



**HAL**  
open science

## **S-ROGUE: Routing protocol for Unmanned Systems on the Surface**

Sébastien Bindel, Serge Chaumette, Benoit Hilt, Pascal Lorenz

► **To cite this version:**

Sébastien Bindel, Serge Chaumette, Benoit Hilt, Pascal Lorenz. S-ROGUE: Routing protocol for Unmanned Systems on the Surface. International Conference on Communications ICC 2017, IEEE, May 2017, Paris, France. 10.1109/ICC.2017.7997082 . hal-01529287

**HAL Id: hal-01529287**

**<https://hal.science/hal-01529287>**

Submitted on 30 May 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# S-ROGUE: Routing protocol for Unmanned Systems on the Surface\*

Sébastien Bindel, Serge Chaumette

LaBRI, University of Bordeaux

Talence, France

Email: {sebastien.bindel, serge.chaumette}@labri.fr

Benoit Hilt, Pascal Lorenz

MIPS GRTC laboratory, University of Haute Alsace

Colmar, France

Email: {benoit.hilt, pascal.lorenz}@uha.fr

**Abstract**—The cooperation of heterogeneous unmanned systems, for instance, between aerial engines and terrestrial engines, relies on reliable communication. Data delivery is ensured by routing protocols, but traditional routing approaches, MANET and DTN, are not efficient in such networks. In this paper, we propose the S-ROGUE routing protocol combining the paradigms MANET and DTN and switching between them according to the network connectivity. On the one hand, the S-ROGUE MANET algorithm relies on a proactive approach and a novel metric to anticipate link disruptions and detect unidirectional links. On the other hand, the S-ROGUE DTN algorithm uses on a reinforcement learning technique to select the best routing action. It implements also a replication control and packet prioritization to improve routing performances. We lead a performance evaluation of S-ROGUE with similar routing protocols in realistic simulated environments and conclude that S-ROGUE has the best routing performance regardless the scenarios.

## I. INTRODUCTION

Unmanned Systems are becoming an emerging topic and found a wide range of applications. These systems have been defined by the National Institute of Standards and Technology (NIST) [1], which describes them as *a powered physical system, with no human operator aboard the principal components, which acts in the physical world to accomplish assigned tasks*. It may be mobile or stationary and includes both the vehicle and the associated components such as the control unit. We consider only surface systems including, aerial, terrestrial and surface maritime systems. We assume they share the same type of network interface, unlike submarines which use acoustic modems [2]. Each system is designed to ensure an optimal accomplishment of its assigned mission. However, complex missions such as the Rapid Environmental Assessment (REA) [3] require the cooperation of heterogeneous systems as depicted in figure 1.

Cooperation becomes effective only if data are delivered correctly. This task is supported by routing protocols. Routing paradigms can be classified into four categories according to the node velocity and density as depicted by figure 2. MANET routing protocols assume the existence of paths between among all nodes in the network, unlike in Delay Tolerant Networks (DTN) protocols, where nodes meet sporadically each other. Assisted DTN (A-DTN) is a special type of DTN

where the density and the velocity of nodes are low. The last category includes the Hybrid routing approach merging both MANET and DTN approaches to adapt the routing policy according to the current network connectivity. Regarding the speed of aerial, terrestrial and maritime vehicles and their formations in fleet or in swarm, we assume that the density is unpredictable and evolve in the course of time. From these assumptions, the hybrid routing paradigm appears as the best to ensure data delivery in networks of unmanned systems.

