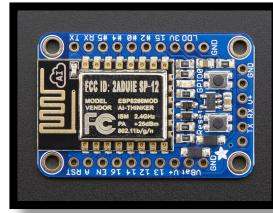


[BSS WiFi Multi-device testing]

5-Mar, 2017

Concordia University



Experiment designed and conducted by:
John Sullivan and Travis West

with the participation of the Body:Suit:Score research-creation team and volunteers



[Overview]

- We organized an experiment to test the capabilities of sending messages to multiple wireless devices on the same WiFi network.
- A series of 37 trials was staged to evaluate the robustness of sending and receiving messages between the devices and a host computer, testing different network setups and a number of simulated performance conditions.
- Message reliability was evaluated by 3 metrics:
 - Dependability (messages received/messages sent)
 - Latency (round trip time for messages sent from host to devices and devices echo back to host)
 - Jitter (timing variation during series of timed messages)

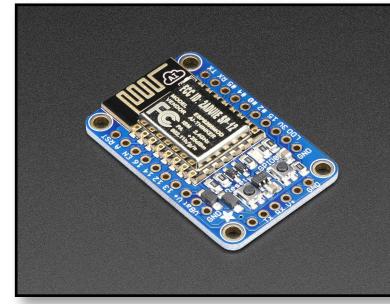
[Experiment]

- For the experiment, 8 users were each given a device comprised of a wireless microcontroller, battery, and onboard LED light.
- For each trial, a host computer sent 16 messages one at a time to all devices which blinked the LEDs in one of three different patterns: **fully on**, **steady blinking**, and **random blinking**.
- Using a web application running on their smartphones, users recorded the patterns that their devices received.
- The devices also echoed back the message to the host computer.
- Data files stored the host message sent, the recorded messages that the users entered, and the messages that the devices echoed back to the host. Sent and echoed messages were timetagged to check roundtrip latency.
- Additionally, the random blinking message was a fast series of on/off messages with random timing that allowed us to test the timing accuracy of the network.

[Devices and protocols]

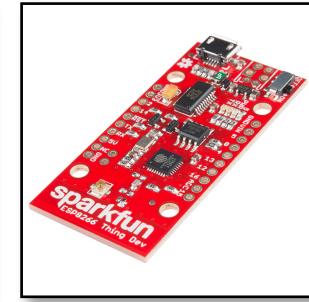
- 6 x Adafruit Huzzah

- <https://www.adafruit.com/product/2471>



- 2 x Sparkfun ESP8266 Thing Dev

- <https://www.sparkfun.com/products/13711>



- Both devices utilize the Espressivo ESP8266 WiFi-enabled microcontroller, laid out on a development board
- Network connection was made over a regular WiFi network, using a Linksys WRT54G router running Tomato firmware.

[Trials]

- The trials were grouped into 4 sets:

1 – 15: Network configurations

- 1 – 5: WPA2 protected network
- 6 – 10: open, unprotected network
- 11 – 15: open network restricted to authorized MAC addresses (only the host computer, devices, and users' smartphones were authorized)
- For each configuration, trials compared simultaneous vs staggered message sending, same vs different messages to each device, and broadcasting 1 message to all devices.

16 - 21: Sensor data streaming

- Devices sent continuous data at regular intervals to simulate the network traffic of devices streaming sensor data.
- Trials increased the rate of streaming: 250ms, 125ms, 64ms, 32ms, 16ms, 8ms.

22-29: Cross-channel interference

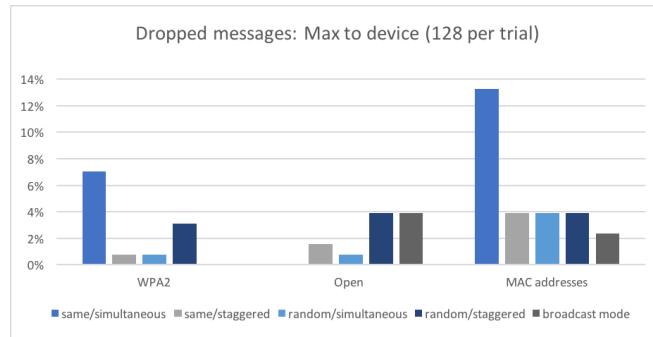
- Trials were conducted while a separate network was utilized for live wireless performance of an audio piece.
- While the device network utilized WiFi channel 1, the other network operated on ch's 11, 8, 4, and 1.
- Trials were conducted with and without devices streaming sensor data as well.

