

Open in app ↗

Sign up

Sign in

## Medium

 Search

# Mosquito Bridge

## How To Bridge Two Mosquitto Brokers — MQTT — Episode # 06

5 min read · Dec 28, 2020



J3

Follow



Listen



Share

Mosquitto has a feature called bridging which basically lets you connect two (or more) brokers together.

Here are our bridge configurations(*br-me-to-broker0*):

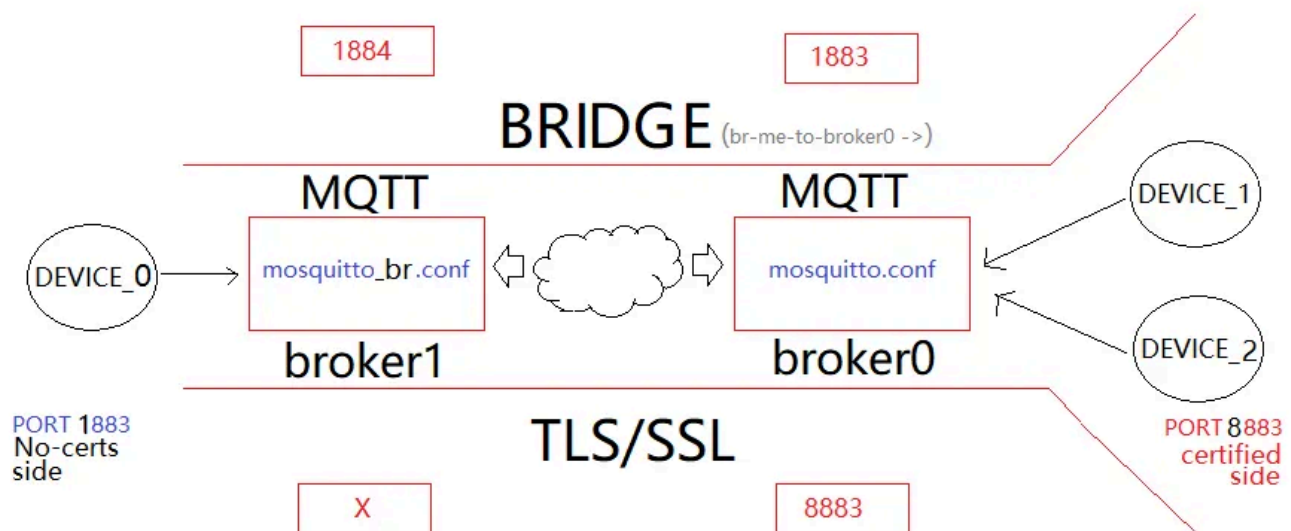


Fig 1. Our bridge set up. The default **mosquito.conf** file will be our **broker0** and a modified **broker1** **mosquitto\_br.conf** file will implement our **br-me-to-broker0** (see that tunneling is extended on the right side) — by working like this we can have devices (0,1,2) spread around the world. A true international sensor network! Can you imagine how cool is it? Cool! Very nice! This is where the fun happens!

We will open two instances of the brokers, *broker0*, and *broker1*;

*The broker1* will be responsible for implementing the bridge solution (for this we will modify the *conf* file at the *BRIDGE* session — see Table 1 below);

To fully understand this lesson, you can download the two configuration files from [my google drive](#) and open them in your [notepad++](#).

I'm sure it will worth the effort!

Let's get it on!

**01** #Step — For **broker0**, Open *conf* file in your [notepad++](#) and edit theses lines in *mosquitto.conf* (assumed you've come from the [last](#) MQTT episode):

#### Line#Description

210 # Port to use for the default listener.

211 **port 1883**

...

310 **tls\_version tlsv1.2**

...

373 # listener port-number [ip address/host name]

374 **listener 8883**

...

431 # Certificate based SSL/TLS support453 # Path to the PEM encoded server certificate.

...

450 **cafile** C:\Program Files\mosquitto\certs\ca.crt

...

454 **certfile** C:\Program Files\mosquitto\certs\server.crt

...

457 **keyfile** C:\Program Files\mosquitto\certs\server.key

...

651 **allow\_anonymous false**

...

669 **password\_file** C:\Program Files\mosquitto\passwordfile.pwd

...