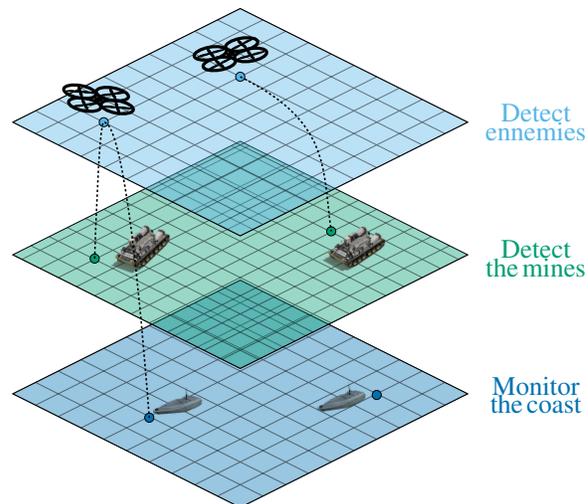


Fig. 1. Examples of tasks assigned to vehicle during a REA mission

Several hybrid protocols have been proposed in the literature, but none takes into account all requirements of DTN networks. In [4] and [5], authors extend proactive protocols (BATMAN and OLSR) with a store and forward mechanism. However, the proposed solutions delays only the transmission if a destination is unreachable, without taking into account sporadic meetings. In [6] authors extend the Dynamic MANET On-demand Routing (DYMO), without taking into account the delivery deadline related to each packet. Conan et Whitbeck [7] have developed the HYMAD protocol. It relies on a clustering technique to determine the routing strategy. Inside a cluster, the routing is ensured by a MANET protocol, meanwhile a DTN routing is used for inter-group communication. However, such an approach requires a low mobility of the nodes in a cluster to guarantee a good performance

\*This is a detailed version of the paper presented at the IEEE ICC 2017 conference (Paris, FR)

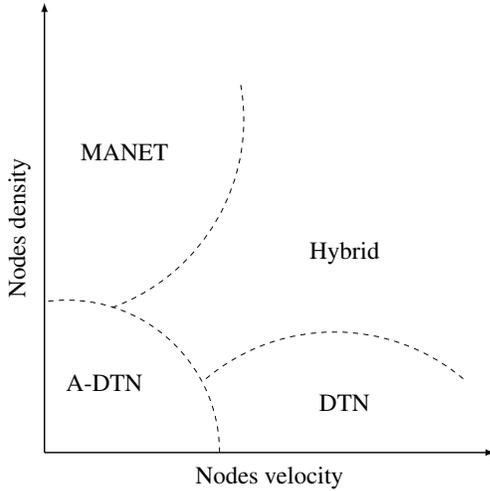


Fig. 2. Routing paradigm vs. nodes mobility and nodes density

of the MANET protocol. Azzuhri et al. [8] have extended OLSR by including a DTN algorithm triggered when a path is broken. Even if authors take into account the packet delivery deadline and use a replication mechanism, they spread packets regardless the capacity of the neighbor to meet the desired destination.

Regarding the lacks of current protocols, we propose the S-ROGUE, a hybrid routing protocol dealing with the requirements of MANET and DTN paradigms. The MANET algorithm relies on a proactive approach to discover and assess both local links and end-to-end paths. Beside, the DTN algorithm relies on the reinforcement learning (Q-learning) in order to select the best routing action and uses a replication control and packet prioritization to improve routing performances.

The remainder of this paper is organized as follows. In Section II, we describe the reinforcement learning approach and detail the Q-learning algorithm. In Section III, we describe carefully our proposed hybrid routing protocol. We detail in Section IV simulation cases and settings to conduct our performance evaluation. In Section V, we give simulation results. At the end, we conclude our works and present perspectives in Section VI.

## II. REINFORCEMENT LEARNING APPROACHES AND Q-LEARNING ALGORITHM

Optimizing the routing performances in an unknown and dynamic environment is a key issue in mobile networks. Reinforcement Learning techniques (RL) provide an interesting approach for solving such a problem. We introduce the concept of RL and describe the Q-learning algorithm used in our routing algorithm.

### A. Reinforcement learning

In a RL approach a system learns how to solve a problem from its experience [9]. A learning agent interacts with its environment by performing actions and obtains a reward and switches to another state as depicted in figure 3. Initially, the

learning agent has any experiences and ignores which action provides the best reward. As a consequence, the learning agent must explore the environment.

