Aditya Raj, Andreas Reinhardt: CLAP: Cooperative Locality-Aware Data Processing in Heterogeneous Fog Environments. In: Proceedings of the 1st GI/ITG KuVS Fachgespräch "Fog Computing" (FGFC), pp. 21-24, 2018.

CLAP: Cooperative Locality-Aware Data Processing in Heterogeneous Fog Environments

Aditya Raj
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
aditya.raj@tu-clausthal.de

Andreas Reinhardt
Technische Universität Clausthal
Clausthal-Zellerfeld, Germany
reinhardt@ieee.org

***Abstract*—Wireless Sensor Networks deployed in recent years often share the commonality of relying on homogeneous hardware platforms. The upcoming vision of the Internet of Things, however, is strongly based on the co-existence of embedded devices manufactured and operated by different stakeholders. Thus, device heterogeneity will become an omnipresent characteristic of future sensor deployments, particularly within the scope of fog computing. In this paper we present CLAP, a collaborative data processing approach that exploits device heterogeneity for collaborative data processing instead of trying to mitigate its negative effects. Through simulations, we demonstrate that CLAP, in conjunction with a data collection protocol, can effectively reduce the volume of traffic and thus cater to energy savings and consequently prolonged network runtimes.**

I. INTRODUCTION

In the past decade, Wireless Sensor Networks (*WSN*s) have shown tremendous potential for application domains like disaster relief, healthcare, or intelligent buildings [1]. Based on low-power embedded sensing systems with a limited energy budget and low computational power, WSNs commonly gather information from distributed locations and collect them at devices on the network edge (often referred to as *base stations*, *BS*), for processing. A dedicated family of routing protocols has emerged to accommodate this traffic flow: *Collection protocols*, such as MultiHopLQI [2] or CTP [3]. However, one of the key limitations of data collection WSNs is their concentration of wireless traffic around the base station, leading to a quicker energy depletion of nodes located there. To cater for a more balanced use of energy inside the WSN, in-network processing techniques have been proposed, e.g., distributed outlier detection [4], clustering [5], or aggregation [6]. When executed close to the data collection points, distributed data processing reduces energy demand and wireless traffic alike.

Along with the vision of the upcoming Internet of Things, sensor nodes *(SNs)* can be expected to carry more diverse traffic than traditional WSNs. Moreover, device heterogeneity in terms of computational power, storage capacity, wireless range, or the available energy budget will become omnipresent characteristics. Despite the technological support for advanced in-network processing operations, however, current systems generally lack the option for a seamless data exchange on application level, and do not leverage heterogeneity at all. By presenting CLAP, we lay the foundation for a better utilization of device heterogeneity by allowing heterogeneous SNs to *cooperate* and harness the in-network data processing capabilities of dedicated Processing Devices (*PD*s). In a nutshell, CLAP allows SNs to find PDs in their vicinity, exploit their heterogeneous computational capabilities for in-network processing, and subsequently forward processed data to their destination.

This work builds on a theoretical energy assessment of heterogeneous networks in own prior work [7]. However, we make the following two novel contributions in this paper: (1) We introduce the concept of Cooperative Locality-Aware Data Processing (CLAP), which enables SNs to actively cooperate, and present selected implementation details in Sec. II. (2) We evaluate CLAP and demonstrate its efficacy in two simulated scenarios in Sec. III. We conclude this paper in Sec. IV.

II. OBJECTIVES AND APPROACH

Our primary research objective is to assess to what extent the presence of computationally heterogeneous nodes in a WSN deployment can lead to a reduction of communications overhead when decentralized data processing takes place. An example application scenario is shown in Fig. 1. Two data sources (DS_1 and DS_2) are part of a collection tree, rooted at the BS. While DS_1 would normally forward its data via the dashed link towards the sink, the availability of a processing device in its one-hop neighborhood changes this route. Its data are consequently routed to processing device PD_1, where they are processed and only the result is forwarded along the established collection tree. In case of DS_2, the next PD is two hops away. An intermediate node is used as a relay device and configured to forward incoming traffic from DS_2 to PD_2, which in turn uses the underlying collection protocol to forward its processed data to the BS.

