Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 205 (2022) 188–197

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Military Communications and Information Systems (ICMCIS 2022)

Freezing Unicast Paths in Sensor Swarms

Erlend Larsen, Håkon Vågsether*, Lars Landmark*

[a]Norwegian Defence Research Establishment, Kjeller, Norway

Abstract

Drone swarms have great potential to convey sensor information. However, a swarm may have a very high degree of topology dynamics. This impedes the sensor data distribution from the sensor node to consumers which are external to the swarm. A high sensor data rate requires reliable and efficient forwarding between the sensor and the consumer. This often makes unicast the preferred data forwarding method. But routing protocols struggle to achieve stable paths in high-mobile topologies. This makes unicast forwarding without suffering substantial packet loss very difficult. In this paper, we investigate how the swarm nodes can help to provide a stable path for the duration of a sensor data transfer. Stability is achieved by freezing the positions of the swarm nodes for the duration of the unicast flow. We investigate three different mechanisms to trigger this freeze, where two are based on the Ad-hoc On-demand Distance Vector (AODV) routing protocol. The results are very promising, with throughput for one flow remaining stable at almost 100% with 20 m/s swarm mobility, compared to the baseline results of 82% throughput. The results also indicate that even swarms without reactive routing may benefit from the freeze method to provide stable unicast paths as required.

Keywords: Swarm; MANET; Sensors; Simulations; Unicast

1. Introduction

Sensor swarms can be very effective information collectors in military operations. A swarm consists of a high number of coordinated units that can provide an omnidirectional coverage over a target area. Distributed control makes the swarm resilient to losses. The units can carry different sets of sensors or even weapons, allowing for flexible and fine-grained data collection, engagement, and battle damage assessment.

A drone swarm may consist of a considerable number of Unmanned Aerial Vehicles (UAVs), coordinating through wireless communications. The swarm network is responsible for conveying information between participating swarm nodes and to external end-users. Information to external users is often associated with multi-hop networking and are

* Corresponding author. Tel.: +47 63807000.
E-mail address: hakon.vagsether@ffi.no ; lars.landmark@ffi.no

10.1016/j.procs.2022.09.020

thus challenged with stale links due to mobility. Such information includes sensor information, which is bandwidth-intensive and has interest primarily for an end-user beyond the swarm, e.g. at a ground station.

Distributed ad hoc networks, and in particular Flying Ad Hoc Networks (FANETs), are characterized by a high degree of topology changes. Unicast routing in such networks struggles with numerous challenges, spanning from interference and hidden nodes, to mobility causing both fluctuating links and link breaks. Mobility, and thus link breaks, challenges routing and further traffic forwarding. The ability to maintain valid and stable routes is the utmost critical factor to ensure uninterrupted data delivery between nodes. The ability to provide stable paths decreases with the number of hops between source and destination along with the mobility pattern.

The topology mobility, the greatest challenge of swarm communications, can also be turned to an advantage. The swarm nodes that take part in the forwarding of unicast traffic can stay still (freeze) to ensure a stable path for the duration of the unicast transfer. Freezing a subset of the swarm nodes will have a functionality cost for the swarm, leading to less freedom of mobility than if none of the nodes froze.

The solutions in this paper are best suited for rotary-wing aircraft. While a fixed-wing aircraft could achieve freezing in place by circling a position, aspects such as antenna propagation patterns will pose additional challenges which are not addressed in this paper.

In this paper, we propose solutions to freeze the nodes that make up the path of a unicast traffic flow in a swarm for the duration of the flow. Two of three investigated solutions are based on a reactive routing protocol. The reactive routing protocol we employ is the Ad-hoc On-demand Distance Vector (AODV) [4]. The rest of the paper is structured as follows: In Section 2, we discuss related work. We present the most important parts of AODV for our purposes in Section 3. In Section 4, we elaborate on the challenges facing unicast traffic in a swarm. In Section 5, we present our solutions to the unicast mobility problem. The solutions are put to the test in a simulation study presented in Section 6. In Section 7, we conclude the paper with some proposed next steps.

2. Related work

Rahman et al. in [10] propose an algorithm to maximize the throughput by positioning the UAVs in a software-defined disaster UAV communication network. The algorithm used the information about network topology as well as data rate demands and paths of flows, and obtained an average throughput enhancement of 26% by optimally positioning the UAVs.

Landmark et al. in [6] show how unicast and broadcast/multicast differ in terms of robustness, and propose a combination of the two communication types to improve robustness in a ground-based MANET supported by a UAV.

In [9], the authors propose simple practical behaviors for a UAV supporting a ground-based MANET, to position the UAV optimally. A connectivity-oriented flight planning algorithm for UAV swarm nodes is proposed in [11]. Finally, in [12], the trade-off between spatial coverage and connectivity for a UAV network is investigated.