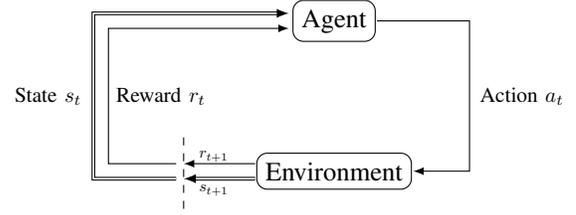


Fig. 3. The agent environment interaction in reinforcement learning

A reinforcement learning task that satisfies the Markov property is called a Markov Decision Process (MDP). We consider a finite Markov decision process, the number of states and actions spaces are finite. Formally, we describe the characteristics of a MDP with the following tuples  $\langle S, A, P, R \rangle$ , each one represents the set of states, actions, state transitions and rewards, respectively. In each iteration, a learning agent is in a state  $s \in S$ , and performs an action  $a \in A$ . The probability to be in the next state  $s'$  is given by the transitions function  $P_{s,s'}^a$  described by :

$$P_{s,s'}^a = Pr \left\{ s_{t+1} = s' \mid s_t = s, a_t = a \right\},$$

$$\text{with } \sum_{s' \in S} P_{ss'}^a = 1. \quad (1)$$

The related expected reward function is given by the following equation:

$$R_{s,s'}^a = E \left\{ r_{t+1} \mid s_t = s, a_t = a, s_{t+1} = s' \right\}. \quad (2)$$

According to the state-action pairs, an agent can expect to receive a reward according to its future actions. To define its behavior we use a policy  $\pi$ , mapping from each state  $s \in S$  and action  $a \in A$  the probability  $\pi(s, a)$ . The value of a state  $s$  under the policy  $\pi$  is denoted  $V^\pi(s)$  and is defined as the total of expected rewards,  $R_t$ , obtained by the learning agent starting from the state  $s$  and following the policy  $\pi$ . The function  $V^\pi(s)$  is defined as follows:

$$V^\pi(s) = E_\pi \{ R_t \mid s_t = s \} = E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s \right\}, \quad (3)$$

with  $E$  the expected value under the policy  $\pi$ ,  $\gamma$  the discount factor that determines the importance of future rewards such as  $0 \leq \gamma \leq 1$ . We can also define from an action  $a$  for a learning agent in a state  $s$  using a policy  $\pi$  the total value of rewards denoted  $Q^\pi(s, a)$  such as:

$$Q^\pi(s, a) = E_\pi \{ R_t \mid s_t = s, a_t = a \},$$

$$= E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s, a_t = a \right\}. \quad (4)$$

It is possible to find an optimal solution to a given problem with the Bellman equation [9]:

$$\begin{aligned} V^\pi(s) &= E_\pi \{R_t | s_t = s\}, \\ &= E_\pi \left\{ \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} | s_t = s \right\}, \\ &= E_\pi \left\{ r_{t+1} + \gamma \sum_{k=0}^{\infty} \gamma^k r_{t+k+2} | s_t = s \right\}. \end{aligned} \quad (5)$$

In praxis, finding such a solution in an unknown system is impossible. This is why, we try to find a sub-optimal solution with the Q-learning algorithm described in the next section.

### B. Q-learning

Q-learning is a reinforcement learning algorithm based on the state-action function  $Q(s, a)$ , which approximates the optimal state-action function and is given by:

$$Q(s_t, a_t) \leftarrow (1 - \alpha)Q(s_t, a_t) + \alpha[r_t + \gamma \max_a Q(s_{t+1}, a)], \quad (6)$$

with  $V^*(s) = \max_a Q^*(s, a)$  the value of the given state  $s_{t+1}$  and  $\alpha \in [0, 1]$  the learning rate determining how much  $Q(s_t, a_t)$  weighs the reward at each iteration. We formulate in section III-B the routing problem in an MDP and use the Q-learning algorithm to find a sub-optimal solution. The Q-learning algorithm is detailed in the algorithm 1.

