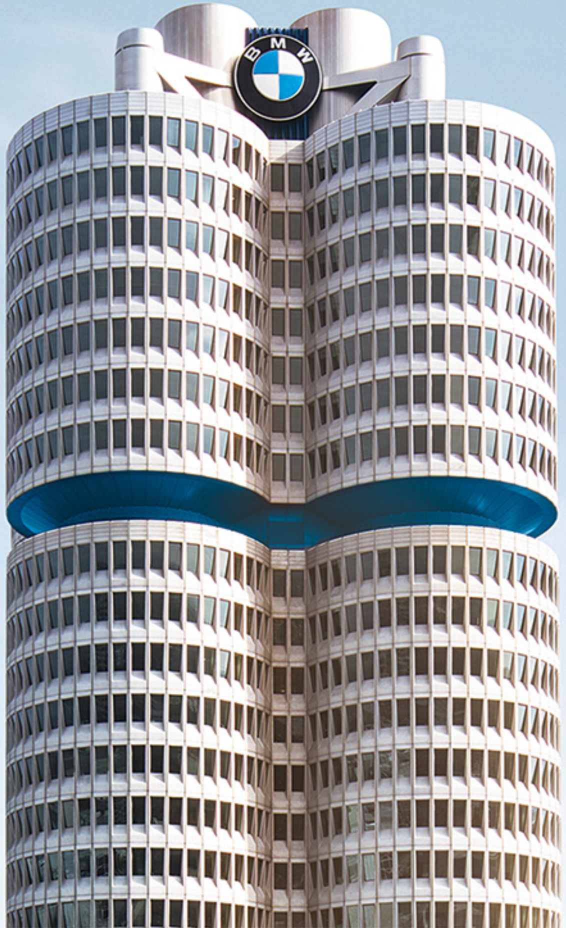
FLATBUFFER PERFORMANCE IN OSI BASED AGENT MODELS



EN-50 | 30.05.22
Fabian Pfeuffer

Experimental setup

- Goal:
  - Comparison between the performance of Protobuf and Flatbuffer as transfer protocol for OSI messages for a complete agent model.
- Agent model:
  - OsiPedestrian Protobuf version
  - OsiPedestrian Flatbuffer version, tables only
  - OsiPedestrian Flatbuffer version, structs for osi_common.fbs elements (Vector3d, Orientation3d, Size3d...)
- Simulator:
  - Minimal custom test simulator that runs the agent model in an open loop simulation.
- Data:
  - Complete simulation step including simulator side serialization, agent side deserialization, agent side simulation step and agent side serialization of traffic update.
  - Increasing message size in steps of 10 dummy moving object with the same position, orientation and size attributes.
  - Dummy object raw data: position(10,10,10), orientation(π,0,0), size(4.5, 1.8, 1.6)
    - 3x24 Byte (position, orientation, size) + 8 Byte (Id) = **80 Byte** (default values included)
    - 2x24 Byte (position, size) + 2x8 Byte (orientation, Id) = **64 Byte** (default values not included)
- Machine: Dell Precision 7520, 64GB RAM, Ubuntu (18.04)

Actions performed for a single simulation time step

1) Simulator side serialization of the osi3::SensorView (SensorView generation is not included)

2) Data is sent to the OsiPedestrian (Data pointer encoded as a 4 Byte integer)

3) Data is received by the OsiPedestrian (4 Byte integer decoded to a data pointer) (start of fmi2DoStep)

4) Data is deserialized to an osi3::SensorView

5) OsiPedestrian performs his update step under consideration of the received osi3::SensorView (Open-Loop)

6) OsiPedestrian generates an osi3::TrafficUpdate

7) Pointer to TrafficUpdate is encoded as a 4 Byte integer (end of fmi2DoStep)

(The simulator will not consider the traffic update, because the benchmark is run in a open loop simulation, and continues with the next simulation step.)

