

Enabling Fog Computing over Delay/Disruption-Tolerant Networks

*Original*

Enabling Fog Computing over Delay/Disruption-Tolerant Networks / Castellano, Gabriele; Risso, FULVIO GIOVANNI OTTAVIO. - ELETTRONICO. - (2018). (Intervento presentato al convegno 4th Italian Conference on ICT for Smart Cities And Communities tenutosi a L'Aquila, Italy nel September 2018).

*Availability:*

This version is available at: 11583/2712605 since: 2018-09-11T22:58:23Z

*Publisher:*

CINI

*Published*

DOI:

*Terms of use:*

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

*Publisher copyright*

(Article begins on next page)

# Enabling Fog Computing over Delay/Disruption-Tolerant Networks

Gabriele Castellano, Fulvio Risso

Politecnico di Torino, Dept. of Computer and Control Engineering, Torino, Italy

**Abstract**—Fog computing enables a multitude of resource-constrained end devices (e.g., sensors and actuators) to benefit from the presence of fog nodes in their close vicinity, which can provide the required computing and storage facilities instead of relying on a distant cloud infrastructure. However, guaranteeing stable communication between end devices and fog nodes is often not trivial. Indeed, in some application scenarios such as mining operations, building sites, precision agriculture, and more, communication occurs over Challenged Networks e.g., because of the absence of a fixed and reliable network infrastructure. This paper analyzes the applicability of Fog Computing in a real *Industrial Internet of Things* (IIoT) environment, providing an architecture that enables disruption-tolerant communication over challenged networks and evaluating the achieved performance on an open-source prototype implementation.

## I. INTRODUCTION

Fog Computing extends the Cloud paradigm to the edge of the network through the deployment of so called *fog nodes*, thus enabling end devices (e.g., IIoT devices) to benefit from computing and storage facilities in their close vicinity.

However, in scenarios such as *Industrial IoT* (IIoT) [1], guaranteeing stable communication between end devices and fog nodes is not trivial. Indeed, operations are often conducted in challenged environments, e.g., remote mines, building sites, large fields in smart agriculture, and more. In these scenarios, many IIoT application would benefit from the possibility of constantly exchanging data with remote fog nodes (e.g., delivery sensors measurements or receive firmware updates), although tolerating high latency and unreliable environments.

Our work investigates the applicability of Fog Computing in a real IIoT environment. To this end, we propose an architecture that enables communication between end devices and fog nodes without requiring a fixed network infrastructure. We focus a smart agriculture scenario, where a fleet of machineries sends data and receive updates to/from the fog nodes over the widely used MQTT publish/subscribe protocol [2], commonly adopted by many IIoT applications. Our approach aims at keeping applications agnostic with respect to the network connectivity, thus potentially enabling an entire class of existing applications to operate also on unconventional scenarios such as challenged networks.

To enable data propagation, our approach exploits occasional contacts occurring between machineries during their movements to establish *opportunistic connections*; data is exchanged during these device-to-device contacts and conveyed toward the destination by means of the *store-carry-and-forward* paradigm, implemented by a Delay/Disruption Tolerant Network (DTN).

Performance of our approach has been evaluated through measurements over our prototype implementation of a proof-

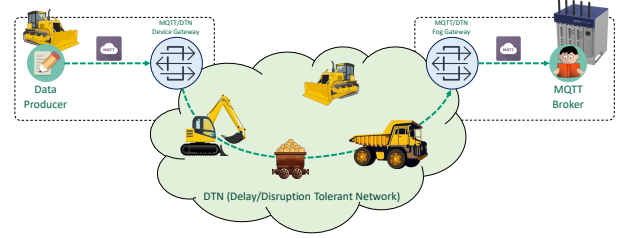


Fig. 1: Overall communication architecture: data is wrapped in DTN bundles and propagated through the *store-carry-and-forward* paradigm.

of-concept telemetry system [3], finding encouraging results and demonstrating advantages and applicability of this paradigm in our target IIoT environment.