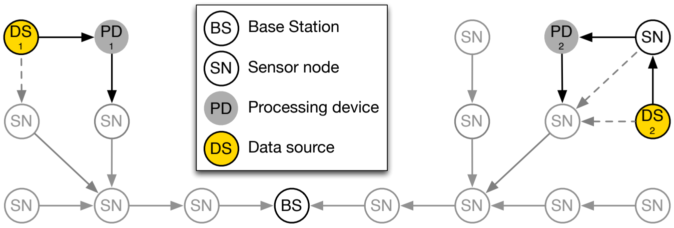


Fig. 1. Two sample application cases for the proposed collaborative data processing approach. Continuous arrows indicate the resultant collection tree, while dashed arrows show the data flow without collaborative data processing.

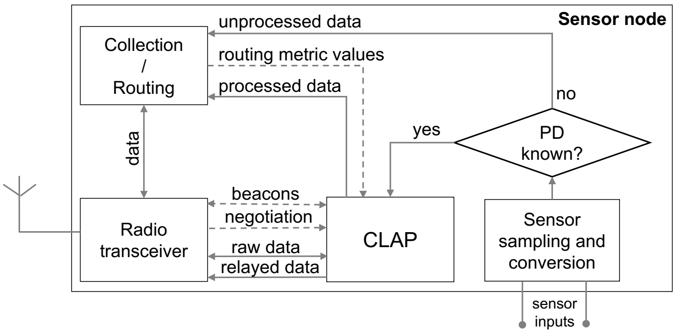


Fig. 2. Sensor nodes' network stack to realize collaborative data processing.

A. Requirements Analysis

To enable heterogeneous devices on the IoT to collaboratively process collected data in their locality, the following fundamental requirements need to be fulfilled:

1) SNs must be able to determine whether any PDs exist within their neighborhood. To avoid excessive deviations from the collection route to the BS and to eliminate the complexity of including an additional routing protocol, CLAP does not consider PDs more than two hops away from a SN.
2) SNs must be able to establish a path to one of the PDs in their vicinity (directly or via an intermediate node) and send their sensor data along this route instead of forwarding them to the BS using the collection tree.

B. Overview of CLAP

A conceptual overview of CLAP's integration with the data flow on a regular sensor node is shown in Fig. 2. If a PD is known and available for collaborative data processing, the CLAP module takes care of data routing and processing. However, if CLAP finds no PD in the node's vicinity, the SN's data flow remains identical to the case of running a collection protocol without CLAP. The functionalities required to accomplish the seamless collaborative in-network processing are detailed in the following subsections.

C. Finding Processing Devices

In order to utilize data processing capabilities of PDs, SNs must become aware of the presence of any PDs in their two-hop neighborhood. CLAP relies on proactive advertising of PDs through beacons, which leads to less announcement traffic in case the number of SNs exceeds the number of PDs in the network. The identification of PDs and their capabilities is accomplished on demand through probing messages to ensure a reliable communication can be established with the PD even when its announcements have been received a while ago.

In CLAP, PDs periodically send advertisement beacons to make their availability known to their neighborhood. Beacons are re-broadcast once by the nodes in the immediate neighborhood of the PD and can thus be assumed to reach all two-hop neighbors. These advertisement beacons contain three fields: the network address of the PD itself, its routing metric value according to the collection protocol (e.g., the ETX to the BS), and the address of a relaying node. The latter field is initially empty, and only populated when the advertisement beacon is re-broadcast by a PD's neighboring device. Knowledge of the PD's routing metric assists in selecting the best PD for the data processing, as detailed in Sec. II-D. However, the routing metric can easily be substituted by other desired quality constraints to improve the PD selection process.

Each node maintains a list of the PDs from which advertisements have been received. Upon reception of an advertisement beacon from a PD or a relaying node, the receiving node updates its list of PDs accordingly. A threshold for the minimum number of beacons received from a PD is being used in order to only enter PDs with stable network links into the list. Stale members are periodically purged from the list.

D. Selecting a Processing Device

In order to choose the processing device for locally collected data, all entries in the PD list are sorted based on their distance to the local device (i.e., hop count) and their routing metric value of the underlying collection protocol. The main reason for using this sorting technique is to ensure that locality is preferred. Relaying nodes to accomplish two-hop transmissions are only used in case no PD exists in the SN's direct neighborhood.