Results

- Each data point averaged over 100 repeated simulation steps
- Regression model:

$$t_{step}(N_{obj}) = t_0 + N_{obj} \cdot t_1$$

- Protobuf:

$$t_{1Proto} = (59.58 \pm 0.08) \frac{\mu s}{10}$$

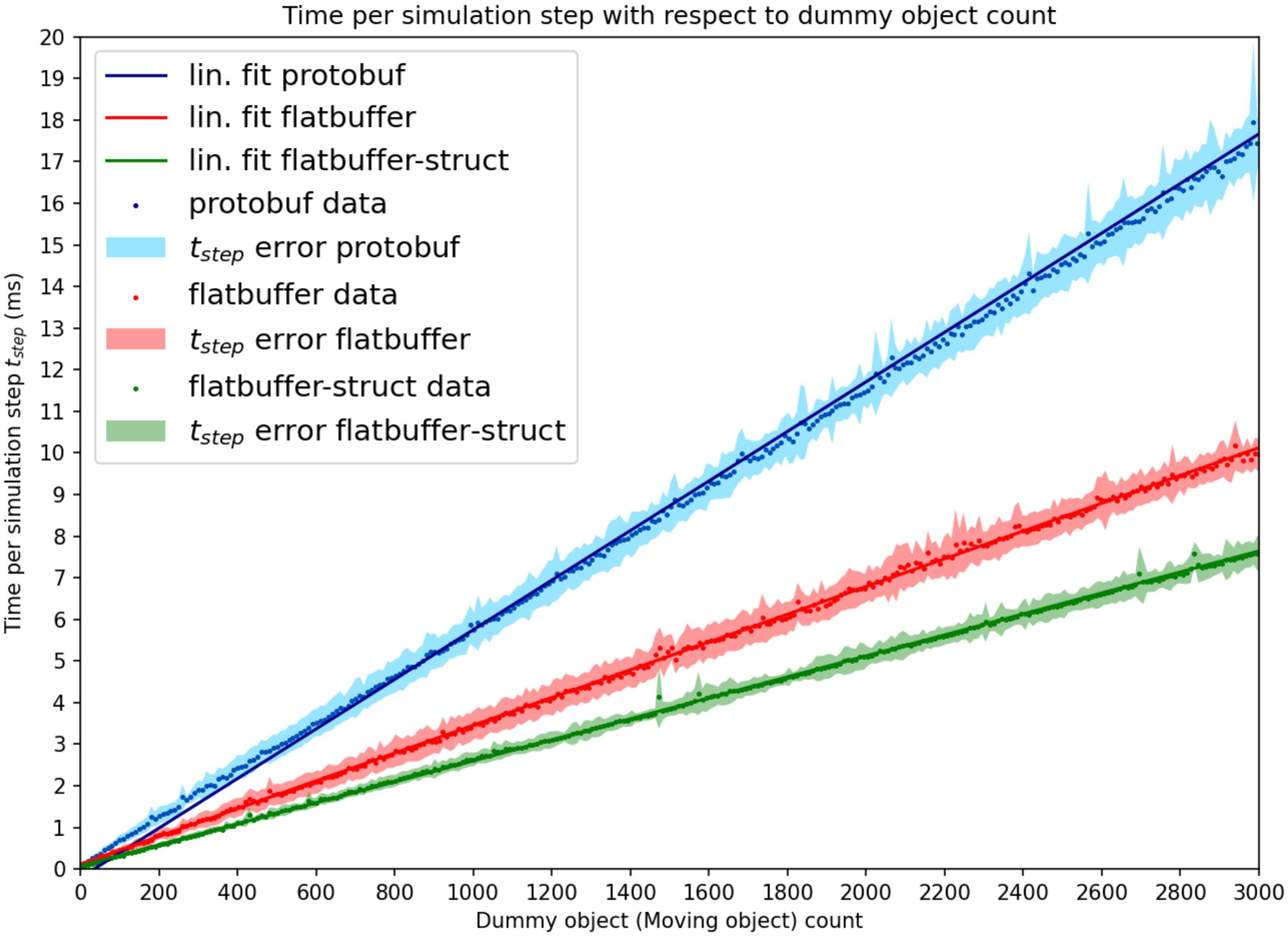
- Flatbuffer (tables only):

$$t_{1FlatT} = (33.36 \pm 0.05) \frac{\mu s}{10}$$

- Flatbuffer (tables + struct):

$$t_{1FlatS} = (25.21 \pm 0.02) \frac{\mu s}{10}$$

- Scaling for Flatbuffer (tables + struct) per dummy object roughly 136% faster than Protobuf



