500 bytes - Various clients at 6 second intervals (1/6 Hz)

This entire set of tests was performed with ***20*** attack bees, so the warheads are somewhat different than in other test pages. Remember that a warhead of 100 processes will make 100 x <number of attack bees> total clients

This entire set of tests was performed with a *beta* preview version of mosquitto 1.2.3, specifically, mercurial revision: https://bitbucket.org/oojah/mosquitto/commits/d36ae5e2d9fa

```
# In a virtualenv with mqtt-malaria installed
$ fab beeup:20,region=us-east-1,ami=ami-a53264cc
$ fab -i ~/.ssh/karl-malaria-bees.pem mstate aptup
$ fab -i ~/.ssh/karl-malaria-bees.pem mstate everybody:True
$ fab -i ~/.ssh/karl-malaria-bees.pem mstate deploy
$ fab -i ~/.ssh/karl-malaria-bees.pem mstate share_key:malaria.pskfile # or whatever
keyfile you are using....
```

500 bytes / 500 clients at regular 1/6 Hz rate

This is a very low load for this machine.

Command:

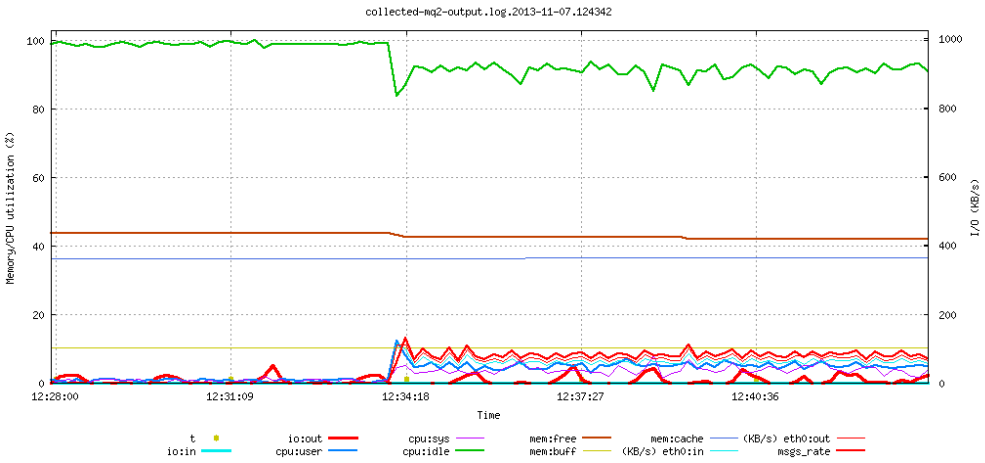
```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:zzz.xxxxxx.com,warhead=warheads/25x0.166mps--1000x500bytes.wh
```

Anecdotally

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
9-11% (single)	15%	250/2000.	fluctuating around 85, levelling out

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'
$SYS/broker/clients/active 504
$SYS/broker/load/messages/received/1min 9908.68
$SYS/broker/load/messages/sent/1min 14921.42
$SYS/broker/messages/stored 575
$SYS/broker/publish/messages/dropped 0
$SYS/broker/heap/current size 8945992
$SYS/broker/load/bytes/received/1min 2840549.23
$SYS/broker/load/bytes/sent/1min 5655039.42
```

Graphically



500 bytes / 1000 clients at regular 1/6 Hz rate

Command:

```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:zzz.xxxxxx.com,warhead=warheads/50x0.166mps--1000x500bytes.wh
```

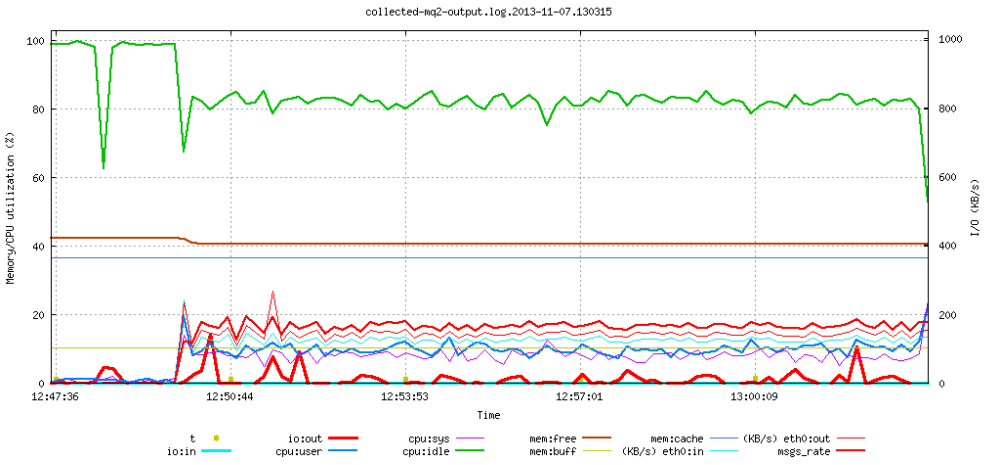
Anecdotally

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
27%	32%	280/2000	~165 (on target)

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'

$SYS/broker/clients/active 1002
$SYS/broker/load/messages/received/1min 19879.14
$SYS/broker/load/messages/sent/1min 29846.02
$SYS/broker/messages/stored 1572
$SYS/broker/publish/messages/dropped 0
$SYS/broker/heap/current size 11169416
$SYS/broker/load/bytes/received/1min 5709338.59
$SYS/broker/load/bytes/sent/1min 11349351.05
```

Graphically



500 bytes / 2000 clients at regular 1/6 Hz rate

Command:

```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:zzz.xxxxxxx.com,warhead=warheads/100x0.166mps--1000x500bytes.wh
```

Anecdotally

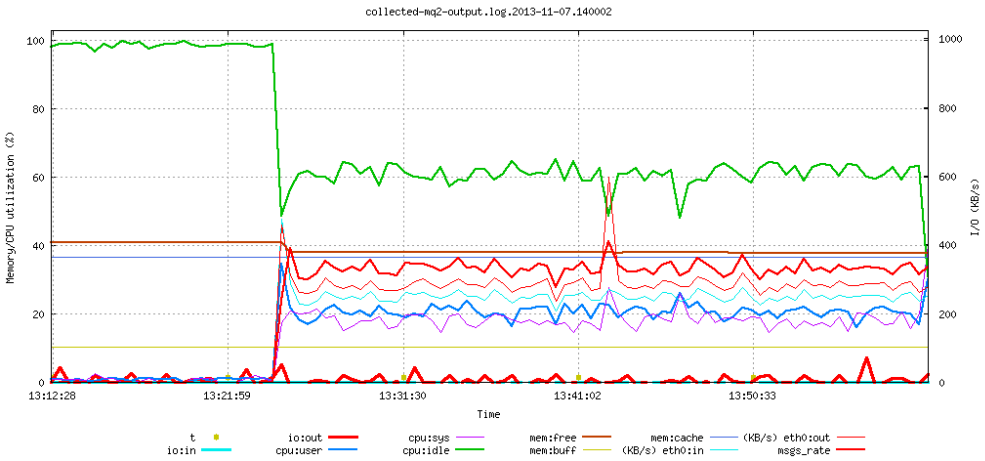
Probably unfair here, as it wasn't a clean start?

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
67-68%	69%	340/2000	~330 (on target)

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'

$SYS/broker/clients/active 2002
$SYS/broker/load/messages/received/1min 39730.73
$SYS/broker/load/messages/sent/1min 59622.31
$SYS/broker/messages/stored 3570
$SYS/broker/publish/messages/dropped 0
$SYS/broker/heap/current size 15612056
$SYS/broker/load/bytes/received/1min 11421097.80
$SYS/broker/load/bytes/sent/1min 22713399.66
```

Graphically



500 bytes / 2500 clients at regular 1/6 Hz rate

mosquitto had been restarted here and the /var/lib/mosquitto/mosquitto.db file removed, due to some heavy load testing earlier. This probably explains why it behaves better than 2000 clients.

Command:

```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:zzz.xxxxxxx.com,warhead=warheads/125x0.166mps--1000x500bytes.wh
```

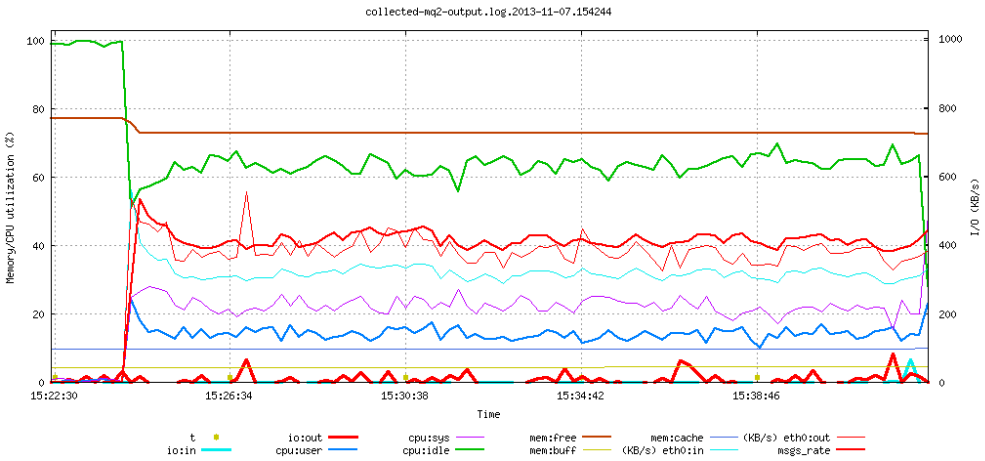
Anecdotally

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
56%	61%	295/2000	~410 (on target)

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'

$SYS/broker/clients/active 2501
$SYS/broker/load/messages/received/1min 49701.50
$SYS/broker/load/messages/sent/1min 74583.20
$SYS/broker/messages/stored 2572
$SYS/broker/publish/messages/dropped 0
$SYS/broker/heap/current size 9252792
$SYS/broker/load/bytes/received/1min 14286319.94
$SYS/broker/load/bytes/sent/1min 28424529.14
```

Graphically



500 bytes / 3000 clients at regular 1/6 Hz rate

Command:

```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:zzz.xxxxxx.com,warhead=warheads/150x0.166mps--1000x500bytes.wh
```

Anecdotally

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
71%	78%	310/2000	~500 (on target)

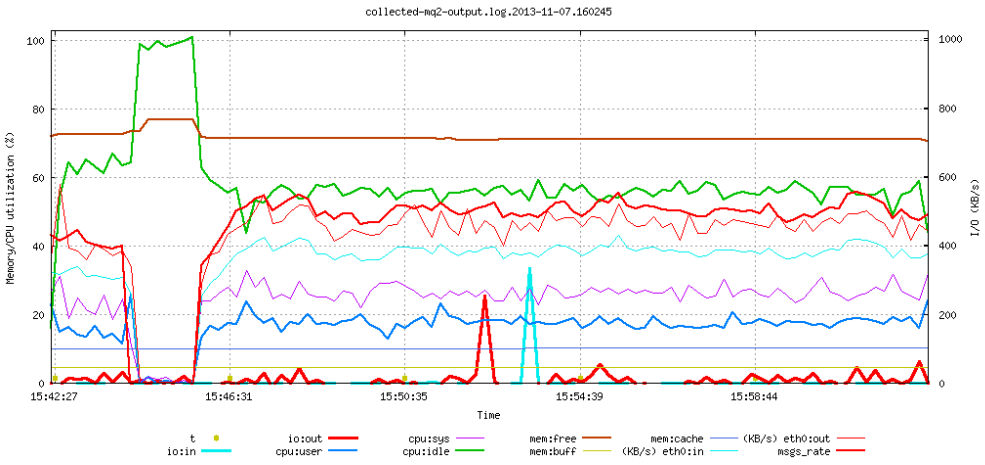
note: fwiw, running qos 0 on the *publishers* brings the cpu usage down to ~55%.

