

MQTT-Benchmark

Speicher- und Datennetze im IoT

Von Patrick Olinger und David Tarnow

Agenda

- (1) Zielsetzung
- (2) Implementierung
- (3) Auswirkung der Nachrichtengröße
- (4) Auswirkung des Zeitintervalls
- (5) Langzeitmessungen
- (6) Fazit

Zielsetzung

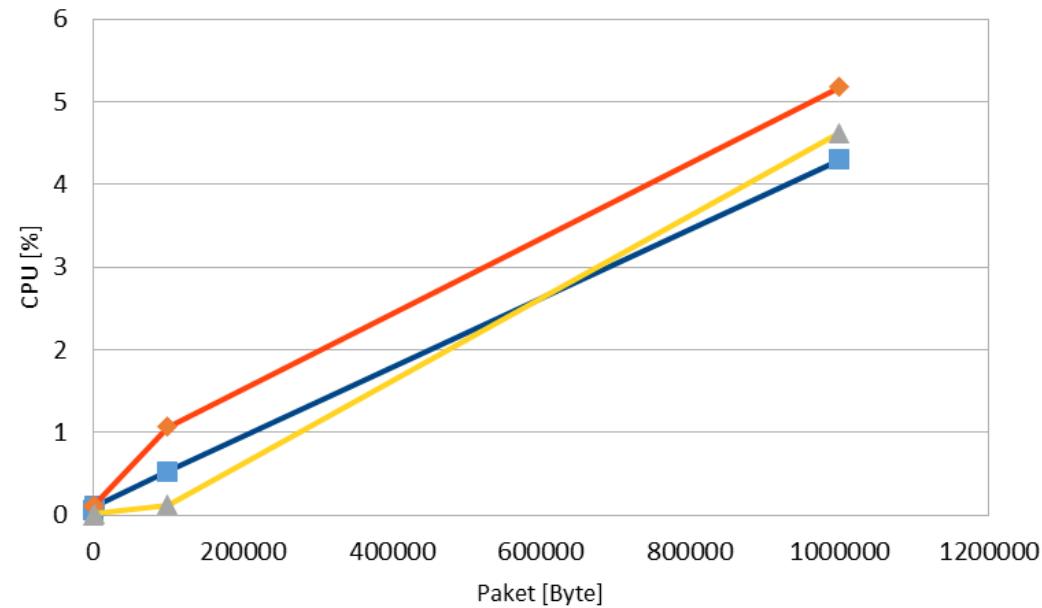
- Implementierung von
 - Paho MQTT-Client in C
 - Paho MQTT-Client in C#
 - Qt MQTT-Client in C++
- Messung
 - RAM-Verbrauch
 - CPU-Nutzung
- Auswirkung der Nachrichtengröße
 - Zeitintervall: 100 ms
 - Nachrichtengröße: 0, 1.000, 100.000, 1.000.000 bytes
- Auswirkung des Zeitintervalls
 - Zeitintervall: 10 ms, 100 ms, 1.000 ms
 - Nachrichtengröße: 100.000 bytes

Implementierung

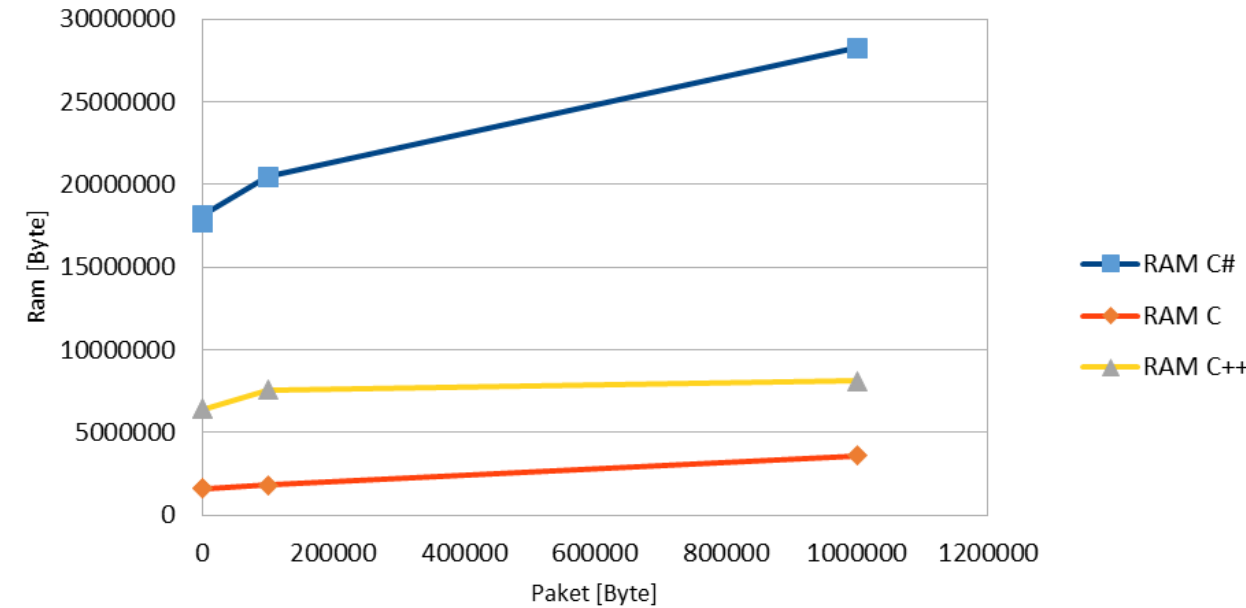
- Jeweils 2 Publisher und Subscriber (Windows)
- MQTT-Broker Mosquitto (Linux)
- Publisher veröffentlichen in vorgegebenem Zeitintervall
- Subscriber beobachten die Topics
- Messung von jeweils einem Subscriber / Publisher