Results

- Simulation results for “low” object counts
- Each data point averaged over 1000 repeated simulation steps
- Regression model:

$$t_{step}(N_{obj}) = t_0 + N_{obj} \cdot t_1$$

- Protobuf:

$$t_{1\,Proto} = (48.57 \pm 0.13)\frac{\mu s}{10}$$

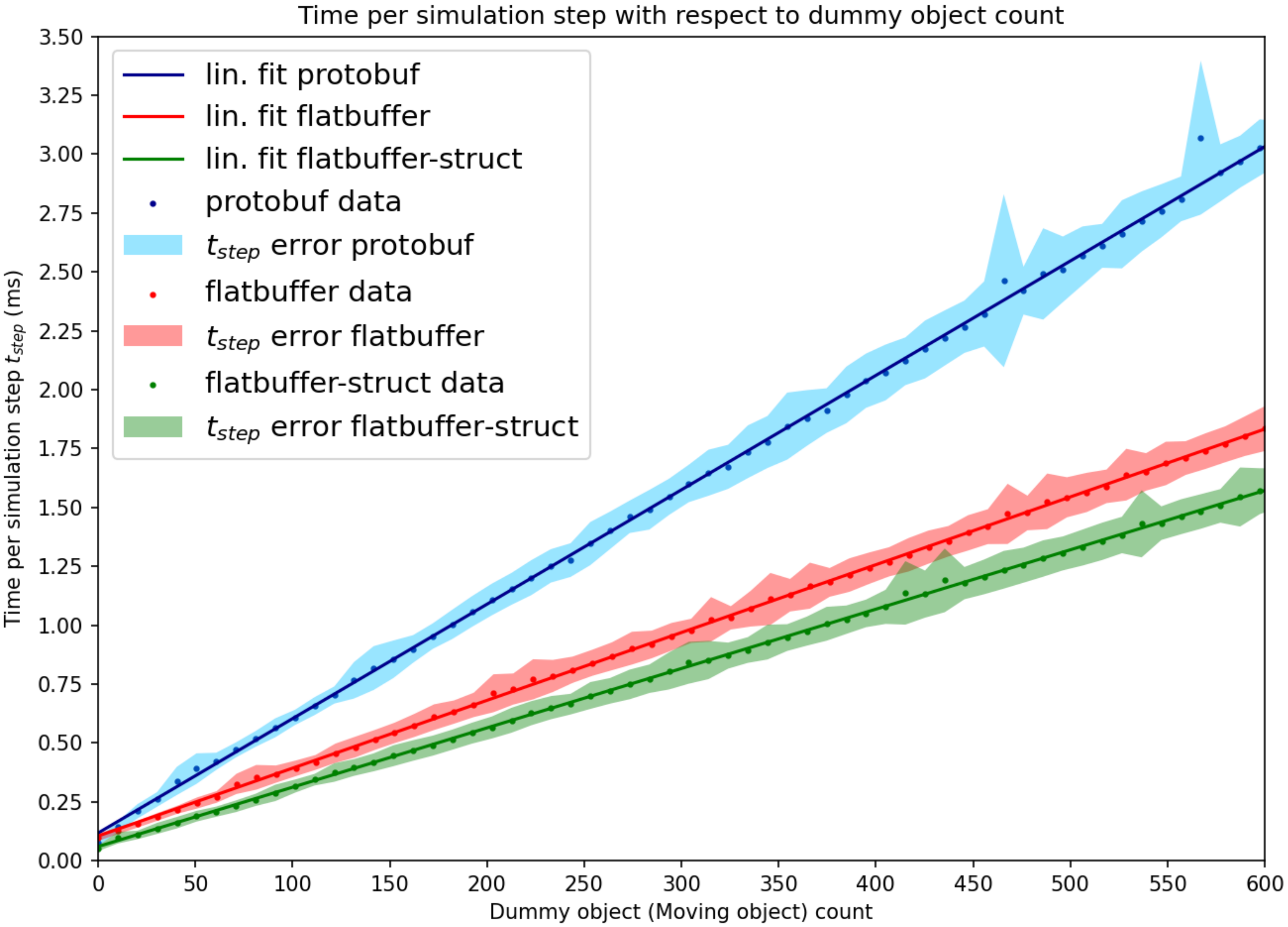
- Flatbuffer (tables only):