3. The AODV routing protocol

Two of the three solutions in our work rely on standard signaling mechanisms for reactive routing. Reactive routing connects any two interacting nodes on demand. Hence, the routing protocol neither creates nor maintains unused routes. We base the implementation on AODV [4] as an example of a generic reactive routing protocol.

The basic functionality of AODV is as follows: When a route to a new destination is needed, the node broadcasts a Route REQuest (RREQ) to find a route to the destination. A route is found when the RREQ reaches either the destination itself, or an intermediate node with a valid route to the destination. If the node receiving the RREQ is neither the destination nor has a valid route, the RREQ is forwarded by broadcast once by the receiving node. The RREQ is forwarded until exceeding the configured hop limit, the Time-to-Live (TTL).

A route is not operational before a Route REPly (RREP) is sent back to the originator of the RREQ. Each node that receives the RREQ caches a route back to the originator of the request, so that the RREP can be unicast by the destination along a path to that originator, or likewise from any intermediate node that is able to satisfy the request.

AODV can differentiate between who is allowed to respond to RREQ, and further reply with RREP. This is configured by the ”Destination Only” (D)-flag. Intermediate nodes are allowed to respond to a RREQ only if the D-flag is not set. If the D-flag is set, only the destination itself is allowed to reply with a RREP. The ”Gratuitous Reply”

(G)-flag is a flag that controls how AODV acts when an intermediate node replies to the originator of the RREQ with an RREP. With the G-flag set, the intermediate node will also send a new RREQ towards the destination for which it has replied with an RREP, to ensure that the route from the intermediate node to the destination is still valid. If the D-flag is set, the state of the G-flag is irrelevant, since there are no intermediate nodes replying to RREQs.

The RREQ potentially reaches all nodes in the network, while the RREP only reaches the nodes involved in data forwarding over a unicast path. To prevent unnecessary network-wide dissemination of RREQs, the originating node may use the Expanding Ring Search (ERS)-technique. With ERS, the originating node starts searching for a route using a small TTL in the RREQ packet along with a timeout for receiving a RREP. If the destination is not found within the TTL vicinity, the source originates a new search with a higher TTL. This continues until the TTL exceeds a configurable TTL_THRESHOLD. Beyond this threshold, the whole network is searched independent of the size.

The node can detect the link as broken either via Link Layer Notification (LLN), where the Medium Access Control (MAC) layer notifies the routing protocol of the broken link, or explicit HELLO packets sent both ways on the link between the source/forwarder node and its downstream peer. If a path is broken, AODV repairs the route by sending a Route ERRor (RERR) packet upstream to the source. This requires the ability of the node upstream to the link break to detect the link as broken. When the source receives the RERR packet, it restarts the route discovery process by broadcasting a new RREQ packet with an incremented sequence number.

4. Unicast for swarming UAVs

4.1. The reason for unicast in UAV swarms

Drone swarms consist of a number of units that self-organize using a simple set of rules. Most commonly small, the units have limited data processing capability. At the same time, several types of sensors can produce large volumes of data. One relevant military application is surveillance, where one or several units with sensors generate a live video feed with detected objects of interest.

The cooperating drones in a swarm need communication capabilities for two distinct uses. The first is for information exchange among the swarm nodes themselves, i.e., swarm coordination. One-hop broadcast packet exchanges fit very well with this purpose. The second is a method to convey sensor information from an information source to the swarm gateway.

Unicast is the preferred communication method whenever data is sent from the swarm nodes towards one or more consumers which may be external to the swarm itself. The main reason is the nature of the traffic flow as a stream of data towards a common gateway, which in combination with the potentially large numbers of network nodes in a swarm makes unicast more efficient than broadcast. Unicast also provides more reliability than broadcast, due to MAC layer retransmissions per link.

4.2. The mobility challenge

Unicast struggles with numerous challenges in UAV swarm networks, spanning from interference and hidden nodes, to fluctuating links and link breaks caused by mobility [2]. In this paper, we focus on mending mobility-induced challenges.

Unicast information exchange over multiple hops requires routing, which is more challenging in a mobile environment. The inherent nature of the swarm as a mobile topology creates the potential for link breaks during the information transfer. The Internet Protocol (IP) was not designed for end-to-end connections in highly mobile environments. While MAC layer retransmissions can increase the link robustness, this does not ensure end-to-end reliability. Packets may be lost after a number of retransmissions. Lost datagrams on intermediate links or nodes are not recovered in the network itself. Instead, recovery relies on functionality in the end-nodes, such as retransmissions by the Transmission Control Protocol (TCP). Thus, IP datagram forwarding favors stable paths, where fewer datagrams are lost.