Note: turning off SSL has almost no effect, as it's poll bound in the kernel

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'

$SYS/broker/clients/active 3000
$SYS/broker/load/messages/received/1min 59654.87
$SYS/broker/load/messages/sent/1min 89508.12
$SYS/broker/messages/stored 3068
$SYS/broker/publish/messages/dropped 0
$SYS/broker/heap/current size 10352016
$SYS/broker/load/bytes/received/1min 17152080.11
$SYS/broker/load/bytes/sent/1min 34116110.04
```

Graphically



Perf report

perf report under load

```
# ========
# captured on: Thu Nov  7 15:51:35 2013
# hostname : mq2
# os release : 3.2.0-55-generic
# perf version : 3.2.51
# arch : x86_64
# nrcpus online : 2
# nrcpus avail : 2
# cpudesc : Intel(R) Xeon(R) CPU E5645 @ 2.40GHz
# cpuid : GenuineIntel,6,44,2
# total memory : 2043236 kB
# cmdline : /usr/bin/perf_3.2.0-55 record -p 19164 sleep 20
# event : name = cycles, type = 1, config = 0x0, config1 = 0x0, config2 = 0x0,
excl_usr = 0, excl_kern = 0, id = { 66 }
# HEADER_CPU_TOPOLOGY info available, use -I to display
# HEADER_NUMA_TOPOLOGY info available, use -I to display
# ========
```

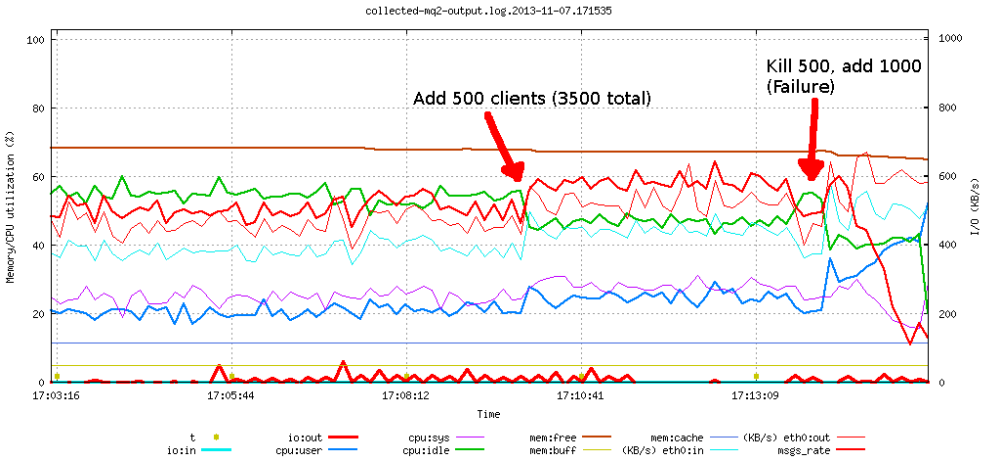
```
#
# Events: 13K cpu-clock
#
# Overhead    Command       Shared Object                                Symbol
# ........  .........  ...................  ...................................
#
    20.89%  mosquitto  mosquitto           [.] mosquitto_main_loop
    20.78%  mosquitto  [kernel.kallsyms]   [k] tcp_poll
    11.06%  mosquitto  [kernel.kallsyms]   [k] sock_poll
     8.36%  mosquitto  mosquitto           [.] mqtt3_db_message_timeout_check
     8.23%  mosquitto  [kernel.kallsyms]   [k] hypercall_page
     8.13%  mosquitto  [kernel.kallsyms]   [k] fget_light
     2.11%  mosquitto  [kernel.kallsyms]   [k] do_poll.isra.4
     1.43%  mosquitto  mosquitto           [.] _mosquitto_packet_write
