Demo: Enabling In-Network Processing utilizing Nearby Device-to-Device Communication

The An Binh Nguyen, Christian Klos
Björn Richerzhagen, Ralf Steinmetz
KOM, TU Darmstadt, Germany
{firstname.lastname}@kom.tu-darmstadt.de

Christian Meurisch
TK, TU Darmstadt, Germany
meurisch@tk.tu-darmstadt.de

Patrick Lampe
University Marburg, Germany
lampep@mathematik.uni-marburg.de

***Abstract*—In disaster situations, relief work can be enhanced and facilitated by acquiring and processing distributed information. However, the communication and computation infrastructures might be impaired or inaccessible in emergency response scenarios. Consequently, approaches for coordination, and resource utilization are still challenging. To this end, we proposed the concept of an *adaptive task-oriented message template* (ATMT), that bundles the control information and the payload data, required to process and extract information, into a single message. Thus, an ATMT enables distributed in-network processing of complex tasks, allows to leverage the idle resources of mobile devices of the first responders. In this paper, we demonstrate the use of the ATMT concept in an example of face detection, which can be used to offer *Person Finder* similar services in an emergency ad hoc network. We utilize Google Nearby peer-to-peer networking API, standard and available on Android-based devices, to realize the handover of an ATMT message between mobile devices. This successful integration underpins the prospective adoption of the ATMT concept.**

I. Introduction

Acquiring and processing distributed information are crucial for disaster situations. Today, several services designed to enhance relief work, and to offer information relevant for emergency situations, such as Google's Person Finder [1] or Facebook's Crisis Reponse [2], are available. However, these services require stable Internet-based communication, which might not be possible in disaster situations. To maintain communication in emergency situations, mobile hand-held devices such as smart phones can be used to create an opportunistic ad hoc network [1], which allows mobile devices to share data and exchange information through device-to-device communication. Combining with the built-in sensors, and the computing resource available on mobile devices, it is possible to provide emergency relief services on top of mobile opportunistic ad hoc network. Hereby, approaches to enable distributed coordination, and efficient resource utilization are necessary. For this purpose, we proposed and designed a message template, called *adaptive task-oriented message template* (ATMT) in [2].

An ATMT message describes a complex task, the operations and the payload data required to complete the defined complex task; which makes each ATMT message a self-encapsulated message. Hence, the ATMT message template is able to support distributed processing, leveraging idle resource of the participating mobile devices. Additionally, since an ATMT message is self-encapsulated, each participating device can make an autonomous decision based on its available capability, and its available resources. Overall, the ATMT message provides a basis to create complex services on an opportunistic network, utilizing mobile devices.

To showcase the advantages, and the applicability of the ATMT concept in practice, we provide a demonstration, that (i) uses ATMT message template to implement a face detection technique, which requires multiple-processing stages, specially designed for mobile devices [3], and (ii) utilizes the Google Nearby Networking API [4] for enabling handover of an ATMT message directly between devices.

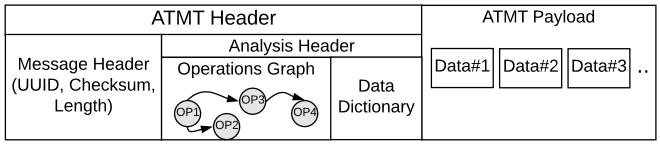

Fig. 1: Structure of the adaptive task-oriented message template (ATMT) as proposed in [2]

II. Adaptive Task-oriented Message Template

The design and construction of the ATMT message template are illustrated in Fig. 1. The goal in designing ATMT message template is to allow a user to define a processing goal, and the steps/operations required to reach this goal. These information are captured in an *operations graph*, modeled by a *directed acyclic graph* within the *ATMT header*. A device can check on the processing status of the *operations graph*, by reading only the *checksum* field, without reading the whole content of the ATMT message. This allows a device to make a fast decision on whether to merge, to drop, or to handover an ATMT task. To incorporate the payload data required for the operations, considering the resource constraints of the devices in an opportunistic mobile ad hoc network, we devise the *ATMT payload* to contain pieces of data, which can be compressed by different encoding methods to allow for more flexibility. Each piece of data is mapped to an operation

[1] https://google.org/personfinder
[2] https://www.facebook.com/about/crisisresponse/

in the *operations graph* through the *data dictionary* in the ATMT header. To organize the coordination among mobile devices, we conceive four roles, which can be assumed by the participating devices according to their available capabilities. These roles are *sensor node* to obtain data, *delegator node* with domain knowledge to set up and adapt the *operations graph* if necessary, *operator node* which executes one or more operations from the operations graph based on its available resource, and *forwarder node* which receives, store and forward an ATMT message. Distributed processing of a complex task is realized by simply passing ATMT messages. Each device, receiving ATMT messages, will act accordingly to its role, adjust the content of ATMT messages w.r.t. the current processing state, and handover the adjusted ATMT message to the other.