30 - 37: Other performance conditions

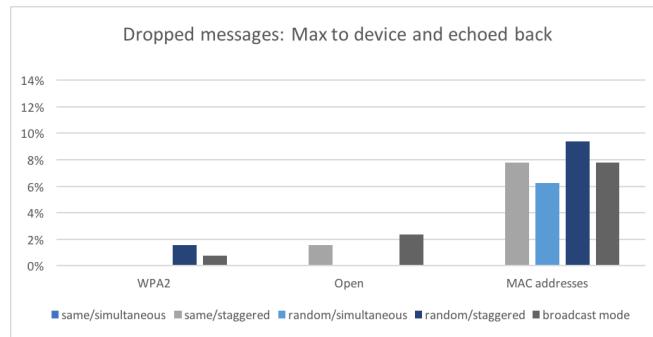
- 30-31: Body occlusion (held tight/not held)
- 32-33: Long distance (with/without sensor data)
- 34-35: Walking (with/without sensor data)
- 36/37: Huddled together (with/without sensor data)

[Results: Message delivery]

a.



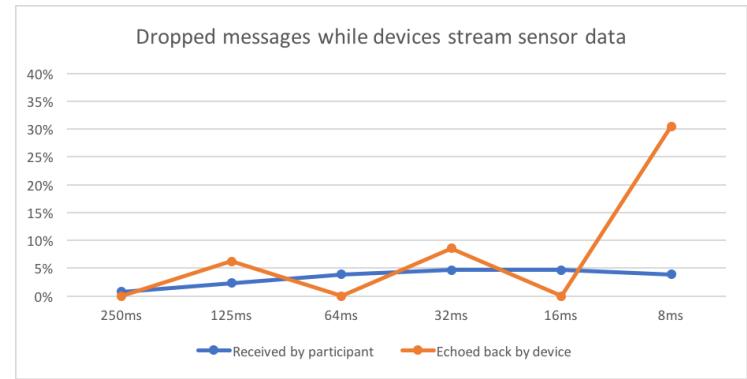
b.



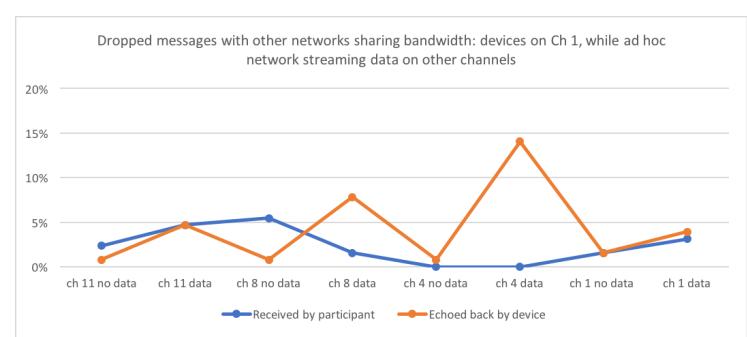
Percent of dropped messages per trial:

- 1-15 Network configurations across conditions (host to device – user recorded)
- Same trials as above (host to device, device echoed back to host)
- 16-22 With sensor data streaming at increasing frequency
- 23-29 With cross-channel traffic on another network (Joe test)
- 30-17 Other conditions, with and without streaming data: body occlusion, distance, walking, huddled together.