E. Negotiating Processing Data Transfers

Data transfers to a PD are always preceded by a request-reply handshake, initiated by the SN in need of in-network data processing. A request packet is sent to the selected PD, with the network addresses of the data source and the requested PD contained in its payload; if a relaying device is required to transfer data from the source to the PD, its address is also explicitly specified. Besides addressing information, the request message contains the type of the requested processing operation and the temporal duration for which the processing is needed. When a PD receives the request message and is ready to accept the client, it acknowledges the reservation by sending a reply message to the sender. The handshake is completed through a final reply message, confirming the successful resource reservation, which carries the same information fields as the request packet. In case a PD does not respond within a timeout period, the SN proceeds to negotiate collaborative data processing with the next entry in its PD list. Once the negotiations of processing device, duration, and function have been completed, the data transfers can commence.

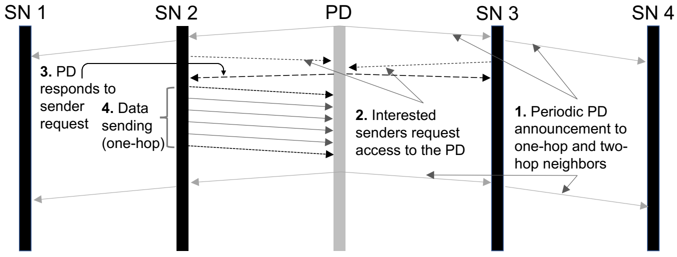
This separate transfer handshake, executed before actual data transmissions, is implemented for three reasons: Firstly, to ensure availability of desired PD to receive and process data. Secondly, to allow the PD to perform computations on the incoming data based on the desired service requested through the handshake messages. Thirdly, to configure an intermediate device (if SN and PD are only within two-hop distance of each other) to relay all following data from the SN to the PD. In case a preferred PD is not ready to process data or cannot offer the requested processing service, the handshake does not complete successfully, and another PD will be approached.

F. *Exchange of Application Data*

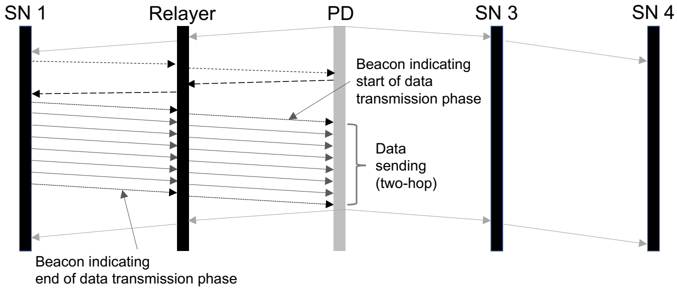
In CLAP, data transmissions between an SN and its selected PD are always preceded by the aforementioned handshake to indicate the start of a data transmission, and terminated by a final delimiting message. By encapsulating data transmissions this way, PDs can schedule the use of their processing resources better. On reception of end-of-transmission packet, the PD performs the requested computation on the data received so far and forwards the resulting packet over the underlying collection tree on behalf of its original sender. CLAP does not explicitly provide flow or congestion control, but is interoperable with transport protocols offering such features.

G. *Typical Communication Flow*

Figs. 3a and 3b show typical communication flows when running CLAP. At first, the PD's announcements of its processing abilities to its one-hop neighborhood are visible. These messages are then re-broadcast by all immediate neighbors to publicize the PD's existence to its two-hop neighbors. Subsequently, a data processing request is made by SN 2 (in Fig. 3a), or SN 1 (in Fig. 3b), respectively. Depending on the number of simultaneously allowed data processing operations, the PD can individually determine whether to accept a request or not. In the figures, the PD is configured to provide processing services to at most one SN at a time. Consequently, the successful handshake completion message sent to SN 2 in Fig. 3a acts as a trigger to initiate data transmissions, and simultaneously serves as a negative acknowledgment for SN 3; therefore, SN 3 continues sending unprocessed data using the underlying data collection protocol. Data transfer takes place through unicast transmissions (possibly involving a relay node, shown in Fig. 3b) and ends with a delimiter message from the sender.



(a) Typical flow of exchanged messages for one-hop data transmissions.



(b) Typical flow of exchanged messages for two-hop data transmissions.

Fig. 3. Communication flows for one-hop and two-hop collaborative data processing using CLAP.

III. EVALUATION

We present our evaluation results after outlining the experimental setup and introducing the network topologies analyzed.

