Software-Defined Architecture for Infrastructure-less Mobile Ad Hoc Networks

Ayush Dusia and Adarshpal S. Sethi
Department of Computer and Information Sciences, University of Delaware, Newark, Delaware, 19716, USA
adusia@udel.edu, sethi@udel.edu

***Abstract*—Most existing Software-Defined Networking (SDN) architectures for ad hoc networks impose one or more of the following constraints: infrastructure support, out-of-band single-hop (direct) links, location tracking, or preexisting IP connectivity. These limitations make the existing architectures inadequate for managing infrastructure-less Mobile Ad Hoc Networks (MANET) having limited bandwidth, unreliable connectivity, and susceptibility to high interference, packet collisions, and losses.**

In our dissertation, we design an architecture for Software-Defined Mobile Ad Hoc Networks (SD-MANET) that does not have any of the above constraints. We present three centralized communication protocols catering to MANET's need for proactive, reactive, and hierarchical routing strategies. We also develop two decentralized zero-control-packet routing protocols for MANETs characterized by ultra-low data rates. We present theoretical analyses of our protocols and perform simulation experiments addressing a wide range of scenarios: network scalability, density, traffic load, node mobility, and data rate. The results indicate that our solutions provide better performance than state-of-the-art approaches in most scenarios and address the inherent MANET challenges.

I. Introduction

Software-Defined Networking (SDN) has been explored extensively in the past decade for a wide range of networks. The design principles of SDN are generic and extend to both wired and wireless networks. The SDN architecture offers centralized network management, data plane programmability, network virtualization, and dynamic enforcements of policies, access control, QoS, and load balancing [1].

However, extending SDN applications to infrastructure-less networks, particularly to Mobile Ad Hoc Networks (MANETs), faces several unique challenges, mainly due to the dynamic nature of MANETs. In a typical MANET, nodes have a limited wireless transmission range and need multi-hop communication. There is no separate channel for control communication. Node mobility results in a dynamic and unstructured network topology. Low-capacity links and unsynchronized transmissions cause high interference and packet collisions. Propagation loss and link instability result in unreliable connectivity, especially over multi-hop links.

Contrary to popular belief, most deployed SDN architectures split the control plane between the SDNC and the local agents running on switches (or on servers hosting the software switches) for a practical and scalable design. Typically, the control communication between the SDNC and the local agent is governed by a custom-designed protocol. The local agent uses a protocol such as OpenFlow for updating flows in the switch, e.g., VMware's NSX [2] uses a Local Control Plane agent for updating flows in Open vSwitch using OpenFlow.

978-3-903176-32-4 © 2021 IFIP

Protocols such as OpenFlow do not have the necessary features for managing MANETs, nor do their large-sized messages suit the low-capacity links. However, most existing SDN architectures [3]–[13] proposed for MANETs [3]–[6] and other similar networks (e.g., sensor [7], [8], mesh [9], [10], vehicular [11], aerial [12], multi-hop Internet-of-Things [13], [14]) propose using one or more of the following: (1) OpenFlow or ForCES control communication over a separate channel, (2) single hop (direct) communication links between SDNC and other mobile nodes, (3) LTE base station for hosting stationary SDNC, (4) location services for learning network topology, (5) preexisting IP connectivity for control communication. These constraints render existing architectures inadequate for infrastructure-less networks having low capacity, unreliable connectivity, and susceptibility to high interference, packet collisions, and losses. Further, these architectures lack autonomous topology discovery and fail to address the network dynamics [1], [10]. They focus on improving certain use cases (e.g., traffic engineering) but fail to evaluate their results against state-of-the-art decentralized ad hoc solutions.

In our dissertation [15], we present an architecture for Software-Defined Mobile Ad Hoc Networks (SD-MANET) without imposing any of the above constraints and design several novel centralized and decentralized protocols for managing the network and establishing connectivity. Below, we enumerate all the contributions of our dissertation.

1. SD-MANET Architecture: Our SD-MANET architecture [16], [17] caters to the need for infrastructure-less networks, in which SDNC is a mobile node within the same network and uses in-band custom-designed protocol to facilitate both control communication and IP connectivity. All nodes, including the SDNC, have limited transmission ranges and need multi-hop control and data communication. The SDNC manages the network by learning network topology – without using any location services – and sending routing information.