728 **acl\_file** C:\Program Files\mosquitto\acl.acl

In summary: the default configuration plus TLS/SSL; no anonymous client and [ACL](#) rules, as usual, nothing fancy, right? (you can download this file from [my google drive](#) right away:).

Please refer to the [second](#) and [third](#) MQTT episodes to fully understand it all:)

Fine! So far, so good!

# 02 #Step — For **broker1**, Copy/paste *mosquitto.conf* and save it as *mosquitto\_br.conf* (this will implement the **bridge**, named *br-me-to-broker0*);

Here is the configurations **line by line** (do not worry about details now; in Table 1 I'll explain everything — just get it running in front of you!):

## Line#Description

```

210 # Port to use for the default listener.
211 port 1884
...
310 tls_version tlsv1.2
...
373 # listener port-number [ip address/host name]
374 #listener
...
431 # Certificate based SSL/TLS support453 # Path to the PEM encoded
server certificate.
...
450 #cafile
...
454 #certfile
...
457 #keyfile
...
651 allow_anonymous false
...
669 password_file C:\Program Files\mosquitto\passwordfile.pwd
...
728 acl_file C:\Program Files\mosquitto\acl.acl
...
755# =====
756 # Bridges
757# =====
...
792 connection br-me-to-broker0
793 address LAPTOP-JAYTHREE:8883
794 topic # both 0
...
817 cleansession false
...
839 notifications false
...
851 remote_clientid broker0
...
857 remote_password 123
...
863 remote_username admin
...
909 start_type automatic
...
922 try_private true

```

```
...  
936 bridge_cafile C:\Program Files\mosquitto\certs\ca.crt  
...  
952 bridge_insecure false  
...  
955 bridge_certfile :\\Program Files\mosquitto\certs\server.crt  
...  
958 bridge_keyfile C:\Program Files\mosquitto\certs\server.key
```

Save *mosquitto\_br.conf* file. Now testing...

## 03 #Step — For the test, open 6 prompts Terminals (2 of them as Administrator:)

Come on, it is a piece of cake!

At Terminal #1 (as admin), type:

```
mosquitto -c mosquitto.conf -v
```

Mosquitto will listen on Ports 1883 & 8883.

Fine!

At Terminal #2 (as admin), type:

```
mosquitto -c mosquitto_br.conf -v
```

The mosquitto will listen only on Port 1884 and, internally, by DN LAPTOP-JAYTHREE:8883 connect to the bridge. See Fig 2:

```

Administrador: Prompt de Comando - mosquitto -c mosquitto.conf -v
C:\Program Files\mosquitto>mosquitto -c mosquitto.conf -v
1609160773: mosquitto version 1.6.8 starting
1609160773: Config loaded from mosquitto.conf.
1609160773: Opening ipv6 listen socket on port 8883.
1609160773: Opening ipv4 listen socket on port 8883.
1609160773: Opening ipv6 listen socket on port 1883.
1609160773: Opening ipv4 listen socket on port 1883.
1609160776: New connection from fe80::4c69:f2a5:6566:8eb8 on port 8883.
1609160776: New bridge connected from fe80::4c69:f2a5:6566:8eb8 as broker0 (p2, c0, k60, u'admin').
1609160776: No will message specified.
1609160776: Sending CONNACK to broker0 (0, 0)
1609160776: Received SUBSCRIBE from broker0
1609160776:      # (QoS 0)
1609160776: broker0 0 #
1609160776: Sending SUBACK to broker0

Administrador: Prompt de Comando - mosquitto -c mosquitto_br.conf -v
C:\Program Files\mosquitto>mosquitto -c mosquitto_br.conf -v
1609160776: mosquitto version 1.6.8 starting
1609160776: Config loaded from mosquitto_br.conf.
1609160776: Opening ipv6 listen socket on port 1884.
1609160776: Opening ipv4 listen socket on port 1884.
1609160776: Bridge local.broker0 doing local SUBSCRIBE on topic #
1609160776: Connecting bridge br-me-to-broker0 (LAPTOP-JAYTHREE:8883)
1609160776: Bridge broker0 sending CONNECT
1609160776: Received CONNACK on connection local.broker0.
1609160776: Bridge local.broker0 sending SUBSCRIBE (Mid: 1, Topic: #, QoS: 0, Options: 0x00)
1609160776: Received SUBACK from local.broker0

```

Fig 2. Now the two brokers are connected \o/

Watch out for the initialization message on both brokers;

The broker1 will connect to broker0 through the bridge named *br-me-to-broker0* (via LAPTOP-JAYTHREE:8883) and will subscribe to all topics at broker0 (Fig 2 red frames).