$$t_{1\,FlatT} = (28.79 \pm 0.06)\frac{\mu s}{10}$$

- Flatbuffer (tables + struct):

$$t_{1\,FlatS} = (25.20 \pm 0.04)\frac{\mu s}{10}$$

- Scaling for Flatbuffer (tables + struct) per dummy object roughly 93% faster than Protobuf



Results

- Each data point averaged over 1000 repeated simulation steps
- Regression model:

$$t_{step}(S_{msg}) = t_0 + S_{msg} \cdot t_1$$

- Protobuf:

$$t_{1\,Proto} = (76.50 \pm 0.10)\frac{\mu s}{kB}$$

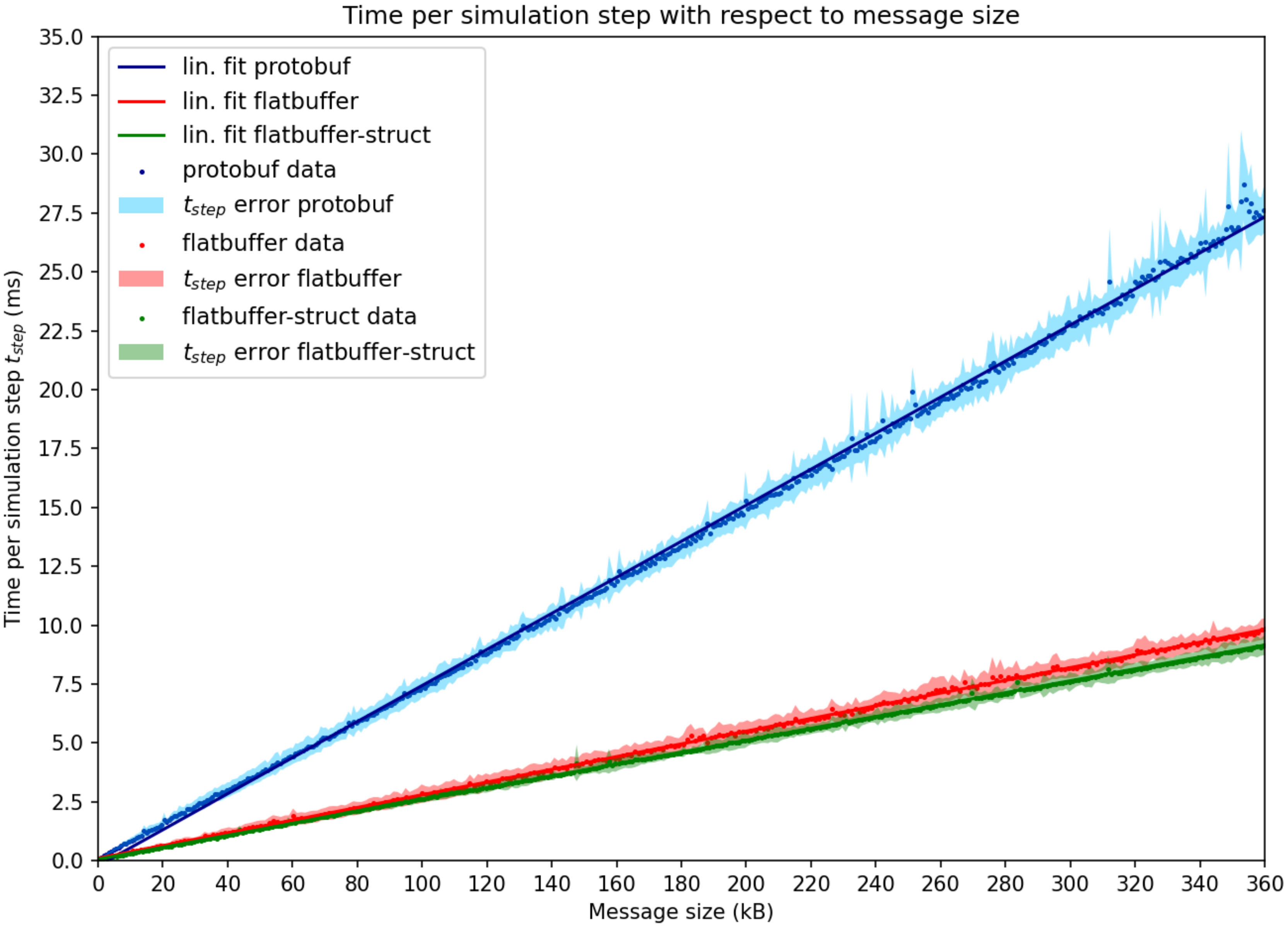
- Flatbuffer (tables only):