We identify three functions that are fundamental for managing a MANET in a centralized manner. We show that all SDN-based architectures for ad hoc networks need these functions to realize the envisioned benefits of SDN. We first discuss their naive implementation and then describe how our optimizations mitigate interference and improve reliability, and thereby, improve overall network performance. Since MANETs are

susceptible to frequent link failures, we implement schemes that enhance the reliability of both control and data communication. These are novel schemes found in no other protocols.

2. Zero-Control-Packet Routing Protocols: MANET applications in narrowband tactical networks, off-grid disaster relief, and long-range outdoor (Industrial) Internet-of-Things (IoT), often need low-cost, lightweight devices that consume low-power and have long ranges. Realizing these requirements requires using ultra-low data rates. Such data rates make the network capacity so low that the overhead of the control packets overwhelms the network by consuming most of the available bandwidth. Further, such low data rate networks experience high interference and packet collisions due to longer transmission delays. To address these needs, we propose an architecture [18] and design two novel zero-control-packet routing protocols: ECHO [19] and VINE [20]. These protocols do not use any routing control packets, whatsoever. ECHO performs efficient network-wide broadcasts, often needed for military operations, emergency beaconing, and collaborating mapping. VINE reliably delivers messages for 1-1 communication over multiple hops. These protocols are already productized and implemented in the goTenna mesh devices [21] and used successfully for communicating during forest fires, hurricanes, and military operations [22].

3. SD-MANET Routing Protocols: Catering to MANET's needs for reactive and proactive routing strategies, we design three centralized protocols: CORR, CPR, and HCPR [15], [23]. These protocols share a few features with ECHO and VINE. They are generic and suitable for adoption by other centralized architectures. CORR is reactive, in which the SDNC sends routing information upon receiving route requests. CPR is proactive, in which the SDNC periodically sends routing information for maintaining up-to-date routes in nodes. HCPR addresses the inherent scalability issues in SD-MANET by building node clusters and configuring intra-cluster and inter-cluster routing.

4. Evaluations: We evaluate our protocols by analyzing their communication complexities theoretically and conducting detailed performance studies for a wide range of scenarios, addressing network scalability, density, traffic load, node mobility, and data rate. We compare their results to standard and widely used MANET solutions.

II. SD-MANET Architecture

In our SD-MANET architecture (shown in Figure 1), SDNC is a mobile node within the network and manages other nodes in a centralized manner. Similar to the architectures in the literature [1], our architecture also has the firewall, policies, QoS, and load balancer applications, running inside SDNC for managing the network and realizing the SDN benefits. However, what makes our architecture unique is the three managers inside the SDNC and the functions they perform.

A. Network Functions

We identify three functions that are fundamental for managing a MANET in a centralized manner. They are (1) learning route to SDNC, (2) learning network topology, and (3) sending routing information. The Connectivity Manager helps other nodes learn their dynamic route to SDNC and use them for control communication. The Topology Manager helps SDNC maintain a global view of the dynamic network topology for selecting the network routes. The Forwarding Manager helps SDNC disseminate routing information efficiently.

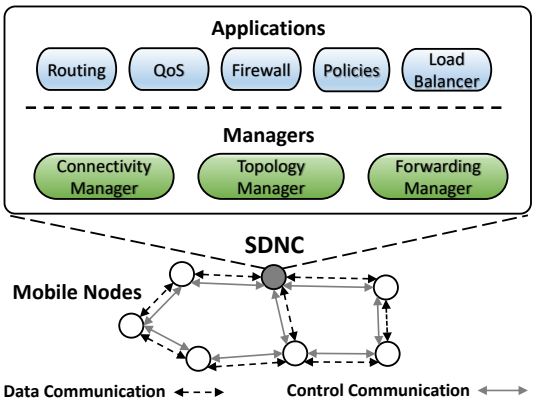


Fig. 1. SD-MANET Architecture

Despite being necessary and fundamental, most architectures fail to address these functional needs and assume that reliable and efficient communication between SDNC and other nodes is always available. However, it is crucial to implement these functions optimally to realize any of the envisioned SDN benefits. In large networks, their naive implementation would increase network congestion and interference, leading to several packet losses. So, learning the network topology by frequently collecting the neighbor information of *all* nodes and sending route updates to *each and every* node increase the congestion and control overhead significantly. Further, unreliable control communication results in SDNC learning partial or disconnected network topology and failing to select appropriate routes, causing catastrophic effects on the network. Therefore, we focus our research on the implementation of these functions. We explain our solutions with the help of our centralized SD-MANET routing protocols in Section IV.