Auswirkung der Nachrichtengröße

Subscriber 100ms



Subscriber 100ms



Auswirkung der Nachrichtengröße

Auswirkung der Nachrichtengröße – C#

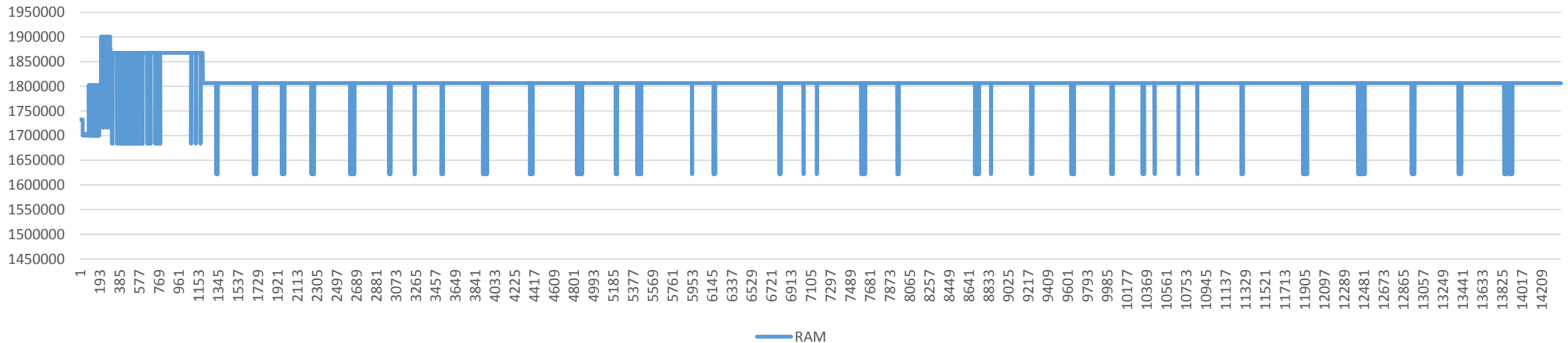
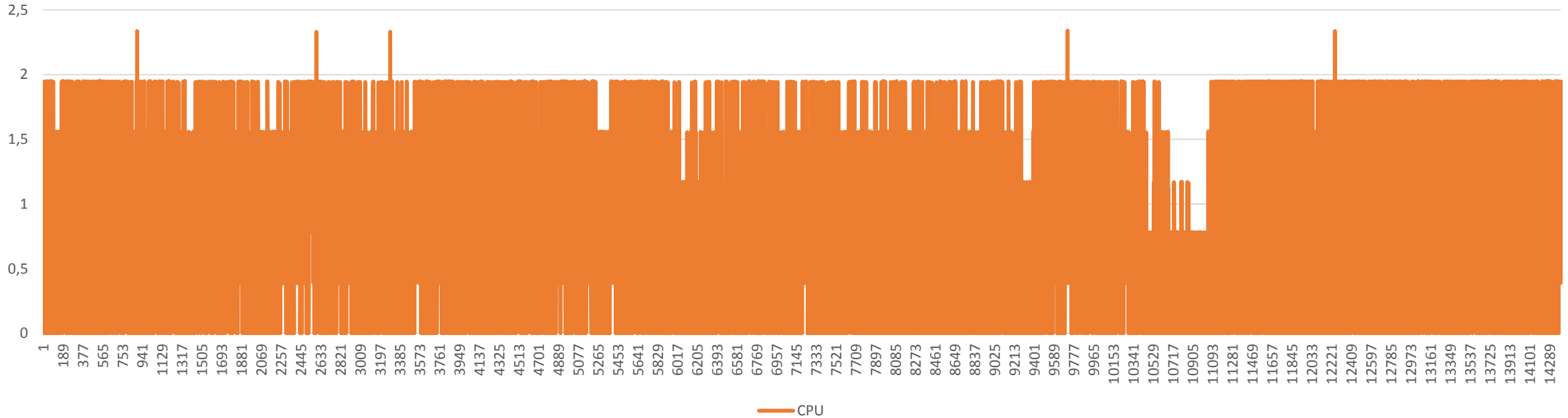
Auswirkung der Nachrichtengröße – C++