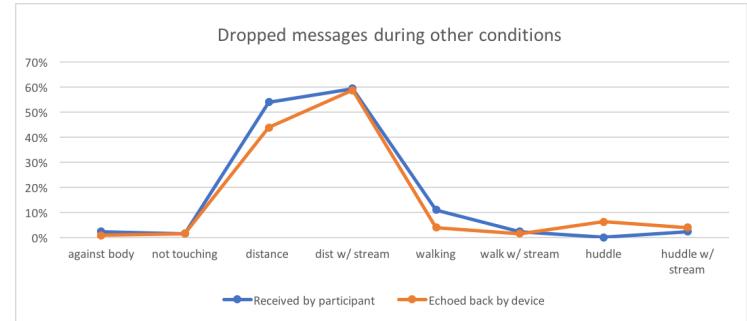
c.



d.

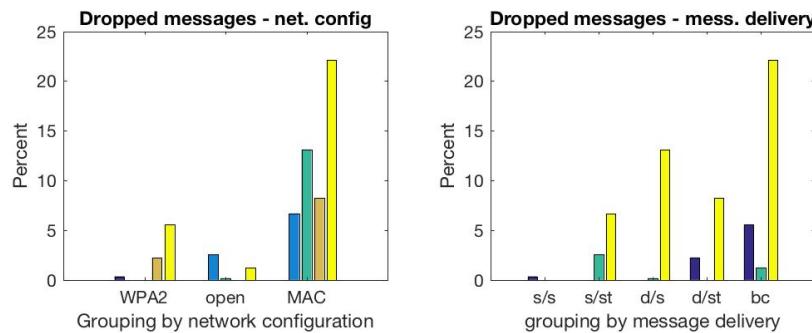
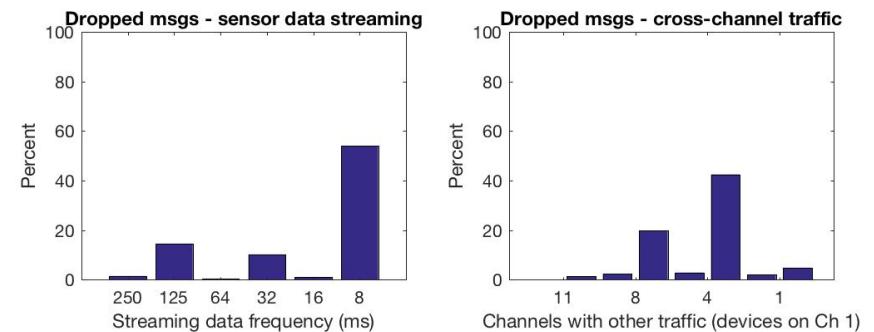
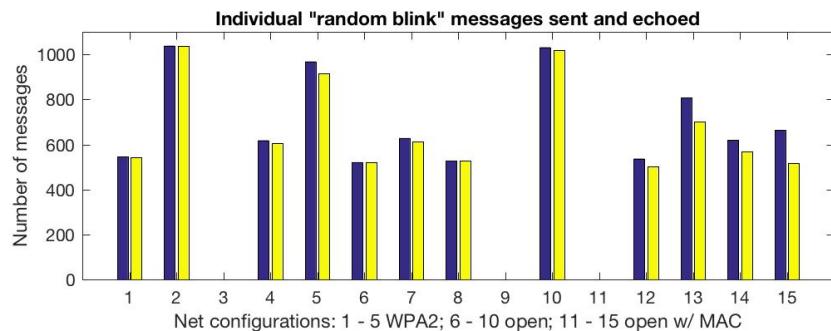


e.



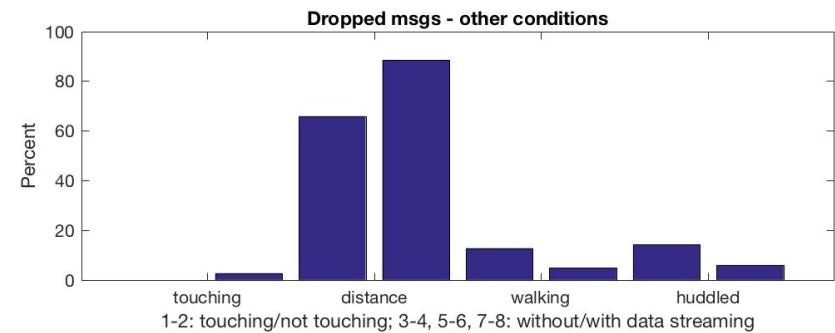
[Message delivery (continuous data)]

Delivery of timing-sensitive continuous control-style messages was tested as well. When one of the “random blinking” messages was sent, instead of a single message, a random sequence of timed on and off messages were sent over a period of 2 seconds. Thus each random message actually contained between 10 and 20 individual time-tagged message. The number sent vs received give a measure of delivery success and percentage of dropped packets (messages), as shown here. See also [Jitter] section.