     1.40%  mosquitto  [kernel.kallsyms]   [k] fput
     1.26%  mosquitto  libc-2.15.so        [.] 0x80353
     1.00%  mosquitto  mosquitto           [.] mqtt3_db_message_write
     0.93%  mosquitto  [kernel.kallsyms]   [k] xen_restore_fl_direct
     0.88%  mosquitto  [kernel.kallsyms]   [k] __pollwait
     0.55%  mosquitto  [kernel.kallsyms]   [k] do_sys_poll
     0.47%  mosquitto  libcrypto.so.1.0.0  [.] sha1_block_data_order_ssse3
     0.44%  mosquitto  libssl.so.1.0.0     [.] ssl3_cbc_digest_record
     0.44%  mosquitto  [kernel.kallsyms]   [k] copy_user_generic_string
     0.36%  mosquitto  libcrypto.so.1.0.0  [.] _bsaes_decrypt8
     0.36%  mosquitto  [vdso]              [.] 0x7fff1ff7098f
     0.25%  mosquitto  [kernel.kallsyms]   [k] xen_save_fl_direct
     0.25%  mosquitto  [kernel.kallsyms]   [k] pvclock_clocksource_read
     0.23%  mosquitto  [kernel.kallsyms]   [k] poll_freewait
     0.17%  mosquitto  [kernel.kallsyms]   [k] _raw_spin_lock_irqsave
     0.15%  mosquitto  libc-2.15.so        [.] read
     0.15%  mosquitto  [kernel.kallsyms]   [k] remove_wait_queue
     0.15%  mosquitto  [kernel.kallsyms]   [k] ktime_get_ts
     0.15%  mosquitto  [kernel.kallsyms]   [k] xennet_poll
     0.15%  mosquitto  [kernel.kallsyms]   [k] __inet_lookup_established
     0.15%  mosquitto  [kernel.kallsyms]   [k] tcp_recvmsg
     0.13%  mosquitto  librt-2.15.so       [.] clock_gettime
     0.13%  mosquitto  [kernel.kallsyms]   [k] _cond_resched
     0.12%  mosquitto  libc-2.15.so        [.] malloc
     0.12%  mosquitto  [kernel.kallsyms]   [k] xen_irq_disable_direct
     0.12%  mosquitto  [kernel.kallsyms]   [k] tcp_sendmsg
     0.12%  mosquitto  libssl.so.1.0.0     [.] ssl3_cbc_copy_mac
     0.12%  mosquitto  [kernel.kallsyms]   [k] __slab_free
     0.11%  mosquitto  libcrypto.so.1.0.0  [.] EVP_MD_CTX_cleanup
```