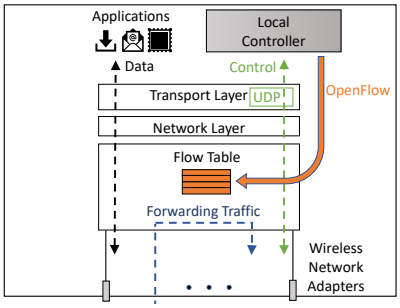


Fig. 2. Internal structure of an SD-MANET Node

B. Control Communication and Network Programmability

Figure 2 shows the internal structure of a node with a Local Controller (LC) for updating flows using OpenFlow. LC communicates with the SDNC using our custom-designed protocol that facilitates both control communication and IP connectivity between the nodes. We have carefully designed

the protocol and each of its messages using the RFC 5444 specifications. A detailed description of these messages is available in Appendix in [15]. These messages carry information to facilitate MANET functions like neighbor discovery and topology discovery, and control communication with SDNC. The embedded routing information allows the LC to create FlowMod (OpenFlow) messages with match fields and actions from the OpenFlow specification. SDNC realized network programmability by updating flows dynamically based on the network topology, routing strategies, and policies.

III. Zero-Control-Packet Routing Protocols

Narrowband (e.g., NATO NBWF), off-grid disaster relief, and other such communication contexts are characterized by ultra-low data rates. For example, NBWF uses 20-82 Kbps [24], the long-range IoT standard (LoRa) uses 0.3-50 Kbps [25], and Zigbee [26] (IEEE 802.15.4-based specification for low-power radios) uses 250 Kbps. These technologies are envisioned to operate in a mobile multi-hop manner [14].

Public safety professionals in disaster relief situations often need an instantly and inexpensively deployable off-grid communications system for collaborative mapping, texting, and emergency beaconing [27]. They need communication devices to be lightweight, low power, and low cost. Besides, they need multi-hop off-grid connectivity for covering a large area with support for user mobility. These requirements necessitate using devices with low data rates.

Using low data rates is not a problem in itself but essentially a routing protocol problem. Routing protocols use different strategies for generating and disseminating control packets, and some may do the job with fewer control packets. However, it turns out that the very act of using control packets consumes a base level of overhead. In networks with high data rates, the overhead of these control packets is tolerable. However, when the data rate is low, it consumes most of the available bandwidth, leaving little or none for the actual traffic. For addressing the above issues, we design two zero-control-packet routing protocols: ECHO and VINE.

A. ECHO

Several MANET applications require Network-Wide Broadcast (NWB), that is, sending a packet from a given source to all nodes in the network. Examples include position updates for collaborative mapping, group chats, clock synchronization messages, and routing control messages. A simple solution to the NWB problem is *Flooding*, but it results in excessive transmissions and collisions. An often-used approach is to determine a (minimal) set of nodes that should re-transmit such that all nodes receive the message. From a graph-theoretic viewpoint, this problem can be formulated as the *Minimum Connected Dominating Set* problem, which is NP-complete.

Most existing NWB solutions are either probabilistic (i.e., do not guarantee delivery even in lossless conditions), assume location information, or utilize control packets to collect local or global topology information. Thus, their application is limited and unreliable in ultra-low capacity networks.

We present the first deterministic, zero-control-packet, location-unaware protocol for efficient network-wide broadcasting in MANETs. Called ECHO, our protocol uses node identifier information (the *prevSender* field) within the data packet header to determine – in a fully distributed and source-independent manner – the set of *critical* (also referred to in the literature as *dominating*, *relay* or *rebroadcast*) nodes whose transmission is sufficient for a NWB.

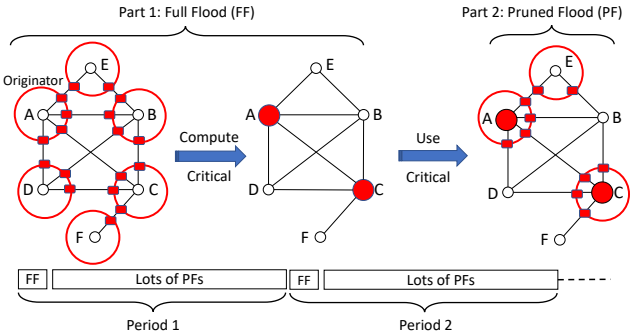


Fig. 3. Full Flood (FF): All nodes rebroadcast and select critical nodes (nodes A and C). Pruned Flood (PF): Only critical nodes rebroadcast.