Top: Total individual messages sent/echoed in the random blinks messages for trials 1 – 15 (network configuration tests)

Bottom: % dropped messages by network type (left) and message delivery condition (right)

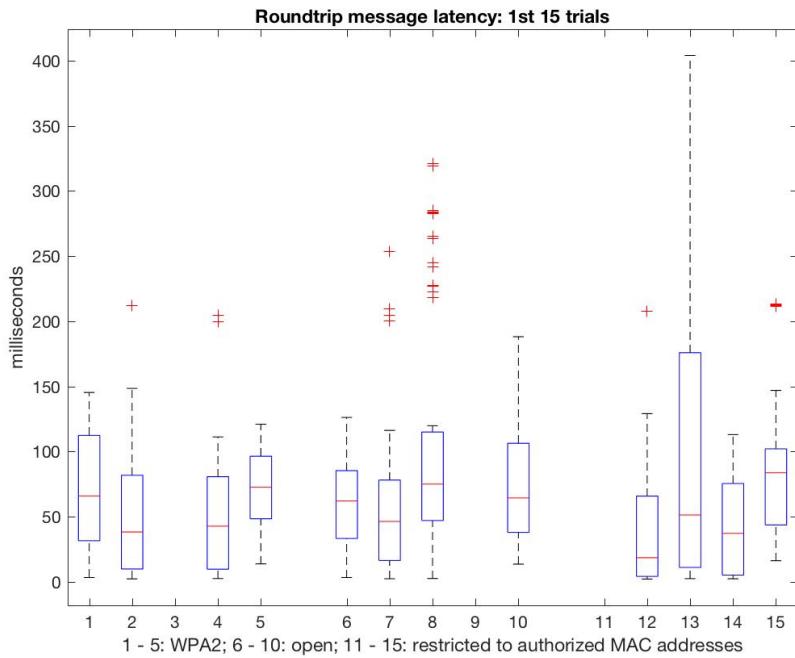


Top left: Trials 16-21, with sensor data streaming

Top right: Trials 22 – 29, with cross channel traffic

Bottom: Trials 30 – 37: Other conditions

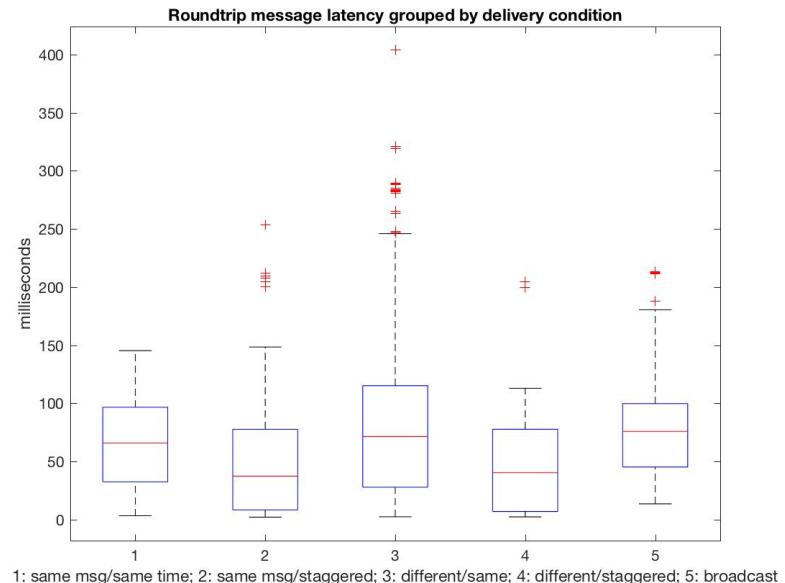
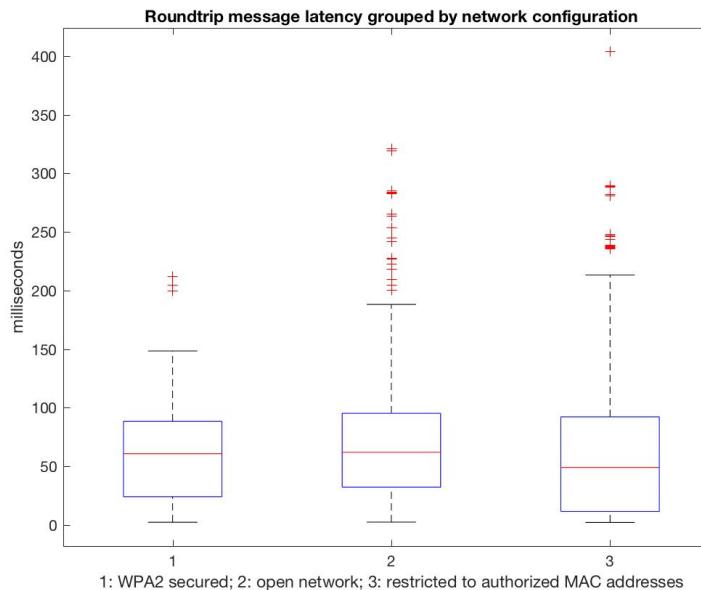
