A Self-Organizing SDN Architecture for Mobile Tactical Edge Networks

Geng Li[‡], Qiao Xiang[‡], Christopher Dearlove[*], Y. Richard Yang[‡]
[‡]Yale University, [*]BAE Systems
{geng.li, qiao.xiang, yang.r.yang}@yale.edu, chris.dearlove@baesystems.com

***Abstract*—Through the separation of control and data planes, software defined networking (SDN) allows a network to be effectively controlled via a logically centralized controller with a global network view. Researchers have started to explore the potentials for introducing SDN to mobile tactical edge networks (MTEN), where moving devices communication through wireless channels. However, SDN was originally developed for wired infrastructures. Its efficiency in MTEN is substantially jeopardized due to the unpredictable state of wireless links and frequent topology changes. In this paper, we design a novel hierarchical, self-organizing SDN architecture for mobile tactical edges. In this architecture, a network is dynamically partitioned into multiple temporary domains in a self-organizing manner. In each domain, a certain node is appointed as a local SDN controller for managing all other nodes in the domain. To realize this architecture, we design a self-organizing network protocol, which includes a neighborhood discovery mechanism, a distributed game-theoretic network partition algorithm, and an abstraction of dynamics. It supports the dynamic adjustment of network partitions and the local controller placement for each domain with a low (re)configuration time, hence ensures the efficiency of the hierarchical SDN architecture in MTEN.**

I. Introduction

Software defined networking (SDN) brings significant benefits for network management by separating the control plane and data plane of a network and allowing it to be controlled by a logically centralized controller. Though SDN has been well studied in the context of wired networks, it was only until recently when people start to investigate its potentials in mobile tactical edge networks (MTEN) [1], [2].

Directly applying the SDN architecture to MTEN, however, is impractical due to some key differences between wired networks, where SDN was originally designed for, and MTEN [3]. In particular, a MTEN is highly dynamic due to the unpredictable state of wireless links and frequent topology changes, while wired networks are relatively stable. As a result, the control path traffic of traditional SDN architecture in MTEN is substantially heavier, jeopardizing the efficiency of network management and the performance of data traffic. What is worse, the overhead to maintain connectivity between the controller and all communication devices in MTEN is too expensive, if not impossible.

In this paper, we propose a novel hierarchical, self-organizing SDN architecture tailored for MTEN. This architecture has two novel features: (1) a network can be dynamically partitioned into multiple temporary domains in a fully decentralized and self-organizing manner; and (2) in each domain, one node is assigned as a temporary local controller to manage all other nodes in the domain. To realize this architecture, we design a self-organizing network protocol. This protocol is composed of a neighborhood discovery mechanism, a distributed game-theoretic network partition algorithm, and two procedures to abstract all network dynamics. It supports the dynamic network partitioning and the local controller placement for each domain with a low (re)configuration time, hence ensures the efficiency of the hierarchical, self-organizing SDN architecture in MTEN.

II. Overview of the Self-Organizing SDN Architecture

We consider an MTEN that consists of one logical controller (central controller) in the control plane and a set of mobile assets (nodes) with communication capabilities, *e.g.*, UAVs, robots, vehicles and etc., in the data plane. Nodes communicate with each other and the controller through wireless channels. Nodes are distributed at geographical locations. Different nodes have different transmission powers depending on their hardware specifications. As a result, each node may only be able to directly communicate with a certain number of nodes (neighbors).

Limitations of existing architectures. If the traditional SDN architecture is applied to an MTEN, the central controller will install the optimal data forwarding decisions to mobile nodes by directly communicating with mobile nodes. However, this design is neither scalable or efficient due to the dynamics of wireless channels and network topology in MTEN. A strawman approach trying to improve its limitations is to pre-partition the network into several domains, and place a sub-controller in each domain as a relay to forward data forwarding decisions from the central controller to all the nodes in this domain. However, the volatile mobility of communication devices in MTEN determines that this approach will also perform poorly in MTEN.

Basic idea. Our hierarchical, self-organizing SDN architecture for MTEN is motivated by protocols for mobile ad hoc networks, *e.g.*, OLSR. In particular, our architecture asks mobile nodes to dynamically partition the network into multiple temporary domains in a self-organizing manner, without any coordination from the central controller. In this way, the network partitioning functionality is offloaded from the control plane to the data plane, substantially reducing the control path traffic under the presence of high dynamics of wireless channels, the node mobility and the network topology.

In each domain, a node is elected as the local controller node (CN). This local controller serves more than a simple relay between the central controller and other nodes in the domain. Instead, certain controller functionalities are further offloaded from the central controller to each local controller. In this way, each local controller is in charge of the management of all other nodes, which are also called member nodes (MN), in the same domain.

Every CN utilizes the OpenFlow communication protocol to compute and manage the data forwarding decisions of its MNs.

Depending on their computation capacities, different CNs may be assigned functionalities. If a CN is equipped with sufficient computation power, *e.g.*, a vehicle CN, it can calculate the optimal data forwarding decisions by itself and hence take most of the control functionalities from the central controller to manage the domain it belongs to. On the contrary, if a sensing or actuator device is elected as the CN, it has a limited computation power, hence it can only take very few control functionalities from the central controller, *e.g.*, simple stateful firewall update.