Auswirkung des Zeitintervalls – C

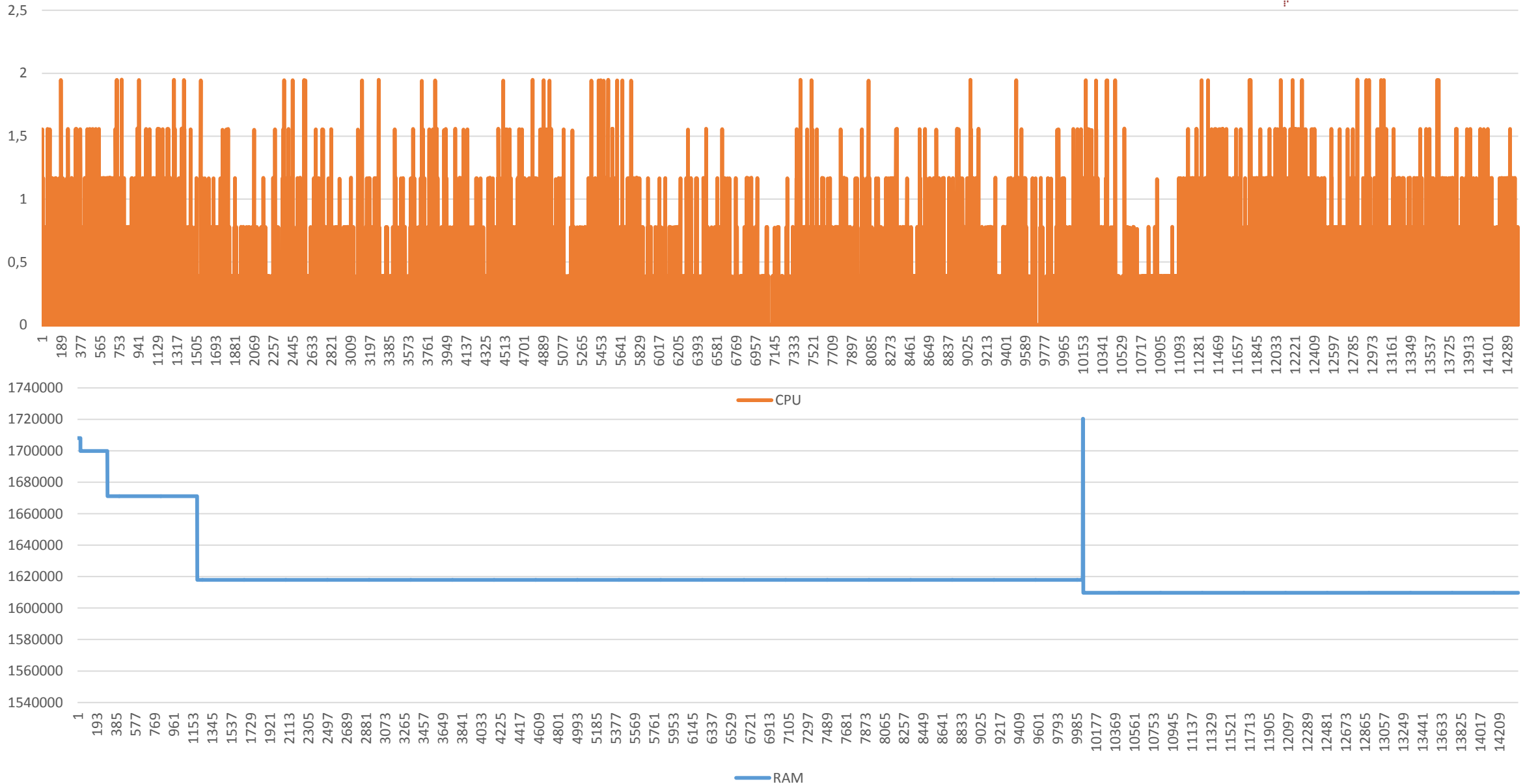
Auswirkung des Zeitintervalls – C#

Auswirkung des Zeitintervalls – C++

Langzeitmessung – C: Subscriber



Langzeitmessung – C: Publisher



Fazit

- RAM-Auslastung
 - Größtenteils Abhängig von Nachrichtengröße
 - C: gering, C++: mittel, C#: hoch
- CPU-Auslastung
 - Größtenteils Abhängig von Zeitintervall