Stable paths and the goal of a mobile and agile swarm are conflicting goals. A mobile swarm application is typically designed around its sensors, while networking is a supporting functionality. Swarm node mobility is tailored for use in a specific type of environment. This creates both possibilities and challenges for routing. Possibilities due to the ability to align routing and swarm node mobility, and challenges due to the high degree of mobility. Routing and forwarding should be tailored to the swarm application itself and, through this, the sensors and its operational environments. This is especially important in sparse swarms, where each network node has few neighbors.

In addition to routing, transmitting IP packets over a multihop network also involve interaction between the MAC layer and the network layer. As a unicast packet makes its way between the source and the destination, the packet forwarder needs to resolve the MAC address of the packet's next hop using the Address Resolution Protocol (ARP). The ARP keeps a cache of recently resolved tiers of IP and MAC addresses. If the ARP cache does not contain an ARP entry for the IP next hop, the sender must request this MAC address. This is done using an ARP request packet. The packet initiating the ARP request is put in the ARP request queue. The unicast packet forwarder broadcasts an ARP request searching for the packet's IP next hop. If the ARP request is successfully received by the next hop, this node sends an ARP reply back to the ARP requester. Upon receiving the ARP reply, the ARP requester commences to send the unicast packet onwards to the next hop. The ARP entries are regularly timed out and refreshed using new ARP requests. If the link is stale, there will be no reply to this ARP request. The sender will send multiple ARP requests, and at last discard the packet that initiated the ARP request. However, this will put a hold on all packets in the queue that lack an ARP entry, and new packets to the same destination will cause older packets to be discarded if the ARP request queue is full.

When nodes move, paths can be broken. Until the routing protocol detects that a path is broken, we consider the links that no longer work to be stale links. In the meantime, packets are routed to a next hop that isn't available and put in the output queue (the egress queue). Link breaks can be detected by notifications from the MAC layer, but this depends on the MAC layer implementation. More commonly, link breaks are detected by missing HELLO packets. A link timeout is typically set to three times the HELLO period. Consequently, packets are forwarded based on stale route entries in the period from an occurring link break until the stale link is purged in favor of a new valid link. In wireless homogeneous networks, the incoming interface is also the outgoing interface. The egress queue holds packets assigned with different next hop addresses. This is especially problematic when packets are transmitted using stale route information, since packets being forwarded based on stale route entries end up being discarded at the MAC layer after being attempted transmitted on the wireless channel a number of times. In the meantime, this behavior causes head-of-queue blocking.

4.3. Mending the mobility-induced challenges

The challenges generated by mobility can be addressed in several ways:

- The performance of the routing protocol can be improved by increasing the frequency, and thereby load, of signaling packets to detect and disseminate changes [3].
- The traffic can be made more resilient against mobility through mechanisms for Delay Tolerant Networking (DTN) [5].
- Relayed packets may be routed just-in-time before forwarded onwards through ingress queuing and routing on egress [8].
- Finally, as the swarm nodes are very mobile, the topology can be adjusted according to the communication needs [9]. Our solution is a variant of this approach.

Historically, the main focus has been on adapting to the current topology. However, with the introduction of drones, the focus has moved more towards constructing a topology that better facilitates routing and traffic forwarding than if left alone. The topology can be improved based on requirements given by the swarm application. The traffic type (e.g., unicast, multicast or broadcast) and the expected traffic pattern will define different requirements that can result in several variants of optimized topologies. Routing protocols contributing to physical topology changes can induce more complexity, due to unwanted or uncontrolled mobility from the routing protocol perspective.

In our paper, we propose to manipulate the topology through simple use of the signaling packets of AODV to freeze nodes that are needed as part of the forwarding path between a sensing swarm node and the consumer of the data.

5. Freezing the unicast traffic paths

In this work, we propose a simple, but effective, method to secure stable paths for unicast flows through freezing swarm nodes in position. Our goal is to provide stable paths, while limiting the impact on the swarm application and the involved sensors. The act of freezing nodes reduces the number of mobile sensors and hence also the swarm application capabilities. Hence, there will be a trade-off between the ability of the swarm to perform its application and sustaining the unicast flow.

AODV signaling can be exploited in two ways in order to freeze nodes. Either one can freeze on the propagation of RREQ through the swarm, or one can freeze on the returning RREP. In addition to the two solutions based on AODV signaling, node freeze can wait until the unicast data traffic has started propagating over the established path. This would avoid freezing on what may turn out to be failed AODV route discovery-processes. This last solution will also work independently of routing protocols. The three ways of freezing the forwarding path explore the trade-off between the consideration to the swarm application and the need for stable paths for information out of the swarm. Hence, the three proposed freeze methods are:

- Freeze upon receiving AODV RREQ – **rreq**
- Freeze upon receiving AODV RREP – **rrep**
- Freeze on the first forwarded datagram – **fwd**

The first method, *rreq*, uses the AODV RREQ packet as a freeze trigger. This method is the most invasive on the swarm application, freezing all nodes that receive the RREQ packet, regardless of whether they are needed to form the unicast path. We use the AODV ERS technique to prevent all nodes from freezing at every new route search. Nevertheless, the result of employing this method will be that a large number of nodes are frozen unnecessarily. It will be important to release them back to the control of the swarm application. Therefore we employ a soft timer. The timer is set upon receiving the first RREQ, and updated on consecutive RREQ and subsequently on the forwarded datagrams.