III. Integration of ATMT with Google Nearby

Support to setup and to provide device-to-device communication using mobile devices such as smart phones is still restricted. To enable WiFi ad hoc communication between Android devices, these are required to be rooted. Even though WiFi direct allows for direct communication, but it requires a master-slave model, which makes transmission of a message through multi-hops difficult [1]. Recently, Google enables and provides *Nearby* [4], a peer-to-peer networking API that allows for discovery, connection and data exchange with devices in the close vicinity. We use *Nearby Connections* to let each device advertise its role/service. Thus, each participating device is able to look for the next role/service, that it requires. For instance, a *sensor node*, after acquiring data, needs to look for a *delegator node*, so that the *delegator node* can setup and include the corresponding *operation graphs* into the ATMT message. Similarly, a *delegator node* searches for *operator nodes* to execute the operations on the obtained data. If an *operator node* cannot complete the task described in ATMT message alone, this *operator node* will look for further *operator nodes* which possess the capabilities, e.g., special hardware, special algorithms/libraries, to take over the upcoming operations. When an *operator node* notices the completion of the ATMT task, it can look for *forwarder nodes* to transport the final result to a predefined destination. In this manner, a multi-stage processing is realized through multi-hops device-to-device communication.

IV. Scenario and Demonstration

We use the ATMT concept to implement the face detection technique proposed in [3], which can be used to support *person finder* service in an emergency ad hoc network. The face detection technique as introduced in [3] relies on multiple stages pipeline to detect multiple faces within an image; these are (1) preprocessing to handle parameters, (2) a fast face detection to determine important regions within the image, and (3) a validation phase using *dlib* library to detect and validate the faces within the image.

Despite the fact, that the face detection technique in [3] is designed considering the resource and energy constraint of a mobile device, this technique can be further enhanced using the ATMT concept. The processing pipeline as described can be distributed to multiple devices; consequently, each device has to process only one phase of the pipeline. The advantages for the utilization of the ATMT concept in this scenario are (i) further reducing the resources consumption of the participating devices, since now each device has to process only one phase of the pipeline, and (ii) being able to leverage heterogeneous capabilities, for instance, in the described scenario, the *dlib* library might not available on all devices.

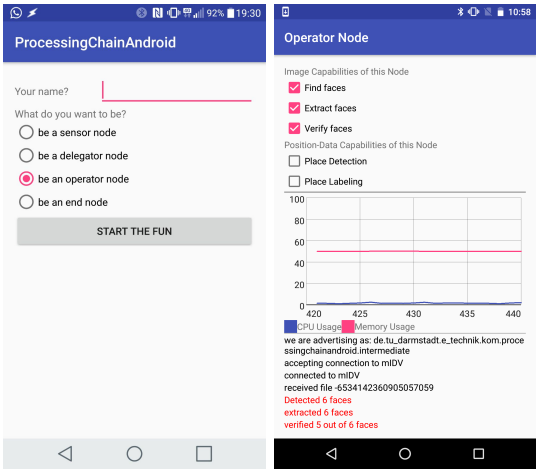

Fig. 2: User interface of the Android mobile devices used in the demonstration

In the demonstration, a user can use a *sensor* device to capture and send a picture containing multiple faces to other devices for further processing. Next, the device chosen as a *delegator node* should receive the picture, adds the *operations graph* for the face detection technique accordingly, and forwards the constructed ATMT message to the *operator devices*. On each *operator* device, the user can choose which operations should be available, and can observe through log messages, how the devices handle ATMT messages, and how the *Nearby Connections* are used (cf. Fig. 2). At the end of the processing pipeline, the results of the face detection can be seen in the chosen end-node device, showing the cropped images of different detected faces. A short video, showing the demonstration as described, can be found at the following link http://bit.ly/2GGenHZ

References

[1] S. Trifunovic, M. Kurant, K. A. Hummel, and F. Legendre, "Wlan-opp: Ad-hoc-less opportunistic networking on smartphones," *Ad Hoc Networks*, vol. 25, 2015.

[2] T. A. B. Nguyen, C. Meurisch, S. Niemczyk, D. Böhnstedt, K. Geihs, M. Mühlhäuser, and R. Steinmetz, "Adaptive Task-oriented Message Template for In-Network Processing," in *NetSys'17*. IEEE, 2017.

[3] P. Lampe, L. Baumgärtner, R. Steinmetz, and B. Freisleben, "Smartface: Efficient face detection on smartphones for wireless on-demand emergency networks," in *IEEE ICT*, 2017.

[4] "Google Nearby." [Online]. Available: https://developers.google.com/nearby/