[Results: Latency]



Latency was measured by comparing the timetags of messages sent from the host computer with the timetags of the same messages echoed back from the devices.

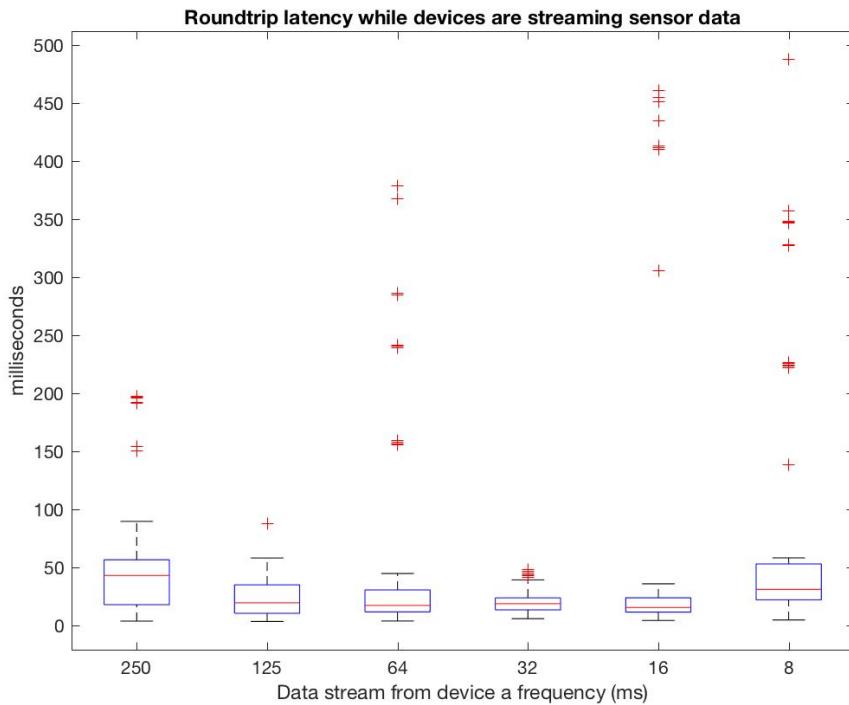
Trials 1 – 15 (testing different network configurations and message-sending strategies):

- all
- by network configuration
- by delivery condition



Boxes = interquartile range; **red lines** = medians; **dotted lines** = upper/lower extremes; **crosses** = outliers