Figure 3 illustrates ECHO's overview. ECHO consists of two interwoven phases: Full Flood (FF) and Pruned Flood (PF). The FF phase executes periodically, flooding a randomly chosen data packet for selecting critical nodes. Specifically, a node marks itself critical if and only if it receives an "echo" of its transmission, that is, a node received a packet with itself marked as the previous sender. The PF follows the FF phase, wherein only the critical nodes rebroadcast. An FF data packet originated at a single node builds critical nodes that are valid for broadcasting packets originated from any node. That is, the selected critical nodes are *source independent*. Nodes transmit an overwhelming majority of packets via PF until the next FF, resulting in highly efficient network-wide broadcasts. Unlike prior deterministic protocols, ECHO does not use any control packets or explicit topology information.

We prove the correctness of ECHO and show that the set of critical nodes selected by ECHO is sufficient for a source-independent network-wide broadcast.

We analyze its communication complexity and show it to be $O(N)$ lower (better) in dense networks than both Flooding and Multi-Point Relay (MPR). We also show using simulations that ECHO outperforms both MPR and Flooding in the packet delivery percentage and communication load (see Table I). ECHO is simple to implement, robust, and scalable, making it a valuable protocol for real-world multi-hop wireless networks. While it is suitable for all multi-hop wireless networks, it is especially crucial for ultra-low bandwidth applications.

B. VINE

VINE is a zero-control-packet protocol for 1-1 messaging over multiple hops. VINE builds routing states by inspecting data packets headers that it then uses for forwarding subsequent data packets. Specifically, VINE uses the *sender*, *prevSender*, and *hop count* fields for building states to *1-hop* nodes, *2-hop* nodes, and the *origin* of the packet, respectively.

Over time as traffic flows, an increasingly rich sink tree toward each node is created, resembling the growth of a *vine* in a *grove*. Nodes use their gradient states for forwarding data packets along non-increasing cost gradients (like water flowing downhill). If there is no fresh-enough gradient, then the node broadcasts. This decision is taken independently at each hop – thus, a packet may alternate between broadcast (if no state exists) and unicast (if state exists) en route to its destination.

VINE improves reliability using Implicit Acknowledgments (IA) and End-to-End Acknowledgments (E2E-A) – features not present in most routing protocols. VINE provides per-hop reliability via IA (i.e., retransmissions based on overheard forwarded packets), and delivery notification via E2E-A. VINE treats E2E-A similar to data packets and uses them for building and updating gradient states.

We derive an expression for VINE's communication complexity and show that it tends to stabilize quickly. We analyze the traffic churn needed to maintain the states and show that the "sweet spot" is very low. AODV is the basis of many low-rate networking solutions, including RPL [28], LOADng [29], and IEEE 802.11s [30], so we compare its results to VINE for many scenarios. VINE outperforms AODV in all of them, ensuring versatility, reliability, and resilience against dynamic topology changes, and proves to be a better protocol in general.

Table I summarizes the ECHO and VINE results for the increasing network size scenario (scalability). Results for the density, load, data rate, mobility scenarios are similar but not included here due to lack of space. They are available in [15].

TABLE I
ZERO-CONTROL-PACKET PROTOCOLS RESULTS

Protocol	Pkt. Delivery %	Comm. Load	Compared To
ECHO	1.4x higher	4x less	Flooding & MPR
VINE	2.5x higher	1.2x less	AODV

IV. SD-MANET ROUTING PROTOCOLS

In Section II, we described our SD-MANET architecture and three functions necessary for managing a MANET in a centralized manner. We also discussed how their naive implementation would result in a high control overhead, congestion, and overall poor performance. Now, we explain how the SDNC establishes control communication with other nodes using different routing strategies, and how it optimizes the three functions using the features of ECHO and VINE.