Compared with controller nodes, MNs do not make any routing decision. They simply follow the data forwarding decisions sent by their local controller to forward and drop packets. Depending on different network partitions, MNs may maintain the connections with the local CN and the central controller (or other physical controllers) simultaneously to improve the fault tolerance of the control path. The synchronization between different controllers is a key piece of this architecture, and currently under investigation.

III. A Self-Organizing Protocol for Dynamic Network Partitioning

To realize the hierarchical, self-organizing SDN architecture, we design a self-organizing protocol for dynamic network partitioning. In the following, we present three key components of this protocol.

Neighborhood discovery. The neighborhood discovery mechanism in self-organizing SDN is based on a HELLO beacon message transmitted from each node. In each node, the HELLO message is broadcast periodically with a unified transmission power. Each node listens to the broadcast channel at all time and measures the received signal strength power of different beacon signals, which will then be used to calculate the link condition and bandwidth requirements. The HELLO message is a 4-tuple $(ID, CN_p, \{MN_p\}, MNCP)$, which denotes the sender node ID, the potential CN, the list of potential MNs and the MNCP value for CN_p defined in Definition 1.

Definition 1 (MNCP)**.** The *maximum number of coexisting peers (MNCP) that can be tolerated* is defined as α_i, which represents that given the control plane bandwidth and adopting the proportional fair model, if a node wants to get associated with node i as his CN, the domina size can not exceed α_i.

Domain partitioning algorithm. Unlike the clustering algorithms based on density or mobility in ad-hoc protocols, our idea is to formulate the dynamic partitioning as a *hedonic coalition game* where we adopt a simplified but efficient measurement of the utility (MNCP) for the control plane. Our game-theoretic, Nash stable-based user association scheme is a local optimal solution to minimize the control plane cost. It consists of four steps as follows.

1) Based on the link costs from neighbors, each node selects a potential local controller CN_p with the minimum cost and calculates the value of MNCP.
2) Based on the neighbors' HELLO messages, each potential CN selects the effective game players, and estimates the number of associated MNs at the Nash-stable state k^{NS} which is an indicator of the expected load condition.
3) Based on the estimated value k^{NS}, each node performs admission control to keep the first k^{NS} players with the highest values of MNCP, and stores them in $\{MN_p\}$.
4) If the set $\{MN_p\}$ includes a node in the HELLO message of his potential CN, the node then gets associated with the CN; otherwise, the node selects another potential CN.

We prove the convergence of this partitioning process by the following theorem.

Theorem 1. *Starting from at any initial states, all players will always end up with a convergence to a final Nash-stable state, in which any movement of a player will lead to a utility reduction, with the domain size k^{NS} obtained below.*

$$k^{NS} = \inf_k k < \alpha_i \quad \forall i = 1, 2, ..., k. \tag{1}$$

Proof. The detailed proof can refer to the paper in [4]. □

Abstraction of dynamics. We define two procedures, namely, *user handoff and controller substitution* to abstract network dynamics. Specifically, all dynamic in the network can be captured and abstracted as either one or a combination of these two procedures. User handoff happens when an MN moves into another domain, *i.e.*, the MN finds it more likely to join another CN based on the partitioning algorithm. In that case, the MN will get associated with the target CN before breaking the current one, so that a lossless control plane is sustained. Controller substitution takes place when an MN intends to become a new CN or one of the existing CNs gets crashed. The central controller that stores the duplicated domain database and routing policies, will then send the information to the newly substituted CN.

IV. Conclusion and Future Work

We design a novel hierarchical, self-organizing SDN architecture for MTEN. In this architecture, the network is dynamically partitioned into multiple domains in a decentralized manner, and a certain node in each domain is appointed as a local controller with certain control functionalities. We design a self-organizing protocol, which includes a game-theoretic network partitioning algorithm, to realize this architecture. As future work, we are building a testbed that integrates SDN functionalities into mobile devices to evaluate our proposed architecture.

Acknowledgment

This research was sponsored by the U.S. Army Research Laboratory and the U.K. Ministry of Defence under Agreement Number W911NF-16-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defence or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation hereon.

References

[1] T. Luo, H.-P. Tan, and T. Q. Quek, "Sensor openflow: Enabling software-defined wireless sensor networks," *IEEE Communications Letters*, vol. 16, no. 11, pp. 1896–1899, 2012.

[2] H. I. Kobo, A. M. Abu-Mahfouz, and G. P. Hancke, "A survey on software-defined wireless sensor networks: Challenges and design requirements," *IEEE Access*, vol. 5, pp. 1872–1899, 2017.

[3] K. Phemius, J. Seddar, M. Bouet, H. Khalifé, and V. Conan, "Bringing sdn to the edge of tactical networks," in *Military Communications Conference, MILCOM 2016-2016 IEEE*. IEEE, 2016, pp. 1047–1052.

[4] G. Li, Y. Zhao, and D. Li, "Game-theoretic user association in ultra-dense networks with device-to-device relays," *Wireless Personal Communications*, vol. 95, no. 3, pp. 2691–2708, 2017.