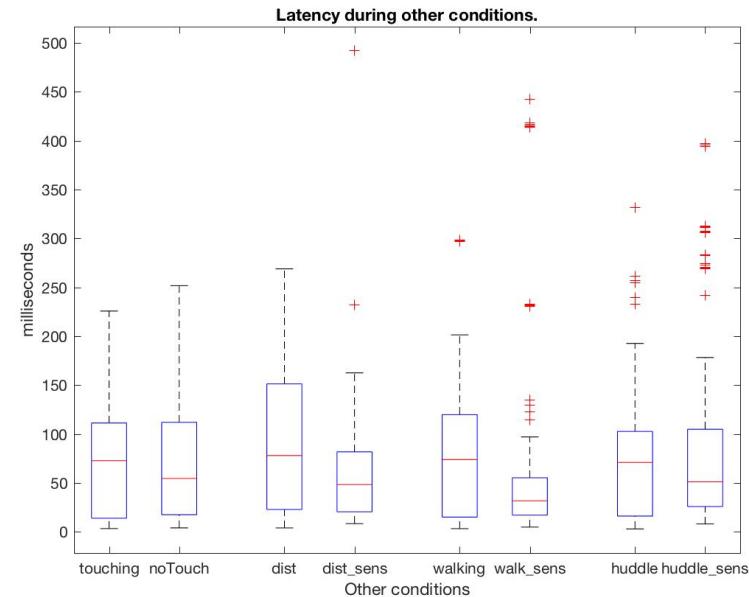
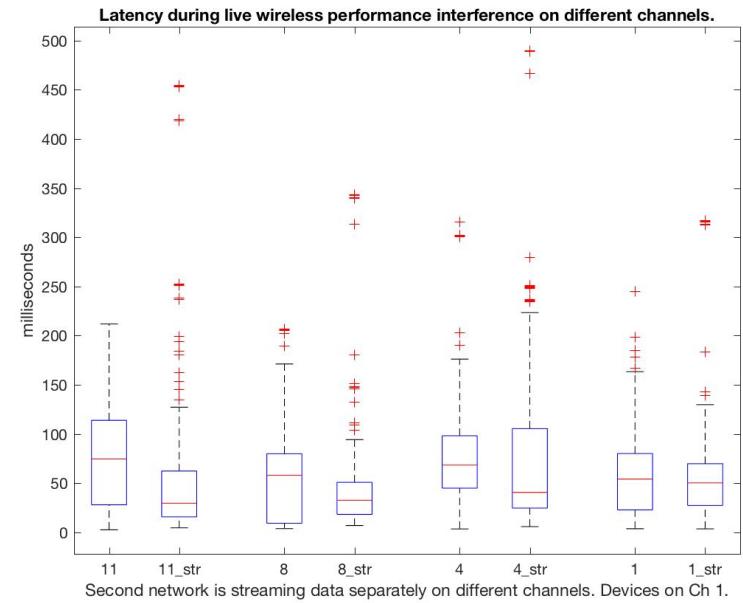
[Latency (cont.)]



Above: Trials 16 – 21 with sensor data streaming at different rates

Right top: Trials 22-29 with cross-channel traffic (with and without devices sensor data)

Right bottom: Trials 30 - 37 testing other conditions: body occlusion (touching/not touching); distance (without/with sensor data), walking (without/with sensor data), huddled (without/with data)