$$t_{1\,FlatT} = (26.99 \pm 0.04)\frac{\mu s}{kB}$$

- Flatbuffer (tables + struct):

$$t_{1\,FlatS} = (25.26 \pm 0.02)\frac{\mu s}{kB}$$

- Scaling for Flatbuffer (tables + struct) per kB message size roughly 202% faster than Protobuf



Results

- Careful when comparing performance with respect to message size!
- Protobuf:

$$S_{Proto\,obj} = 77\ Byte$$

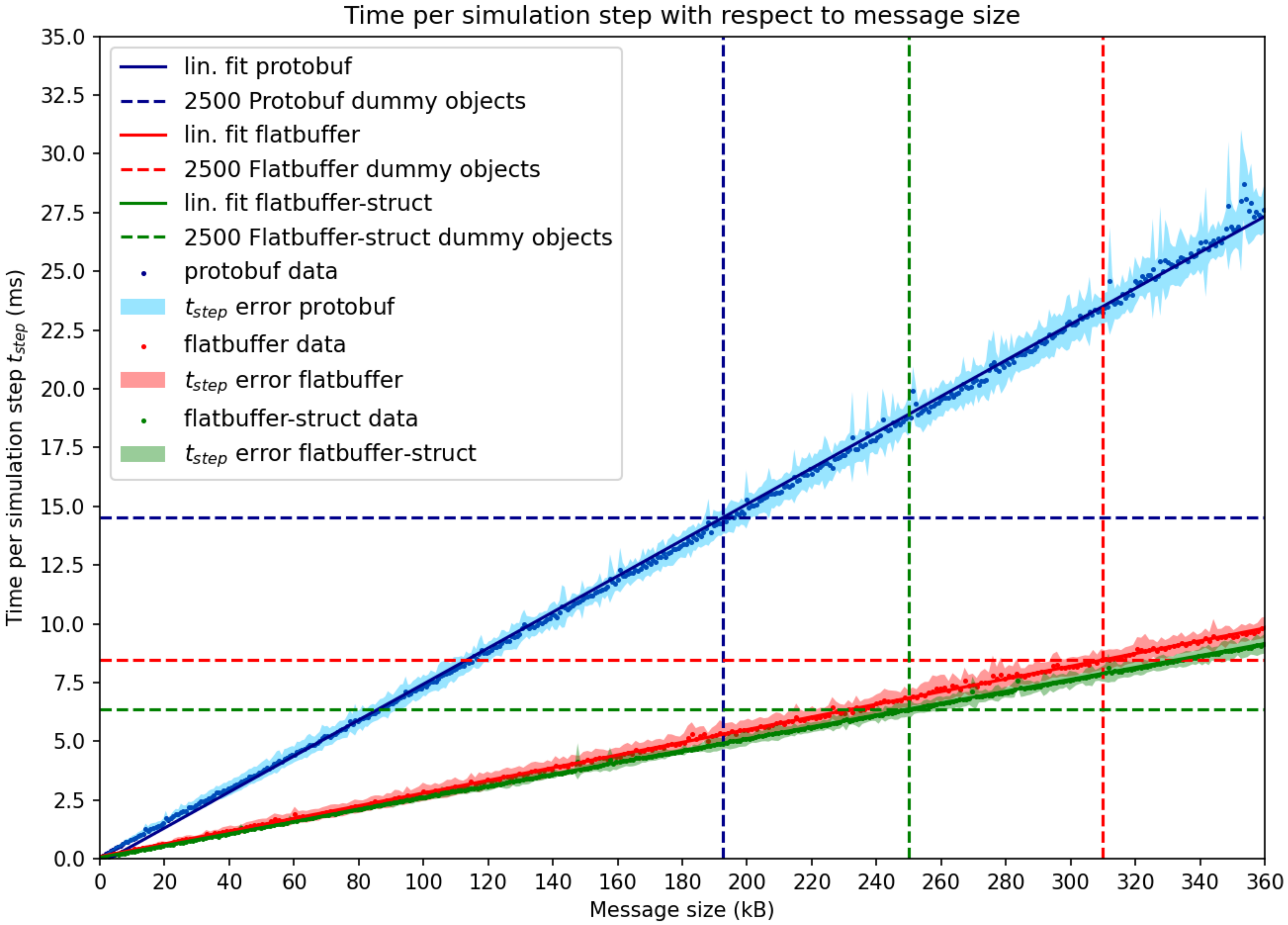
- Flatbuffer (tables only):