Let's publish and subscribe topic *temperature* with a common *user1* (contained inside *acl.acl* file):

On Terminal #3 (subscription — broker0 to broker1 way):

```
mosquitto_sub -h localhost -p 1884 -u user1 -P 321 -t temperature
```

On Terminal #4 (publishing):

```
mosquitto_pub -h LAPTOP-JAYTHREE -p 8883 -u user1 -P 321 --cafile
ca.crt -t temperature -m 47
```

## On Terminal #5 (subscription — broker1 to broker0 way):

```
mosquitto_sub -h LAPTOP-JAYTHREE -p 8883 -u user1 -P 321 --cafile
ca.crt -t temperature
```

## On Terminal #6(publishing):

```
mosquitto_pub -h localhost -p 1884 -u user1 -P 321 -t temperature -m
48
```

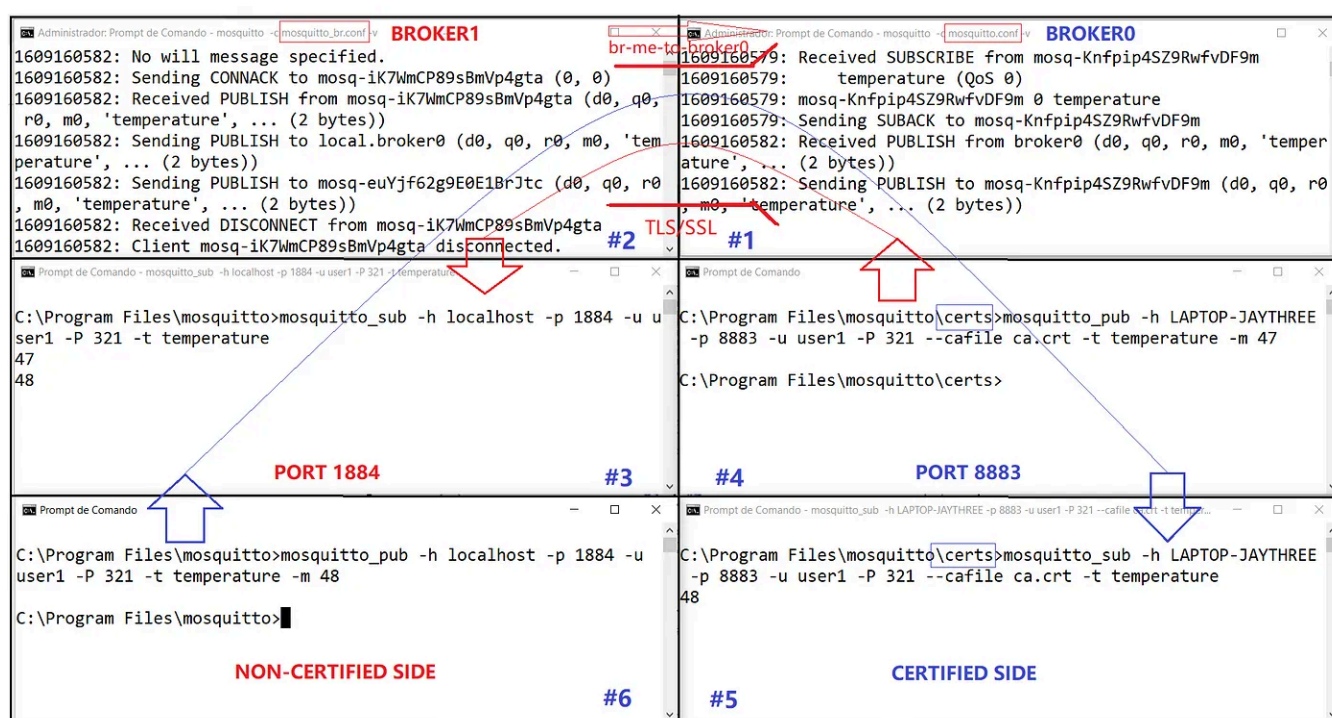


Fig 3. Here is the final result. Huge success! (To access the certifications we need to be at \certs directory)

Tested and approved in both directions; both the certified and non-certified sides send published messages and the other side receives all messages perfectly.