```
0.11%  mosquitto  [kernel.kallsyms]   [k] add_wait_queue
0.10%  mosquitto  libssl.so.1.0.0     [.] ssl3_read_bytes
more snipped as irrelevant
```

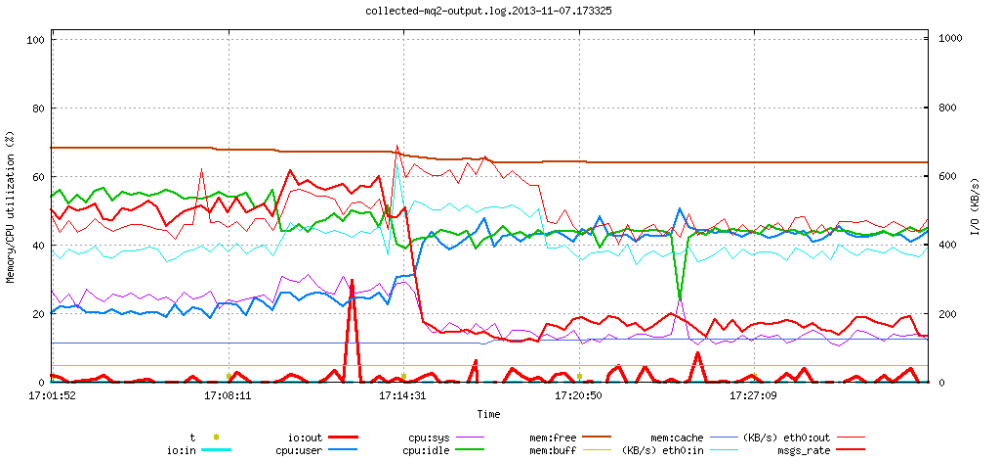
500 bytes / 3000 clients, with X clients added later...

Experimenting with having a quite loaded server, and seeing what happens if we add a bunch of clients, and remove them. We use the basic 20 bees, 3000 clients mode from above, and then add in varying numbers of clients from our local laptop

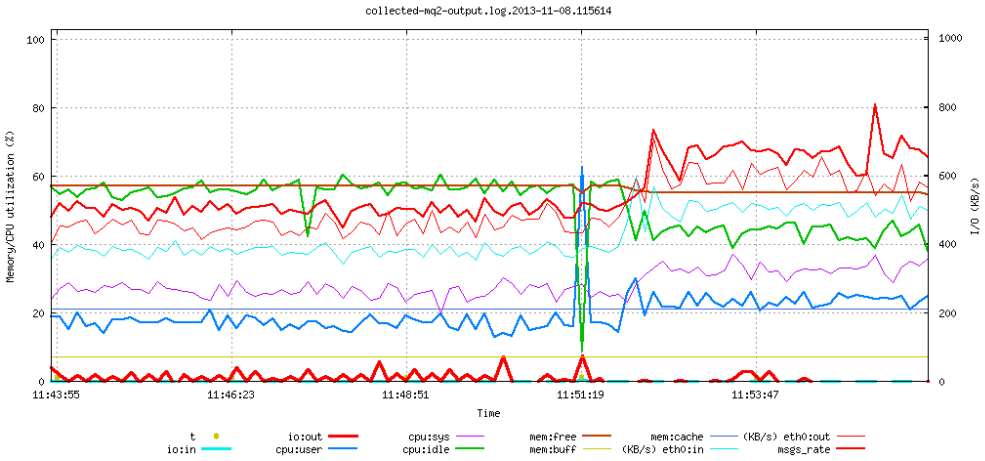
Adding 500 from locally worked just fine, we got a cpu bump up to ~88% cpu, at around 580 MPS. This is more like the real world would behave, as clients would be coming online in dribs and drabs. However, turning those 500 off and adding 1000 killed it.



Even after removing those 1000 clients, the system never recovered, note the drop in thin red, thin blue lines when the 1000 clients were dropped off, but that the thick red line (MPS) doesn't ever come back up. (We were also dropping here)



With a private patched build of mosquitto from roger, running 3000 clients and then adding 1000 actually worked, topping out at ~90% cpu overall



(This lasted for about 30minutes over lunch, but eventually failed as well)

500 bytes / 4000 clients at regular 1/6 Hz rate FAILURE

This doesn't seem to work. It might work if you very slowly ramp up to it, but it sure won't be stable, and any undue load will tip it over the edge from which it will be very unlikely to recover

With a clean clean mosquitto start, this *almost makes it*, running abotu 98% cpu, and managing 650 mps or so, but it slowly starts to fill queues, and it's all over when that happens in steady state. It starts to drop in a few minutes

Command:

```
fab -i ~/.ssh/karl-malaria-bees-2013-oct-15.pem mstate
attack:mq.dcc02.etactica.com,warhead=warheads/200x0.166mps--1000x500bytes.wh
```

Anecdotally

Mosquitto CPU	Overall CPU	MemoryUsage	Messages per Second
x%	xx%	x/2000	~x (on target)

```
karlp@mq2:~$ mosquitto_sub -t '$SYS/broker/messages/dropped' -v -t
'$SYS/broker/clients/active' -t '$SYS/broker/load/messages/received/1min' -t
'$SYS/broker/load/messages/sent/1min' -t '$SYS/broker/messages/stored' -t
'$SYS/broker/publish/messages/dropped' -t '$SYS/broker/heap/current size' -t
'$SYS/broker/load/bytes/received/1min' -t '$SYS/broker/load/bytes/sent/1min'

x
```