## II. COMMUNICATION SYSTEM ARCHITECTURE

Both fog nodes and end devices (i.e., operating machinery) are modeled as nodes of a DTN. Thus, the *store-carry-and-forward* paradigm exploits the continuous movement of the operating machinery to enable data flowing toward the destination. Data is delivered by the DTN protocol, which extends the existing network stack (e.g. Ethernet/IP/TCP, Bluetooth, etc.) with a *bundle protocol* layer that encapsulates application messages; these are delivered hop-by-hop to another DTN node based on tunable forwarding behaviors, such as *epidemic* (data is replicated on all encountered nodes), *prophet* (data is forwarded only in the direction that looks the best), and more. Nodes of the DTN are identified by an Endpoint Identifier (EID), while each application is identified by extending the local EID with an *application token*.

An high level view of our communication architecture is depicted in Figure 1; in particular, it shows the communication between an originating vehicle (on the left) and the target fog computing node (on the right), which hosts the MQTT broker. In order to deliver data produced by a vehicle to the MQTT broker, each vehicle sends a data bundle to other peers through a small-range wireless connection. This process is repeated by the second vehicle when it detects other opportunistic connections, until the data carrier enters in range of the fog node, hence the data bundle is delivered to the destination (together with other bundles possibly collected in the meanwhile).

As shown in the figure, each node features a *gateway* that conveys MQTT messages over the DTN in order to guarantee protocol transparency to end-to-end applications, namely to allow them to interact with the MQTT primitives without being aware of the underlying (challenged) network.

Figure 2 details the architecture components in case of a Device-to-Fog (upstream) communication, i.e., when data flows from end devices to the fog node. Each end device hosts an *MQTT Publisher*, i.e., an application generating data

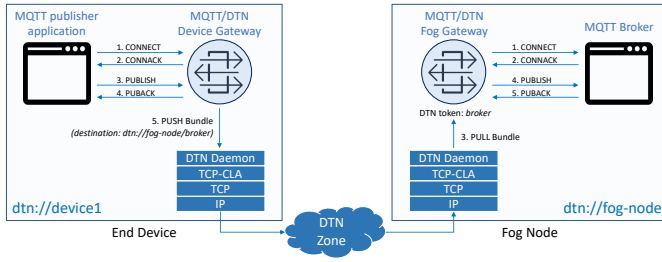


Fig. 2: Upstream communication system overview.

(e.g., through sensors), and an *MQTT/DTN Device Gateway*, that transparently sends MQTT messages on the DTN stack (PUSH Bundle operation in the figure), using as destination the fog node EID, which is extended with the broker application token. The fog node hosts an *MQTT/DTN Fog Gateway*, that pulls from the DTN bundles having destination `dtn://fog-node/broker` and forwards enclosed MQTT messages to the broker.

The Fog-to-Device (downstream) communication (namely, when devices subscribe to receive data) features a similar configuration, but the role of the two gateways is inverted. However, although in this case end devices act as MQTT subscribers (i.e., the receive messages from the broker), an initial data flow in the upstream direction is also needed to propagate the *subscribe* messages toward the broker. Finally, in this case the MQTT/DTN Fog Gateway has to duplicate bundles before pushing them in the DTN, since each message may be delivered to multiple destinations (a.k.a. subscribers).

In both cases, the MQTT/DTN Gateway enables MQTT applications to send and consume data using the traditional MQTT protocol, without being aware of connectivity issues and of how messages are propagated.

The proposed architecture has been used to design a telemetry system addressing a real use case scenario: a fleet of vehicles is equipped with sensors that, periodically, measure temperature, humidity and acidity of the ground; the stream of data generated this way is collected to a nearby fog node, where some applications perform preliminary analytic and aggregate data for the Cloud. The prototype extends the commercial Fog solution provided by Nebbiolo Technologies [4].