We reduce the control overhead by employing ECHO in the Topology Discovery (TD) procedure used by SDNC. The TD procedure floods a message that results in the following: (1) all nodes learning or updating their route to the SDNC, (2) identification of nodes that form a Connected Dominating Set (CDS) of the network graph, i.e., *critical nodes*, and (3) all nodes knowing their neighbor nodes. The SDNC learns the network topology using the neighbor information of *only* critical nodes. Further, the SDNC selects routes *only* for critical nodes and disseminates the routing information as network-wide broadcasts via critical nodes. Both these optimizations reduce the overhead, significantly, and make the identified critical nodes act as the *network backbone* for forwarding both control and data packets. The CDS property of critical nodes ensures the following: (1) the learned network topology is connected, and (2) each node has a connected path to every other node through the critical nodes. Figure 4 shows an SD-MANET with one possible set of critical nodes. We note that SDNC is also a critical node, and it addresses the network dynamics by periodically calling the TD procedure, which updates the set of critical nodes.

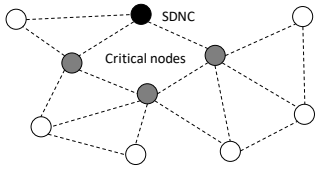


Fig. 4. Example SD-MANET and one possible set of critical nodes.

We address the issue of unreliable connectivity by using VINE's Implicit Acknowledgment (IA) feature for forwarding control packets to the SDNC. In particular, critical nodes use IA for sending their neighbor information (i.e., NI messages) to the SDNC and improve the per-hop reliability as well as the likelihood of SDNC receiving all NI messages, and hence, learning a connected network topology.

We also propose a novel way of forwarding data packets that suits the dynamic nature of MANETs. Most traditional protocols use schemes for identifying link breaks and then update the affected routes. During this process, nodes either drop or buffer data packets, resulting in either low delivery ratio or high delay. By contrast, when a link breaks, our centralized routing protocols broadcast the data packets instead of dropping or buffering them. This functionality makes forwarding a combination of unicast and broadcast. However, only critical nodes forward and leverage their CDS property for delivering packets to the destinations. After that, the SDNC opportunistically updates routes and suppresses the need for broadcasting subsequent packets. These features reduce the delay and improve the packet delivery ratio.

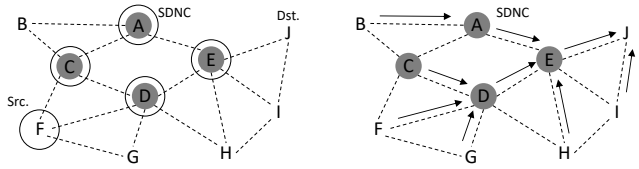
Using the above optimizations, we design three routing strategies: CORR, CPR, and HCPR. We evaluate them in an enhanced ns3 simulator (results in Table II) and derive their asymptotic communication complexities (shown in Table III).

A. Centralized Opportunistic Reactive Routing (CORR)

CORR is a reactive protocol, in which the SDNC learns the network topology *proactively* but sends the routing information *reactively*. The SDNC initiates the TD procedure periodically, allowing nodes to identify their state (i.e., critical or non-critical), recognize their neighbors, and learn their route to the SDNC (RTS). *Only* the identified critical nodes send their neighbor information in the NI messages, allowing the SDNC to maintain up-to-date network topology.

Figure 5 shows an overview of CORR. Nodes broadcast data packets in the absence of valid routes. Only critical nodes forward and leverage their CDS property for delivering packets to the destination. They also send route request messages to SDNC for requesting routes to the destination. The SDNC knowing the network topology (from the TD procedure) selects

routes for *all* critical nodes, regardless of the message originator, and disseminates the routing information via network-wide broadcasts. All nodes *opportunistically* update their routing information. Thus, a single request message results in updating routes in all nodes (unlike AODV) and suppresses the need for sending multiple route requests for a particular destination. Nodes can forward all subsequent data packets to that destination as unicast. As traffic flows and the SDNC receives requests for different destinations, it rapidly updates routing information in *all* critical nodes, creating a strong network backbone for data packet forwarding.



(a) Broadcasting via critical nodes (b) Unicast forwarding

Fig. 5. CORR overview: (a) Node F broadcasts packet in the absence of route to node J and critical nodes forward; (b) Nodes unicast packets with destination J to their next hop once SDNC opportunistically updates routes.