The second method, named *rrep*, uses the AODV RREP packet as a freeze trigger. The method aims to balance the likelihood of obtaining a stable operational path, while not drawing too many resources from the swarm application itself. The *rrep* leaves more nodes to the swarm application and its mobility pattern. The *rrep* solution can be seen as an optimization of *rreq* in terms of leaving more control to the swarm application when it comes to controlling a node's mobility pattern. Instead of freezing on received RREQ, it stays mobile and freezes only on a received RREP. Hence, only the nodes required for forwarding over the path will freeze. Compared to the previous method using RREQ, using RREP runs the risk of link break during the exchange of the RREQ and RREP. The method is more vulnerable to the combination of high mobility and the number of hops between source and destination. The benefit is the preservation of stable paths, while fewer resources are drawn from the swarm application. The frozen state of the nodes involved in the route is only maintained as long there are unicast datagrams forwarded over the route.

The third method, *fwd*, does not rely on AODV route signaling to trigger freeze. Instead it only freezes the node on datagram forwarding. A node is controlled by the swarm application until a new to-be-forwarded unicast datagram is received. That is, a node only freezes on unicast datagrams (RREP exempt). Similar to the freeze on RREQ and RREP, this method uses a soft-timer to release a frozen node back to the mobility requested by the swarm application.

As mentioned above, the maintenance and removal of the nodes' freezing condition is based on a soft timer. A soft timer is initiated upon either receiving RREQ, RREP or the first datagram using the new constructed route entry. The soft timer is further updated whenever a datagram is forwarded and associated with the route entry connected to the RREQ/RREP or fwd. If no datagram is being forwarded, the freeze expires along with the route information.

The methods' main goal is the same, that is to provide a stable End-to-End (E2E) path. Their main differences can be divided in two: 1) the difference in elapsed time from the source initiates the first RREQ to the data forwarding nodes have established a stable path, and 2) the number of nodes affected and the associated costs.

Table 1: Default simulation parameter settings.

Parameter	**Setting**
Communication frequency	2.4 GHz
Propagation model	Two-ray-ground
UAV antenna altitude	10 m
Control rate	1 Mbps
Data rate	1 Mbps
MAC protocol	IEEE 802.11b
MTU	1500 bytes
Data packet size	1250 bytes
Traffic start	0 s
Traffic start jitter	30 s
Measurement start	30 s
Measurement stop	330 s
Traffic flow duration	30 s
Pause between traffic flows	±0.5 s
Sampling times	1 s
Confidence interval	95%
Simulation time per run	360 s
Number of runs per data point	10

6. Simulations

6.1. Simulation setup

The simulations are run in the ns-3 [1] simulator (version 3.35) on an area of 7.5x2.0 km^2. The routing protocol is AODV and the topology consists of 31 nodes where one is a sink and placed at the rightmost edge in the y-axis center. The sink node altitude is 2 m, while the swarm nodes fly at 10 m altitude.

The sink node is static, while all others are moving. The nodes move according to the Random Waypoint Mobility Model (RWMM) with no pause time. This means that each node will pick a random waypoint (in the simulation area) and move towards it with a constant speed until the waypoint has been reached. The process then repeats immediately. Note that the node pauses its motion when frozen, and resumes moving when the frozen status is timed out.

The packet size is 1250 bytes plus IP and UDP headers. The Maximum Transmission Unit (MTU) is below the fragment threshold, i.e., no packets are fragmented at the MAC layer. The traffic load depends on the number of flows. It is tuned so that all sources generate and send one packet per second. For one flow this amounts to a load of 10 kbps. At 30 flows, this amounts to a load of 300 kbps. The traffic pattern is as follows: for each simulation, one or multiple nodes (depending on the number of packet flows simulated) generate traffic for 15 s, after which another randomized set of nodes start sending traffic for 15 s. The traffic is directed from one or multiple random nodes (1 to 31) towards the sink node (node 0). Request-to-Send/Clear-to-Send (RTS/CTS) is enabled for all unicast packets. Simulation results with varying number of flows are included to illustrate the effect on node mobility, and thus the amount of time where the swarm application itself has control over the mobility pattern.

The simulations employ the two-ray ground propagation model, which dictates that each wireless link consists of a line-of-sight component and a single ground-reflected component.