Our bridge is working fine!

Although the `mosquitto_br.conf` instance disables the secure port (8883, no Extra listener), it does use it in the bridge (via LAPTOP-JAYTHREE:8883); this instance is responsible for loading all communication certificates that will be used between the brokers. This fact is very curious about this solution!

Summary table:

Search this file...

Line	Properties	broker1 Values (mosquitto.br_conf)	broker0 Values (mosquitto.conf)	
210	port	1884	1883	P
310	tls_version	tlsv1.2	tlsv1.2	tl
374	listener	-	8883	li
450	cafile	-	C:\...\mosquitto\certs\ca.crt	d
454	certfile	-	C:\...\mosquitto\certs\server.crt	P
457	keyfile	-	C:\...\mosquitto\certs\server.key	P
651	allow_anonymous	false	false	lf
669	password_file	C:\...\mosquitto\passwordfile.pwd	C:\...\mosquitto\passwordfile.pwd	lf
728	acl_file	C:\...\mosquitto\acl.acl	C:\...\mosquitto\acl.acl	lf
792	connection	br-me-to-broker0	-	A
793	address	LAPTOP-JAYTHREE:8883	-	a
794	topic	# both 0	-	<
817	cleansession	false	-	V
839	notifications	false	-	lf
851	remote_clientid	broker0	-	S
857	remote_password	123	-	S
863	remote_username	admin	-	S
909	start_type	automatic	-	A
922	try_private	true	-	lf
936	bridge_cafile	C:\...\mosquitto\certs\ca.crt	-	b
952	bridge_insecure	false	-	V
955	bridge_certfile	C:\...\mosquitto\certs\server.crt	-	P
958	bridge_keyfile	C:\Program Files\mosquitto\certs\server.key	-	P
-----	-----	-----	-----	-

Table 1. Summary of the two config files for your convenience ;-)

I hope you enjoy this experiment!

It takes all day to bring it to you on a silver tray:)

That's all for now.

Thanks!

Bye!

[Download All Files For This Project](#)

Summary (cmds)

```
mosquitto -c mosquitto.conf -v  
mosquitto -c mosquitto_bd.conf -v  
mosquitto_sub -h localhost -p 1884 -u user1 -P 321 -t temperature  
mosquitto_pub -h LAPTOP-JAYTHREE -p 8883 -u user1 -P 321 --cafile  
ca.crt -t temperature -m 47  
mosquitto_sub -h LAPTOP-JAYTHREE -p 8883 -u user1 -P 321 --cafile  
ca.crt -t temperature  
mosquitto_pub -h localhost -p 1884 -u user1 -P 321 -t temperature -m  
48
```

## Related Posts

**01** # Episode — Mosquitto — [Intro To MQTT](#) — It is Suitable for the Internet of Things Applications — MQTT

**02** # Episode — Mosquitto — [User Access Configurations Setups — Editing mosquitto.conf File to Configure SSL Authentications](#) — MQTT

**03** # Episode — Mosquitto — [ACLs — Wildcards & ACL — access control lists](#) — MQTT

**04** # Episode — Mosquitto — [MQTT QoS](#) — How To Set QoS at Mosquitto Broker — MQTT

**05** # Episode — Mosquitto — [Bulletproof TLS & SSL Mosquitto](#) — How To Set Up Mosquitto Broker/Client Keys & Certificates — MQTT

**06** # Episode — Mosquitto — [Mosquitto Bridge](#) — How To Bridge Two Mosquitto Brokers — MQTT(this one)

07...be tuned for the upcoming post about MQTT and IoT o/

## Credits & References

[Microgênios — Treinamento em Sistemas Embarcados — Microchip Regional Partner](#) — Microchip Certified Brazilian Training Education Company & A Simpício-Owned-Awesome Enterprise o/

[Bridging by owntracks.org](#)