CORR being a reactive protocol, we compare its results to AODV (shown in Table II). Similar to Table I, we show results for the increasing network size scenario (scalability) and other results are available in [15]. Not only CORR attains lower overhead and delay but also delivers more packets than AODV. The low overhead is due to an opportunistic way of updating routes. The low delay is because, unlike AODV, nodes do not buffer packets while waiting for routes.

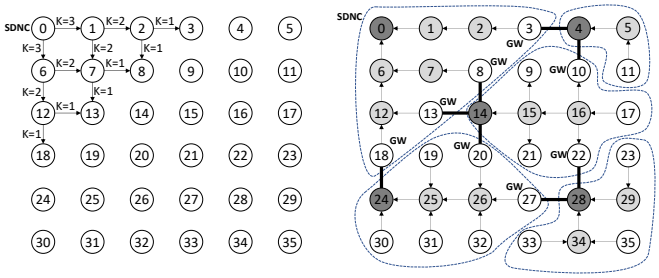
B. Centralized Proactive Routing (CPR)

CPR is a proactive routing protocol, in which SDNC learns the network topology and sends the routing information both *proactively*. Similar to CORR, CPR allows the SDNC to learn the network topology using neighbor information of *only* critical nodes and to disseminate routing information as network-wide broadcasts via critical nodes. We show that these optimizations reduce the communication complexity by $O(N)$ in dense networks, and for the generic case, the relative gain increases with the decreasing number of critical nodes (N_c).

The simulation results indicate that CPR provides the same or better delivery ratio than OLSR but causes significantly lower overhead and delay for networks of size up to 100 nodes (see Table II). However, on repeating the simulations for large networks, we observed scalability issues. In a network of 250 nodes, CPR selects 37 critical nodes on average. The NI messages of these critical nodes cause congestion at SDNC. Also, SDNC sends routing information in several network-wide broadcast messages, which further increases congestion and overhead. These issues motivated us to design a hierarchical approach for routing packets in large networks.

C. Hierarchical Centralized Proactive Routing (HCPR)

HCPR is a hierarchical routing protocol designed for addressing the scalability issues in CPR. It builds clusters of nodes and identifies gateway nodes in the network. Each cluster has its Cluster Head (CH). Nodes receive intra-cluster routing information from their CHs, while the gateway nodes facilitate inter-cluster routing. HCPR is proactive and performs all its functions periodically to account for network dynamics.



(a) SDNC initiating TD procedure (b) Clusters and gateway nodes

Fig. 6. HCPR protocol overview: (a) SDNC uses a TTL-like field (*K*) in the Topology Discovery (TD) procedure for defining the cluster radius; (b) One possible cluster formation (*K*=3) in a 6x6 Manhattan Grid. CHs shown with darker shade and critical nodes with lighter.

Figure 6 shows an overview of HCPR. The SDNC uses a *time-to-live* field, called *K*, in the Topology Discovery (TD) procedure for identifying clusters in the network. Nodes at a *K-hop* distance from the SDNC become candidates for CHs. Each candidate waits for a random interval before initiating its TD procedure. This random delay prevents all candidates from becoming CHs. Each cluster has its own set of critical nodes. When a node that is already in a cluster receives a message from a CH, it elects itself to be a gateway node.

Each CH learns the cluster topology using the neighbor information of critical nodes in its cluster and then selects the intra-cluster routing information for sending via critical nodes as cluster-wide broadcasts. CHs also propagate their cluster topologies to the SDNC via the gateway nodes, allowing all intermediate nodes to learn routes to the nodes in the source cluster and facilitate inter-cluster routing.

The simulation results of HCPR show significant improvement over the results of CPR. Table II summarizes the results of CORR, CPR, and HCPR for the increasing network size scenario. Other scenarios also have similar results [15].

TABLE II
SD-MANET ROUTING PROTOCOL RESULTS

Protocol	Pkt. Delivery %	Comm. Load	Delay	Compared To
CORR	10% better	1.5 less	3x less	AODV
CPR	similar	2.4x less	1.4x less	OLSR (size 100)
HCPR	similar	2.7x less	2.4x less	OLSR (size 250)

Table III presents the asymptotic control communication complexities of the CPR and HCPR protocols for generic, dense, and sparse networks. PCC is a protocol that does not use critical nodes. Here, d is the average node degree, N is the network size, D is the network diameter, N_c is the number of critical nodes, C_{ch} is the number of clusters, and C_c is the average number of critical nodes in a cluster. Both CPR and HCPR have *O(N)* lower complexity than PCC in dense networks. For a generic network with average node degree

d, CPR's gain over PCC depends on the critical nodes (N_c), and HCPR's gain over CPR depends on K. A more detailed theoretical explanation of these complexities (and CORR's communication complexity) is available in [15].