The IEEE 802.11 MAC is employed in these simulations. It has an interface to notify the routing protocol of link breaks using LLN, and thus AODV HELLO packets are disabled. Further, to fully take advantage of the freeze on routing signaling, we depart from the default AODV setup in two ways: 1) the ”Gratuitous Reply” (G)-flag is unset and the ”Destination Only” (D)-flag is set. The rest of the simulation parameters are shown in Table 1.

6.2. Sustaining a unicast flow

We begin by examining the results for simulations with one unicast flow (Fig. 1), comparing the three different freeze methods with baseline AODV. The Packet Delivery Ratio (PDR) (Fig. 1 (TL)) for the baseline AODV (*base_1*) are clearly affected by the increasing node velocity, falling from almost 100% at 1 m/s to 82% at 20 m/s.

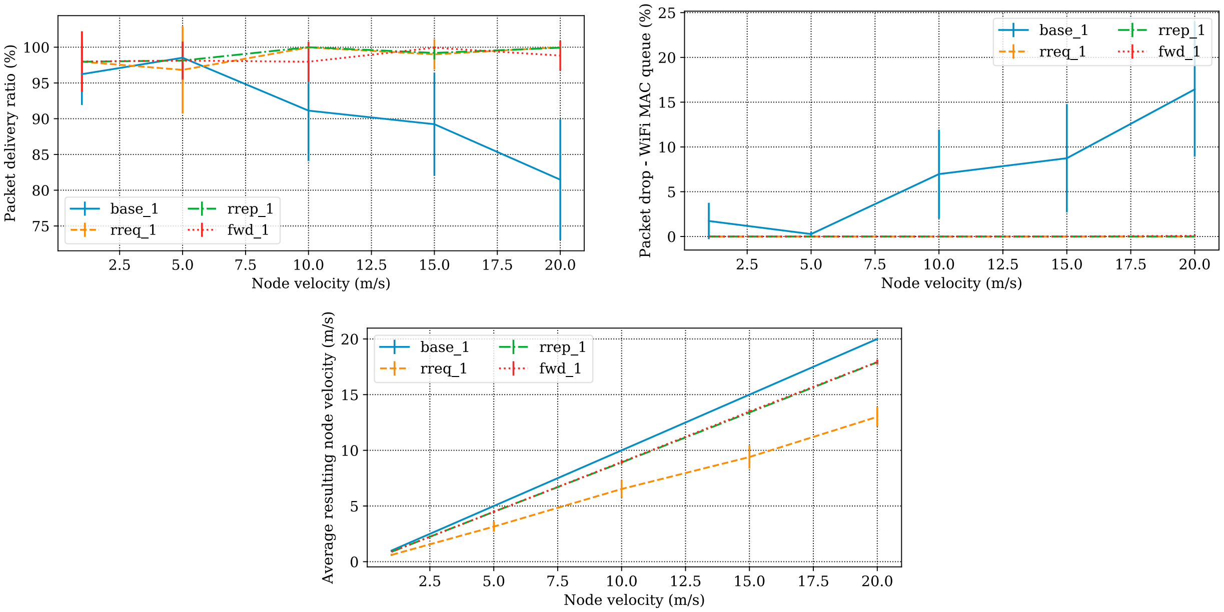

Fig. 1: Simulation results for sustaining one unicast flow. Results for standard AODV (*base*), *rreq*, *rrep* and *fwd* for increasing node velocities: (TL) Packet delivery ratio; (TR) Packet loss from the WiFi MAC queue; (B) Average resulting velocity.

The main reason for the declining PDR is the failure to successfully transmit already routed packets to the next hop. Packets are lost from the MAC transmission queue (Fig. 1 (TR)) because of the input rate exceeding the output rate and packets timing out from the queue (maximum time in queue is 0.5 s). In our configuration, the total offered traffic is below the congestion point, and the reason for the growing queue is mobility, whereby packets are tried sent to unreachable next hop addresses. An attempted transmission to an unreachable next hop node leads to head-of-queue blocking [7]. This is because the node has to unsuccessfully transmit the stale packets until the transmission threshold expires before discarding the packets. Furthermore, many Carrier Sense Multiple Access / Collision Avoidance (CSMA/CA) protocols use exponential backoff in the event of transmission failure, in addition to defer time when detecting an ongoing transmission. Therefore, the node network capacity decreases exponentially with the number of failed transmissions and detected ongoing transmissions. Consequently, the queued packets experience longer queuing times, and the MAC transmission queue will grow as the transmission time increases. In mobile wireless networks, the challenge is therefore mainly to reduce the influence of error prone links and stale route entries into the routing protocol.

The difference between the baseline and the freeze solutions' results is explained by:

1. The freeze solutions provide stable paths.
2. Packets are mainly discarded at the source before being transmitted. Head-of-queue blocking [7] is mitigated.