$$S_{FlatT\,obj} = 124\ Byte$$

- Flatbuffer (tables + struct):

$$S_{FlatS\,obj} = 100\ Byte$$

- Flatbuffer (tables + struct) message roughly 30% larger than Protobuf message



Benchmark Source Code

- Protobuf

  https://gitlab.setlevel.de/deliverables/model/traffic-agents/osipedestrian/-/tree/ProtobufBenchmark

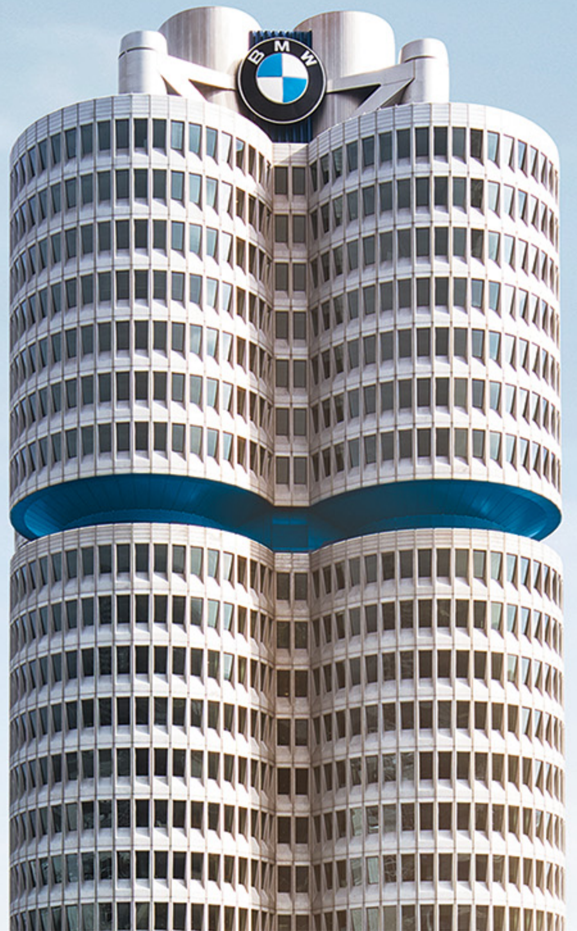
- Flatbuffers (using tables for osi_common.fbs)

  https://gitlab.setlevel.de/deliverables/model/traffic-agents/osipedestrian/-/tree/FlatbufferBenchmark-table

- Flatbuffers (using structs for osi_common.fbs)

  https://gitlab.setlevel.de/deliverables/model/traffic-agents/osipedestrian/-/tree/FlatbufferBenchmark-struct

THANKS FOR YOUR ATTENTION!



EN-50 | 26.04.22
Fabian Pfeuffer