TABLE III
CONTROL COMMUNICATION COMPLEXITIES

Protocol	**Generic**	**Dense** $d = O(N)$	**Sparse** $d = O(1)$
PCC*	$O(dND + N^2D)$	$O(N^2)$	$O(N^3)$
CPR	$O(dN_cD + NN_c^2)$	$O(N)$	$O(N^3)$
HCPR	$O(dN_cK + N(C_c^2 + C_{ch}K))$	$O(N)$	$O(N^3)$

* Protocol using a naive implementation of the three functions

V. CONCLUSIONS AND IMPACT

In this dissertation, we have presented a novel SDN architecture suitable for infrastructure-less MANETs. Our architecture helps us realize the benefits of SDN without limiting its applicability to MANETs. Our architecture can easily extend to serve the needs of sensor, mesh, vehicular, and IoT networks.

We have identified three functions that are fundamental for managing a MANET in a centralized manner and proposed solutions to perform them efficiently. Our implementation of these functions mitigates interference, congestion, and overhead, thereby, improves reliability and network performance.

We have designed centralized protocols that establish control communication and cater to the needs of reactive and proactive routing strategies of MANETs. We have also designed a hierarchical routing protocol to address the inherent scalability issue in SD-MANET. We have presented theoretical analyses of the communication complexities of all our routing protocols and conducted detailed performance studies to evaluate them on scenarios addressing network scalability, density, traffic load, and node mobility. The evaluation results indicate that not only our centralized SD-MANET routing protocols are competitive in performance but also better, in most scenarios, than state-of-the-art decentralized MANET protocols. Our SD-MANET routing protocols break the dogma that centralized approaches are inappropriate and unscalable for MANETs.

Our architecture moves the complex topology discovery and route selection procedures into the SDNC and saves nodes' resources (and improves battery life). It prevents the periodic exchange of routing information between all neighbor nodes. It also enables the opportunity of adjusting routing parameters based on network dynamics and leveraging existing topology control approaches for managing network connectivity.

We have also designed two zero-control-packet routing protocols for addressing the challenges of ultra-low capacity MANETs. With the design of these protocols, we have presented a radical departure from the prevalent thinking that routing requires collecting topology information via control packets. We have proved their significance not only by showcasing their better communication complexity and performance results but also by their deployment in the goTenna mesh devices [21]. These devices are being successfully used for communicating during fighting forest fires, hurricanes, and military operations [22].

REFERENCES

[1] I. T. Haque and N. Abu-Ghazaleh, "Wireless Software Defined Networking: A Survey and Taxonomy," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2713–2737, Fourthquarter 2016.

[2] NSX-T Data Center Reference Design Guide. https://docs.vmware.com/en/VMware-NSX-T-Data-Center/index.html. Accessed: Nov-2020.

[3] K. Poularakis, G. Iosifidis, and L. Tassiulas, "SDN-Enabled Tactical Ad Hoc Networks: Extending Programmable Control to the Edge," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 132–138, 2018.

[4] K. Poularakis *et al.*, "Hybrid SDN Control in Mobile Ad Hoc Networks," in *Proc. of SMARTCOMP*, Washington, DC, USA, 2019, pp. 110–114.

[5] H. C. Yu, G. Quer, and R. R. Rao, "Wireless SDN mobile ad hoc network: From theory to practice," in *Proc. of ICC*, Paris, May 2017.

[6] P. Bellavista, A. Dolci, and C. Giannelli, "MANET-oriented SDN: Motivations, Challenges, and a Solution Prototype," in *Proc. of WoWMoM*, Chania, Greece, Jun. 2018, pp. 14–22.

[7] A. De Gante, M. Aslan, and A. Matrawy, "Smart wireless sensor network management based on software-defined networking," in *Proc. of QBSC*, Kingston, ON, Canada, Jun. 2014, pp. 71–75.