The RREQ-freeze method benefits from E2E stable paths (1), which again reduces the mobility associated challenges. With RREQ-freeze, packets that are otherwise discarded due to stale next hop addresses are preserved. As a result, nodes spend less time incorrectly in back-off, thus reducing the packet loss from the MAC transmission queue and improving the channel utilization, further achieving higher PDR.

Freeze shifts the packet drop probabilities from the intermediate nodes back to the source (2). A consequence of stable paths is reduced packet drop due to mobility issues. Hence, packet drop at intermediate nodes are mainly caused by hidden node or other medium caused challenges and is further related to traffic volume and traffic pattern. In low traffic situations, lost packets are mainly caused by route discovery failures and thus discarded before handed to IP forwarding.

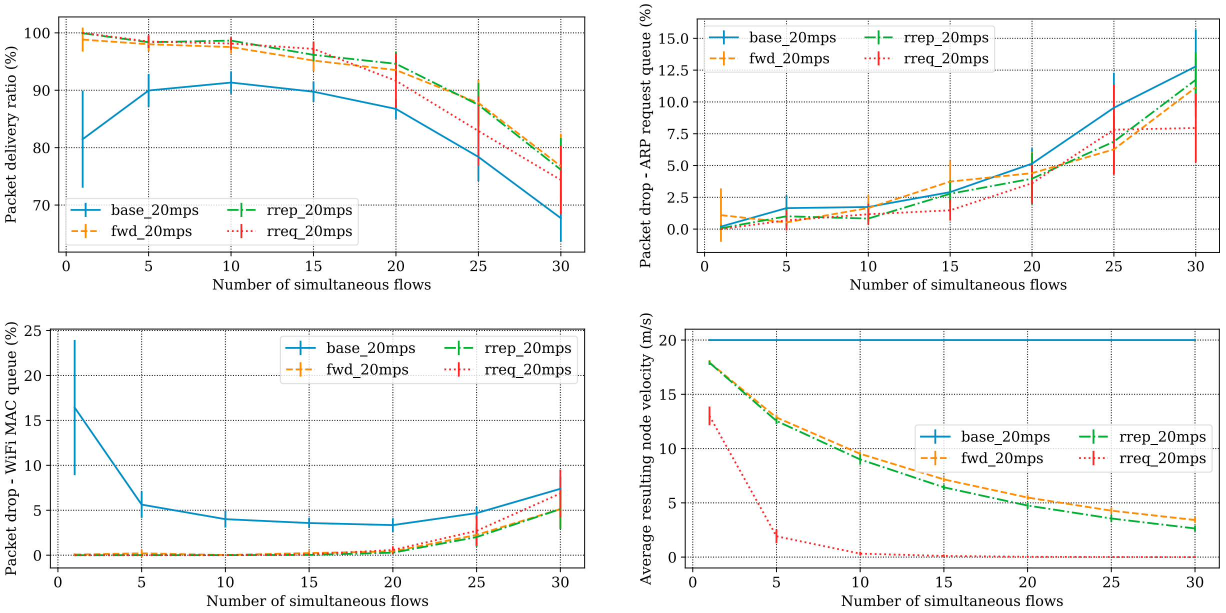

Fig. 2: Results for all methods in a topology with 20 m/s velocity for an increasing number of simultaneous flows: (TL) Packet delivery ratio; (TR) Packet loss from the ARP request queue; (BL) Packet loss from the WiFi MAC queue; (BR) Average resulting velocity.

There is also some loss caused by packet drops due to route error. This loss is stable at around 2%, and is caused by the topology being partitioned at some times, which limits the success ratio of the routing protocol.

Looking at the PDR results for the three freeze solutions (Fig. 1 (TL)), we see that all three are successful in limiting the impact of increasing mobility on the PDR. There is almost no packet loss at 20 m/s for any of the three solutions. The three solutions perform significantly better than the baseline results at increasing velocities.

Looking at the resulting velocity of the swarm (Fig. 1 (B)), we see that the *rreq* is the most invasive solution, freezing as early as possible, and impacting the most nodes. The result is the lowest average velocity of the solutions and the baseline. The baseline results show no reduction in average velocity, as expected. The *rreq* reduces the velocity by around a third. The *rrep* and *fwd* solutions are indistinguishable, both reducing the average velocity by around 10% of the set velocity.

Node mobility (Fig. 1 (B)) is correlated to the configured freeze method combined with the time distribution for route discovery. Each received RREQ pauses the receiving node for a configured time. Clearly, the time distribution for when initiating route discovery impacts both the number of nodes freezing, and their accumulated time spent in freeze as seen in Fig. 1 (B). Since *rreq* freezes all nodes, the only parameter a network operator can optimize is when to start new route discovery with regards to node velocity.

To summarize the one flow results, the proposed solutions are all capable of sustaining a unicast traffic flow in a swarm topology at high mobility. For *rrep* and *fwd* there is little reduction of the resulting average node velocity. However, for *rreq*, the reduction is around a third. And with a higher number of simultaneous flows, the impact on the resulting average velocity is expected to increase further.