A. *Evaluation Setup*

Across all of our evaluations, we assume a heterogeneous network setting, in which several SNs and PDs exist, and the BS is the recipient of all data collected on the sensing devices. We use the Collection Tree Protocol (CTP) [3] as the underlying collection protocol because of its reliability and its available implementation for the TinyOS operating system. CTP forms an acyclic tree, based on adaptively updated ETX values, to relay data to the BS. Although the ETX values can be used in CLAP to sort PDs for better performance and reliability, their availability is not mandatory for CLAP's operation.

All following evaluations have been conducted using the COOJA simulator [8], using the *tmote sky* [9] as the SN. The maximum number of clients a data processing device can serve simultaneously has been set to one. This ensures a higher competition for PDs in our analysis.

Heterogeneous processing capabilities in the simulation have been realized by adding a data processing service on a small fraction of the simulated devices. Although the demanded service for data processing by SNs is always a summation of all received values (generated as a 16-bit random payload every 100 ms at every SN), more complex algorithms can be substituted easily. CLAP is agnostic to the choice of processing algorithms offered by PDs, thus we do not investigate different algorithms in this research work.

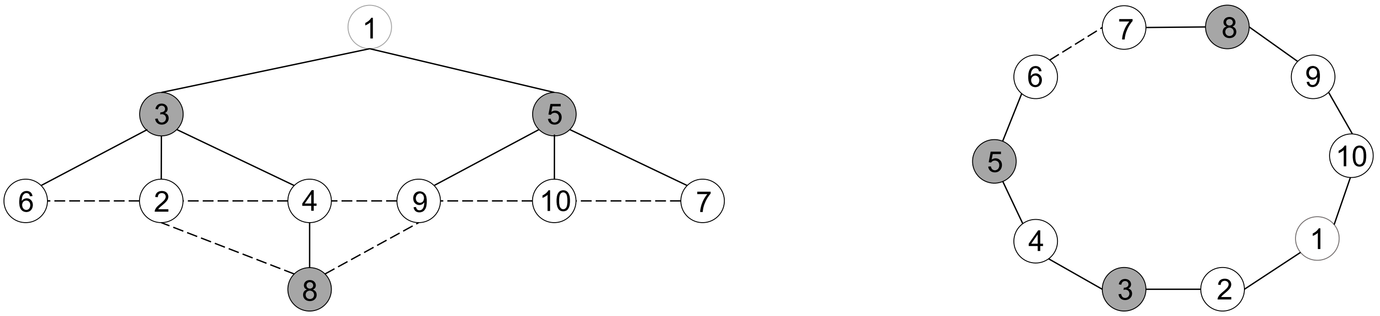
B. *Simulated Topologies*

We run our simulations in two topologies, shown in Fig. 4, which demonstrate differently expected traffic flows. In both scnearios, the devices with identifiers 3, 5, and 8 are configured as PDs, and node 1 serves as the base station. Continuous lines indicate the tree formed by the collection protocol, whereas dashed lines show additional wireless connectivity. In the tree topology (shown in Fig. 4a), each data generating SN, has at least one PD in its direct neighborhood, and most even find two PDs in their two-hop neighborhoods. In contrast, in the ellipsoid topology in Fig. 4b, each SN has exactly two neighbors in one-hop distance.

C. *Advantages of using CLAP and CTP*

Experiments were performed to assess how much messaging overhead can be saved by performing computations on PDs instead of forwarding unprocessed data to the BS. For this, we set up two simulations, one with only CTP is enabled, and another one with both CLAP and CTP enabled. The number of packets received at the base station in 25 minutes of simulation time at different data generation rates is shown in Fig. 5. In all simulations, SNs have been configured to make reservations for data processing operations for durations of 12–20 seconds.

The maximum difference in the number of packets received at the BS can be observed when a high volume of data packets



(a) Tree topology.

(b) Ellipsoid topology.

Fig. 4. Network topologies used in our analyses. The node with identifier 1 always serves as the base station; devices shaded in grey are processing devices.