[8] M. Aslam, X. Hu, and F. Wang, "SACFIR: SDN-Based Application-Aware Centralized Adaptive Flow Iterative Reconfiguring Routing Protocol for WSNs," *Sensors*, vol. 17, no. 12, p. 2893, Dec. 2017.

[9] A. Detti, C. Pisa, S. Salsano, and N. Blefari-Melazzi, "Wireless Mesh Software Defined Networks (wmSDN)," in *Proc. of WiMob*, Lyon, France, Oct. 2013, pp. 89–95.

[10] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for Wireless Mesh Networks," in *Proc. of ICCCN*, Maui, USA, Jul. 2011, pp. 1–6.

[11] S. Correia, A. Boukerche, and R. I. Meneguette, "An architecture for hierarchical software-defined vehicular networks," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 80–86, 2017.

[12] L. Gonzalez *et al.*, "Transport-Layer Limitations for NFV Orchestration in Resource-Constrained Aerial Networks," *Sensors*, vol. 19, 2019.

[13] S. Bera *et al.*, "Software-Defined Networking for Internet of Things: A Survey," *Internet of Things Journal*, vol. 4, no. 6, pp. 1994–2008, 2017.

[14] H. Kharrufa *et al.*, "Dynamic RPL for multi-hop routing in IoT applications," in *Proc. of WONS*, Jackson, WY, USA, 2017, pp. 100–103.

[15] A. Dusia. (2019) Software-Defined Architecture and Routing Solutions for Mobile Ad Hoc Networks. Ph.D. Dissertation. [Online]. Available: https://udspace.udel.edu/handle/19716/25641

[16] A. Dusia, V. K. Mishra, and A. S. Sethi, "Control Communication in SDN-based Dynamic Multi-hop Wireless Infrastructure-less Networks," in *Proc. of IEEE ANTS 2018*, Indore, India, Dec. 2018.

[17] V. K. Mishra, A. Dusia, and A. S. Sethi, "Routing in Software-Defined Mobile Ad hoc Networks (SD-MANET)," US Army Research Laboratory, Tech. Rep. ARL-TR-8469, Aug. 2018.

[18] R. Ramanathan, C. Servaes, W. Ramanathan, A. Dusia, and A. S. Sethi, "Long-Range Short-Burst Mobile Mesh Networking: Architecture and Evaluation," in *Proc. of IEEE SECON 2019*, Jun. 2019.

[19] A. Dusia, R. Ramanathan, W. Ramanathan, C. Servaes, and A. S. Sethi, "ECHO: Efficient Zero-Control-Packet Broadcasting for Mobile Ad Hoc Networks," *IEEE Trans. Mobile Computing*, 2021.

[20] ——, "VINE: Zero-Control-Packet Routing for Ultra-Low-Capacity Mobile Ad Hoc Networks," in *Proc. of IEEE MILCOM*, Nov. 2019.

[21] goTenna Pro. https://gotennapro.com/. Accessed: Nov-2020.

[22] "goTenna Deployment After Action Reports," https://gotennapro.com/pages/resources#case-studies, Accessed: Nov-2020.

[23] A. Dusia, R. Ramanathan, and A. S. Sethi, "CORR: Centralized Opportunistic Reactive Routing for Mobile Multi-hop Wireless Networks," in *Proc. of IEEE ICCCN 2019*, Valencia, Spain, Jul. 2019.

[24] "NATO Standardization Agreement (STANAG) 5631/AComP-5631, Narrowband Waveform Physical Layer, Draft, Edition 1," 2015.

[25] LoRaWAN. https://lora-alliance.org/. Accessed: Nov-2020.

[26] Zigbee. https://www.zigbee.org/. Accessed: Nov-2020.

[27] R. Ramanathan. (2019, Jun.) Long-Range Short-Burst Mobile Mesh Networking. https://inthemesh.com/archive/long-range-short-burst-mobile-mesh-networking.

[28] T. Winter *et al.*, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," https://tools.ietf.org/html/rfc6550, Mar. 2012, RFC 6550.

[29] M. Vučinić, B. Tourancheau, and A. Duda, "Performance comparison of the RPL and LOADng routing protocols in a Home Automation scenario," in *Proc. of WCNC*, Shanghai, China, Apr. 2013.

[30] G. R. Hiertz *et al.*, "IEEE 802.11s: The WLAN mesh standard," *IEEE Wireless Communications*, vol. 17, no. 1, pp. 104–111, 2010.