6.3. Sustaining multiple simultaneous unicast flows

Fig. 2 shows simulation results for a topology with 20 m/s node velocity at an increasing number of simultaneous flows. Fig. 2 (TL) shows that for a topology at 20 m/s, the baseline AODV (*base_20mps*) first increases its PDR with an increasing number of flows, before it starts to plummet at 20 simultaneous flows. However, the PDR for baseline AODV is outperformed by all three freeze solutions.

The three freeze solutions all achieve their best performance at one flow. They do not experience an increase in PDR as the number of flows grows. This is in contrast with the baseline results. The reason for this contrast is as follows. The freeze solutions will only trigger a lasting freeze in an unpartitioned topology between the source and the sink. Only when there is a viable path between the source and the sink, this subset of the topology will remain frozen. Until that time, the topology, or parts of the topology, stays mobile until the path can be established. Thus, it is an inherent mechanism of the freeze solutions to work towards an unpartitioned topology. This is particularly easy to see with the *rrep* and *fwd* solutions. The nodes will only freeze when the RREP is successfully delivered back to the RREQ source and data is forwarded. This requires a connected E2E path. For the *rreq* solution, larger parts of the topology may be frozen by the RREQ. This happens without a guarantee that the topology from the source to the sink is unpartitioned. However, the nodes beyond the reach of the RREQ will stay mobile. This increases the probability of them being in a better position at a later time. A better position is one where they can be part of connecting the source(s) and the sink.

The PDR results (Fig. 2 (TL)) show that the three freeze solutions have some resilience to the increasing number of flows, compared to the baseline results, but nevertheless there is a decline. At a lower number of flows, the PDR for the *fwd* solution is marginally lower than the two other solutions, while at a high number of simultaneous flows, the *rreq* seems to struggle a little more than the two others. The decline of the PDR at increasing number of flows can be explained by the combination of concurrent flows, and the dependence on the reactive routing protocol AODV. Hence, the observed results are not strictly correlated to our freeze method, but more towards AODV as a reactive routing protocol and its associated traffic load.

The reason for the PDR decline with the number of flows (Fig. 2 (TL)) is attributed to two main causes:

1. The inability to sufficiently regulate searches in AODV.
2. Stale route entries and resulting ARP request queue loss.

To the first point, AODV does not have mechanisms to balance the ongoing number of searches according to the network carrying capacity. In a situation where there are multiple sources, and especially if the sources are searching for the same destination, a search storm can occur. This problem is amplified in sensor networks where multiple sensors are triggered by the same observation. Multiple sensors triggered by the same observation at the same time searching for the same sink will lead to a search storm.

To the second point, the steady decline of the PDR for all solutions at the higher number of flows is directly connected to the packet loss from the ARP request queue (Fig. 2 (TR)), due to the traffic flows sharing a common destination. The impact is higher with increased distance from the sources towards the common destination or gateway.

Further, we observe that the packet loss from the MAC transmission queue (Fig. 2 (BL)) is very low for the three freeze solutions. Interestingly, the relative packet loss from the MAC transmission queue is first reduced for the baseline results before they start growing with a higher number of simultaneous flows. The reason for the initial reduction is explained by the number of simultaneous flows reducing the impact of any source node partitioned from the sink. At the higher number of simultaneous flows, the MAC transmission queue loss starts growing even for the three freeze solutions, indicating an amount of queue blocking. The probability of queue blocking increases closer to the sink node.

The resulting average node mobility (Fig. 2 (BR)) shows how the number of freezing nodes and their accumulated time in freeze correlates to AODV and its ERS configuration, the number of route searches and the traffic flows' time distribution and average flow duration. Since *rreq* stops node mobility on received RREQ packets, the average node mobility declines more with the number of nodes and their searches, compared to the other freeze solutions. The *rrep* has a lower impact on the swarm's mobility than *rreq*. For one flow, the average resulting velocity is reduced by half, compared to *rreq*, but for increasing number of simultaneous flows *rreq* quickly brings the topology to a stop, while the two other freeze solutions are less impacted by the growing amount of simultaneous flows.

6.4. General observations

Surveillance swarm applications require swarm nodes to have a proper geographic distribution. The act of freezing nodes reduces the number of mobile sensors and hence also the swarm's surveillance capabilities. Manipulating the physical behavior of the nodes thus lead to unexpected and unwanted swarm effects. We have observed a clumping effect of the freeze solutions. This behavior is more pronounced for the *rreq* than for the *rrep* (Fig. 3). The use of a swarm-coordination algorithm instead of the RWMM would counter this effect.