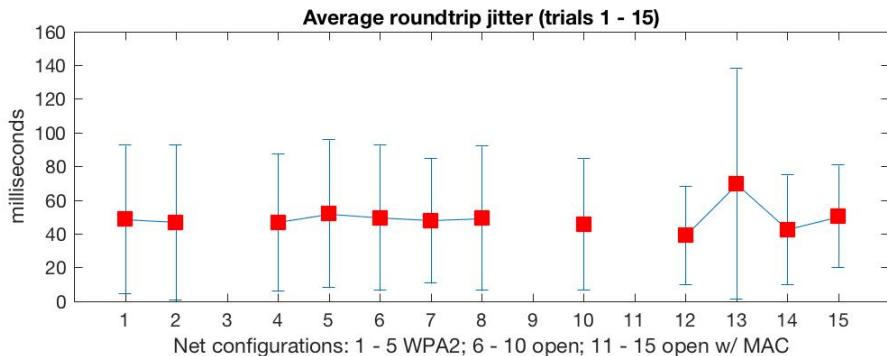
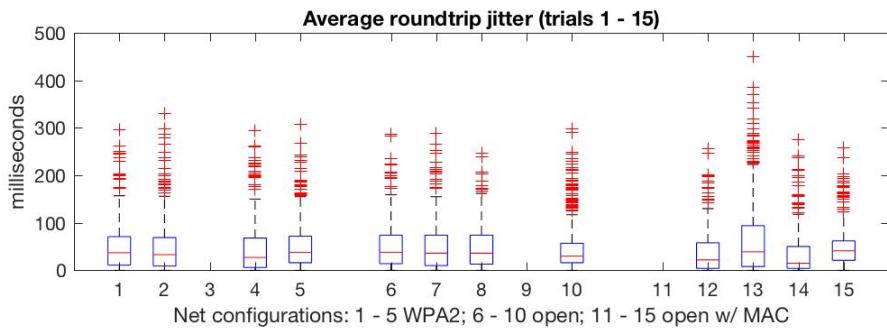


Boxes = interquartile range; **red lines** = medians; **dotted lines** = upper/lower extremes; **crosses** = outliers

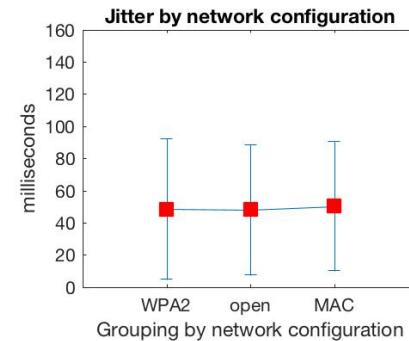
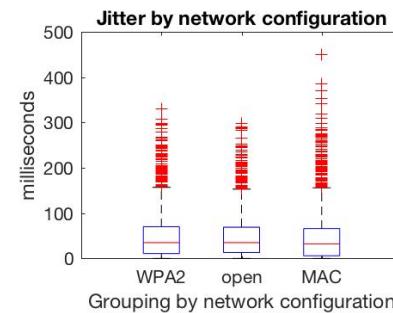
Jitter

Jitter was measured when one of the “random” blinking messages was sent. A random message consisted of a series of randomly timed on and off messages. Jitter was calculated by comparing the specific timing sequence of the string of sent messages with the timing sequence of the messages echoed back.

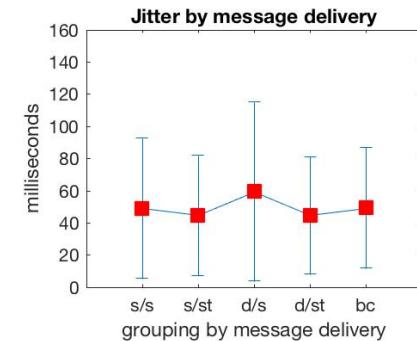
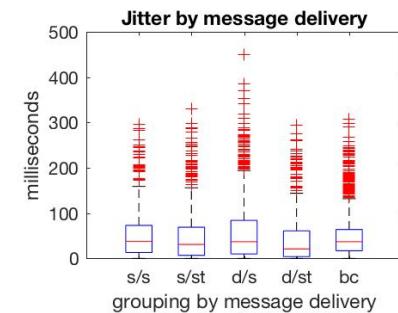
Trials 1 – 15: Network configuration and message delivery condition



Jitter (ms) across all trials.