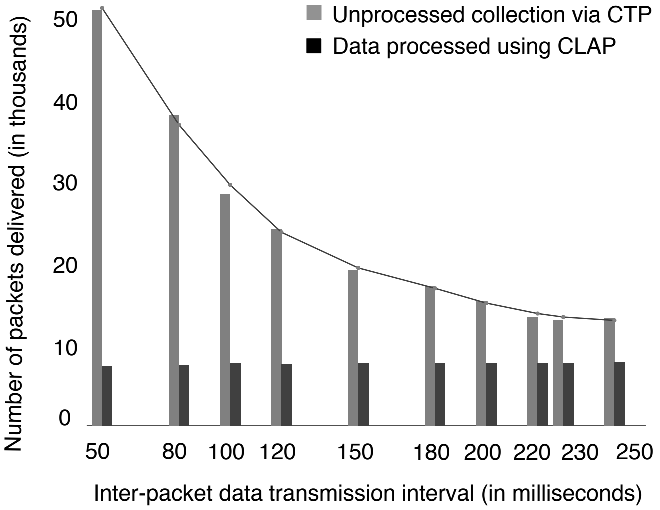


Fig. 5. Comparison of the number of packets delivered to the base station.

are being generated. In fact, more than seven times as many packets are received at the BS when running CTP as compared to the use of CLAP for a 50 ms data transfer interval. While the absolute difference in terms of the number of received packets shrinks with growing inter-packet intervals, collecting unprocessed data via CTP always results in a higher number of messages being delivered to the BS. In contrast, the number of processed data packets is almost constant when CLAP is being used, as collection packets are only emitted whenever a processing operation has terminated. Even though this is the expected result from shifting data processing operations into the network and allowing for their collaborative execution, the reduction of traffic directly translates into energy savings. It thus confirms CLAP's efficacy and its viability to combat energy holes around the BS.

IV. CONCLUSION

The IoT is growing fast and expected to be composed of billions of heterogeneous systems [10]. However, even when IoT devices use the same wireless communication standards, they suffer from a lack of application-level interoperability. This is especially important in the domain of fog computing, where IoT devices will be interfaced with a broad range of other systems. In this paper we have proposed CLAP as a method to overcome this limitation. CLAP runs on top of any tree-based routing or data collection protocol (such as CTP or RPL) and provides an abstraction from system heterogeneity. Simulations comparing CLAP to a setting without in-network data processing show the high potential of exploiting device heterogeneity to reduce the number of packets delivered to the base station. Even though heterogeneity is often seen as an obstacle, we have shown the potential of exploiting it in data collection sensor network deployments instead of trying to mitigate its effects. CLAP has performed demonstrably well for data aggregation; we consider investigations into other distributed data processing algorithms and the scalability of the algorithm to different topologies as future work.

REFERENCES

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, no. 4, 2002.

[2] R. Fonseca, O. Gnawali, K. Jamieson, and P. Levis, "TEP119: Collection," Online: https://github.com/tinyos/tinyos-main/blob/master/doc/txt/tep119.txt, 2006.

[3] O. Gnawali, R. Fonseca, K. Jamieson, M. Kazandjieva, D. Moss, and P. Levis, "CTP: An Efficient, Robust, and Reliable Collection Tree Protocol for Wireless Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 10, no. 1, 2013.

[4] J. W. Branch, C. G. amd Boleslaw Szymanski, R. Wolff, and H. Kargupta, "In-network Outlier Detection in Wireless Sensor Networks," *Knowledge and Information Systems*, vol. 34, no. 1, 2013.

[5] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences (HICSS)*, 2000.

[6] C.-M. Chao and T.-Y. Hsiao, "Design of Structure-free and Energy-balanced Data Aggregation in Wireless Sensor Networks," *Journal of Network and Computer Applications*, vol. 37, 2014.

[7] A. Reinhardt and D. Burgstahler, "Exploiting Platform Heterogeneity in Wireless Sensor Networks by Shifting Resource-Intensive Tasks to Dedicated Processing Nodes," in *Proceedings of the 14th IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, 2013.

[8] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA," in *Proceedings of the 31st IEEE Conference on Local Computer Networks (LCN)*, 2006.

[9] Memsic Inc., *TelosB data sheet*. [Online]. Available: http://www.memsic.com/userfiles/files/Datasheets/WSN/telosb_datasheet.pdf

[10] D. Evans, "The Internet of Things – How the Next Evolution of the Internet is Changing Everything," Cisco Internet Business Solutions Group (IBSG) White Paper, 2011.