12
9
1
22
20
10
6
7
28
30
19
4
25
29
15
11
16
0
1
23
22
14
3
28
7
10
17
5
2
8
16
29
13
24
20
26
9
4
11
19
27
0

Fig. 3: End state for two simulated topologies after 300 s with RREQ-freeze and RREP-freeze: (L) RREQ-freeze; (R) RREP-freeze.

7. Conclusions and future work

In this paper, we have evaluated three solutions to sustain unicast flows for sensor traffic. With the solutions, a subset of the nodes in the swarm freeze to sustain temporary flows of unicast traffic. There is a trade-off between stable paths for the unicast flows and the mobility of the swarm. Freezing on RREQ gives the most stable environment for unicast, but is at the same time the most invasive solution, reducing the mobility the most. At the other end, triggering freeze on forwarded data traffic has the lowest impact on mobility, and is able to maintain throughput performance with increasing mobility. Freezing on RREP has similar throughput performance as freezing on forwarded traffic, although RREP-freeze is expected to be better for long paths where the path is at an increased risk of breaking in the route setup time. Although this study has focused on the AODV signaling as freeze trigger mechanism, freezing on forwarding seems to be able to provide stable paths for unicast forwarding. This indicates that the mechanism could be helpful in a network without AODV signaling as well.

The results are very promising, but the presented solutions need to be better integrated with swarm coordination algorithms to be of any practical use. The solutions in this paper demonstrate only basic methods for improving conditions for unicast flows in swarm networks. Future work in this direction should include integration with swarm coordination algorithms and exploiting methods to tune the selection of freezing nodes, e.g., a combination of willingness and a more nuanced freeze on RREQ. Another venue to explore is to address the observed challenges with regards to freeze and the challenge of multiple simultaneous unicast flows. When a sensor has something to report, there is high probability that others nearby will desire to report the same information. If left unhandled, this creates an information storm, creating collisions and congestion, in addition to bringing the mobility of the swarm down.

References

[1] , 2022. Network Simulator 3. https://www.nsnam.org/. Last accessed 2022-02-04.

[2] İlker Bekmezci, Sahingoz, O.K., Şamil Temel, 2013. Flying Ad-Hoc Networks (FANETs): A survey. Ad Hoc Networks 11, 1254–1270. URL: https://www.sciencedirect.com/science/article/pii/S1570870512002193, doi:https://doi.org/10.1016/j.adhoc.2012.12.004.

[3] Benzaid, M., Minet, P., Al Agha, K., 2002. Integrating fast mobility in the olsr routing protocol, in: 4th International Workshop on Mobile and Wireless Communications Network, IEEE. pp. 217–221.

[4] Das, S.R., Perkins, C.E., Belding-Royer, E.M., 2003. Ad hoc On-Demand Distance Vector (AODV) Routing. RFC 3561. URL: https://rfc-editor.org/rfc/rfc3561.txt, doi:10.17487/RFC3561.

[5] Fall, K., 2003. A delay-tolerant network architecture for challenged internets, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, pp. 27–34.

[6] Landmark, L., Larsen, E., Fongen, A., Kure, Ø., 2017. Improving simplified multicast forwarding using an elevated relay node, in: 2017 International Conference on Selected Topics in Mobile and Wireless Networking (MoWNeT), IEEE. pp. 1–6. doi:10.1109/MoWNet.2017.8045957.

[7] Landmark, L., Øvsthus, K., Kure, Ø., 2007. Alternative Packet Forwarding for Otherwise Discarded Packets, in: Future Generation Communication and Networking (FGCN 2007), pp. 8–15. doi:10.1109/FGCN.2007.52.

[8] Landmark, L., Øvsthus, K., Kure, Ø., 2008. Test-Bed Evaluation of Ingress Queuing for Improved Packet Delivery, in: Fourth International Conference on Networking and Services (ICNS 2008), pp. 102–108. doi:10.1109/ICNS.2008.13.

[9] Larsen, E., Landmark, L., Engebråten, S.A., Kure, Ø., 2019. Practical optimization methods for uav relay positions in an adaptive rate manet, in: In Proceedings from 2019 International Conference on Military Communications and Information Systems (ICMCIS), IEEE. pp. 1–8. doi:10.1109/ICMCIS.2019.8842664.

[10] ur Rahman, S., Kim, G.H., Cho, Y.Z., Khan, A., 2018. Positioning of uavs for throughput maximization in software-defined disaster area uav communication networks. Journal of Communications and Networks 20, 452–463. doi:10.1109/JCN.2018.000070.

[11] Teacy, W.L., Nie, J., McClean, S., Parr, G., 2010. Maintaining connectivity in uav swarm sensing, in: 2010 IEEE Globecom Workshops, IEEE. pp. 1771–1776.

[12] Yanmaz, E., 2012. Connectivity versus area coverage in unmanned aerial vehicle networks, in: 2012 IEEE International Conference on Communications (ICC), IEEE. pp. 719–723.