Jitter by network configuration



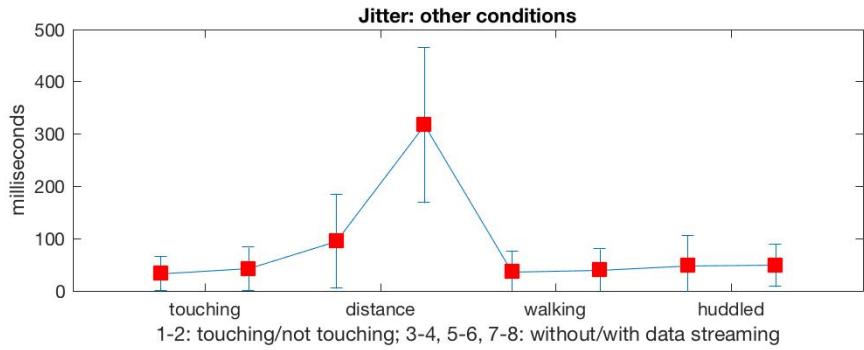
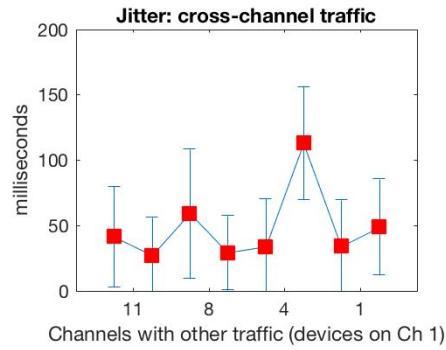
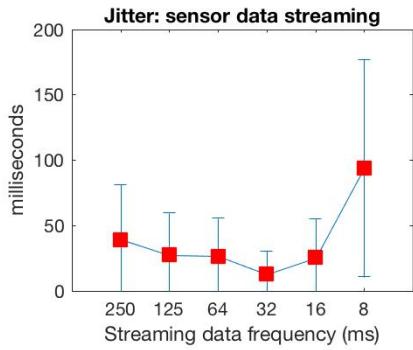
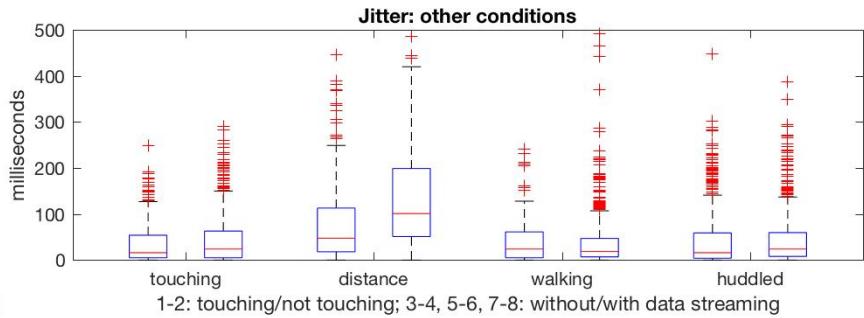
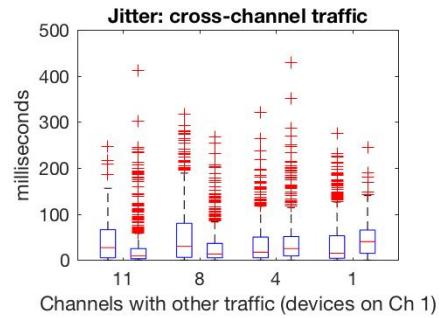
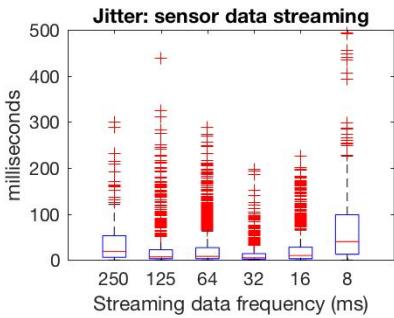
Jitter by message delivery condition

Top: **Boxes** = interquartile range; **red lines** = medians; **dotted lines** = upper/lower extremes; **crosses** = outliers

Bottom: **Red squares** = mean; **vertical bars** = standard deviation

[Jitter (cont.)]

16-21: Data Streaming; 22-29: Cross-channel traffic; 30 – 37: Other conditions



Jitter during data streaming

Jitter with cross-channel traffic

Jitter across other conditions.

Top: **Boxes** = interquartile range; **red lines** = medians; **dotted lines** = upper/lower extremes; **crosses** = outliers

Bottom: **Red squares** = mean; **vertical bars** = standard deviation