### III. EXPERIMENTAL RESULTS

The evaluation tests, performed on our proof-of-concept telemetry system implementation, have been conducted on a setup with 15 devices generating sensing data every 5 seconds, and compare different significant configurations of (i) probability of connections among devices, and (ii) duration of these connections. The probability of a connection with the fog node has been set to 10%, while the lifetime of a bundle in the DTN is 5 minutes. Results have been collected over multiple runs of 30 minutes each.

Figure 3 shows results concerning the bundle delivery evaluation. Particularly, Figure 3a shows that, if the inter-device connections last at least 2 seconds, almost all bundles are successfully delivered to the fog node. Figure 3b shows

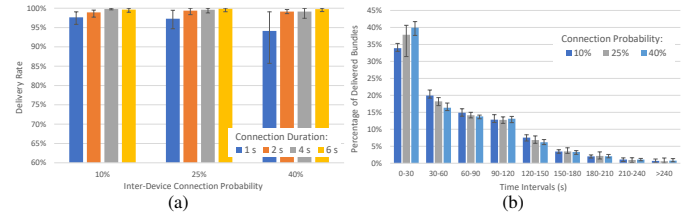


Fig. 3: (a) Bundle delivery rate varying the probability and duration of inter-device connections. (b) Distribution over time of delivered bundles.

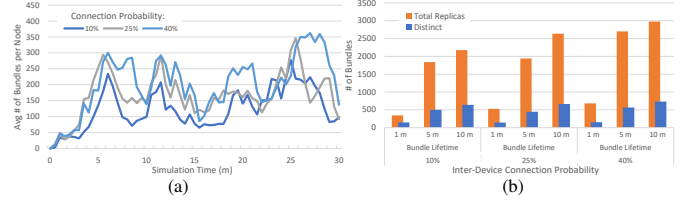


Fig. 4: (a) Average number of bundle stored on each device during the simulation and (b) average storage overhead for different configurations.

instead the delivery distribution over time: an higher connection probability increases the percentage of bundles that are delivered in the first 30 seconds; however, the graph also shows that, in any case, more than 95% of bundles are delivered within 3 minutes from their generation.

On the other hand, Figure 4 shows the storage consumption on end devices: in Figure 4a is shown that, after fully operational, the average number of bundles carried from each device ranges between two values (for a connection probability of 10%, it ranges between  $\approx 100$  and  $\approx 250$ ). Since the average bundle size of our telemetry system is 260 bytes, no more than 150 KB are required on each device. Finally, Figure 4b shows, for different configurations, the average overhead (i.e., the number of distinct bundles compared with the number of total replicas in the network).

### IV. CONCLUSION

This paper presents an approach to employ Fog Computing over challenged environments, with focus on a real Industry IoT use case. In particular, this work proposes an architecture that enables a fleet of machineries to send data and receive updates to/from fog nodes over the MQTT protocol, largely adopted by widespread IIoT applications, also keeping applications agnostic toward the type of network connectivity. To enable data propagation over challenged networks, the proposed architecture makes use of the *store-carry-and-forward* paradigm, typical of Delay/Disruption Tolerant Networks (DTN).

Performance evaluation of the proposed communication architecture shown encouraging results, demonstrating advantages and applicability of this paradigm in IIoT environments.

### REFERENCES

- [1] L. Da Xu *et al.*, “Internet of things in industries: A survey,” *IEEE Transactions on industrial informatics*, vol. 10(4), pp. 2233–2243, 2014.
- [2] “Information technology – Message Queuing Telemetry Transport (MQTT),” International Organization for Standardization, Standard, 2016.
- [3] Fog over dtn repository. <https://github.com/netgroup-polito/for-over-dtn>.
- [4] Nebbiolo Technologies. Bridging the intelligence gap between the cloud and iot and end-points. [Online]. Available: <https://www.nebbiolo.tech>