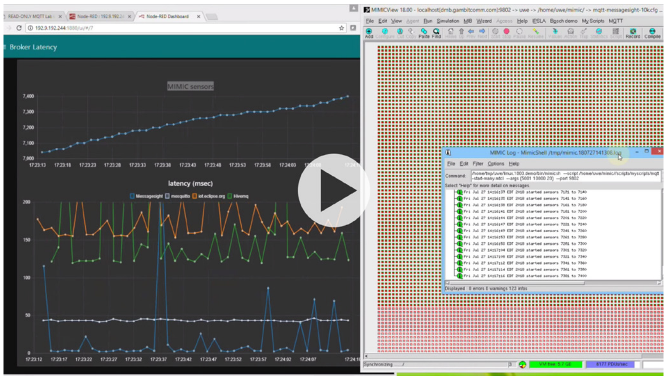
```
estEntry.tab
ableLastChange.var
tackStatus.var
tackEntry.tab
cvAddressType.var
cvAddressStatus.var
cvAddressEntry.tab
tackLastChange.var
ndex.var
escr.var
ame.var
```

```
grep: topo: Is a directory
grep: vlabl: Is a directory
grep: vlabs: Is a directory
grep: web: Is a directory

3eb[130] BRANCH_6_00
% gimp &
[1] 28842

3eb[131] BRANCH_6_00
% libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
libpng warning: zero length keyword
libpng warning: Empty language field in iTXt chunk
```

Figure 1. MIMIC MQTT Simulator in action, with 400K connections and 16,600 messages arriving each second.

Simulating different arrangements of publishers and subscribers or significantly increasing the total number of connections are good ways to benchmark the flexibility and performance of an IoT solution.

The video below **(Video 1)** shows the effect on end-to-end latency when 10,000 sensors are being brought online.



```
/home/tmp/uwe/linux.1000.demo/bin/mimicsh  --script /home/uwe/mimic//scripts/myscripts/mqtt
-start-many.mtcl  --args (5001 10000 20)  --port 9802
Select "Help" for more detail on messages.
Fri Jul 27 14:16:32 EDT 2018 started sensors 7121 to 7140
Fri Jul 27 14:16:35 EDT 2018 started sensors 7141 to 7160
Fri Jul 27 14:16:38 EDT 2018 started sensors 7161 to 7180
Fri Jul 27 14:16:41 EDT 2018 started sensors 7181 to 7200
Fri Jul 27 14:16:44 EDT 2018 started sensors 7201 to 7220
Fri Jul 27 14:16:47 EDT 2018 started sensors 7221 to 7240
Fri Jul 27 14:16:49 EDT 2018 started sensors 7241 to 7260
Fri Jul 27 14:16:53 EDT 2018 started sensors 7261 to 7280
Fri Jul 27 14:16:58 EDT 2018 started sensors 7281 to 7300
Fri Jul 27 14:17:04 EDT 2018 started sensors 7301 to 7320
Fri Jul 27 14:17:08 EDT 2018 started sensors 7321 to 7340
Fri Jul 27 14:17:12 EDT 2018 started sensors 7341 to 7360
Fri Jul 27 14:17:14 EDT 2018 started sensors 7361 to 7380
Fri Jul 27 14:17:16 EDT 2018 started sensors 7381 to 7400
Displayed : 0 errors 0 warnings 123 infos
```

Video 1. MIMIC simulation video.

As the number of sensors increases, so does the total (simulated) traffic load. Overall latency begins to creep up. A number of variables can impact latency. These include the distance between sensors and brokers, the message payload size, QoS settings, and more. An IoT simulator should provide the ability to adjust these settings as needed.

The IoT, as an industry, is still in its infancy. Simulators were initially designed for cloud testing, because that's where the majority of IoT processing was happening. The shift to edge computing is driving the need for IoT simulators to add features that integrate with edge services. Gambit has focused on adding these capabilities to its MQTT simulator to ensure this growing test area is also thoroughly covered.