---

#### Algorithm 1: Q-learning algorithm

---

```

1 Initialize  $Q(s, a)$ ;
2 while for each episode do
3   Initialize  $s$ ;
4   while for each step of episode do
5     Choose  $a$  from  $s$  using policy derived from  $Q$ ;
6     Done action  $a$ , observe  $r, s'$ ;
7      $Q(s, a) \leftarrow$ 
8        $Q(s, a) + \alpha[r_t + \gamma \max_{a'} Q(s', a') - Q(s, a)]$ ;
9      $s \leftarrow s'$ ;
10  end

```

---

## III. S-ROGUE DESIGN

In this section, we describe the design of S-ROGUE. We start with a description of its MANET strategy used to discover local links and end-to-end paths. Then, we describe the DTN algorithm relied on the Q-learning algorithm and its associated mechanisms. In each section, we take care to describe how our protocol switches between these strategies.

### A. MANET routing algorithm

The use of the MANET paradigm allows us to discover end-to-end paths. S-ROGUE uses a proactive routing algorithm based on a controlled flooding technique to reduce the routing overhead. Information about links is provided by the Fast-ETX metric (F-ETX), which assesses both the quality and the state of local links [10]. This information is also used by the algorithm to assess the quality of end-to-end paths.

1) *Local link discovery and assessment*: The algorithm uses a proactive scheme, wherein all nodes broadcast periodically announcement messages to their neighborhood. These messages are also used by F-ETX metric to assess the quality and the state of a link. This metric requires two information: (i) the number of packets transmitted to a neighbor with success and (ii) the number of acknowledgment returned by this one. Such a technique requires the exchange of the first information, but if a packet is lost, a node cannot determine if the last transmitted is lost or received. As a consequence, a delay is introduced during the assessment process. To avoid such a situation, we propose another solution relied on the technique used by the better approach to mobile ad-hoc networking (BATMAN) routing protocol, by changing the transmission process as depicted by the figure 4. The  $d_r$  ratio indicates

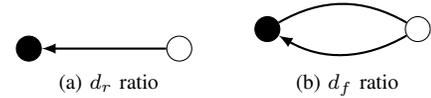


Fig. 4. Local link assessment

the number of packets received from a neighbor, meanwhile The  $d_f$  ratio indicates the number of forwarded packets by a neighbor. This technique ensures the acquisition of information required by the F-ETX metric and let to a same node the computation of the ratios. In order to avoid an infinite forwarding of announcement messages, a node receiving its announcement message does not retransmit it. The local link computation is giving is the algorithm 2 with  $SrcAddr$  the address of the message creator and  $PrevAddr$  the address of the last forwarder. In order to be identified an unique sequence number is assigned to each packet.

---

#### Algorithm 2: Packet announcement policy

---

```

input: packet: received packet
          addr: the node's address
1 if  $packet.SrcAddr = packet.PrevAddr$  then
2   ComputeDr();
3    $paquet.PrevAddr \leftarrow addr$ ;
4   SendPacket(packet);
5 else if  $SrcAddr = addr$  then
6   ComputeDf();
7   RemovePacket(packet);
8 else
9   RemovePacket(packet);
10 end

```

---

2) *Routing path discovery process*: The forwarding process ensures the path discovery, by propagating announcement messages through the network. These messages contain information about the path quality ( $Q_p$ ), computed from three information: (i) the announced path quality ( $Q_{ap}$ ), (ii) the local link quality ( $Q_{ll}$ ) and (iii) the number of hops ( $P_h$ ) such as,  $Q_p = Q_{ap} \times Q_{ll} \times P_h$ . In praxis information is coded on 8

bits. As a result, the path quality is computed by the following equation.

$$Q_p = \frac{Q_{ap} \times Q_{ll} \times P_h}{(Q_{max})^2}, \quad (7)$$

with  $Q_{max}$  the maximal link quality (255). When an announcement message is created, the information on the quality is fixed at 255. Then, the announced quality decreases at each forwarding according to the quality of local link and the number of hops. The aim of the hop count information is to penalize the path quality according to a penalty  $P$  related to the number of nodes presents in the path and is computed as  $P_h = Q_{max} - P$ . We assume the maximal quality at 255 such as  $Q_{max} = 255$ , and fix empirically  $P$  at 10, like in the BATMAN protocol. An example of the link quality path assessment is given in the figure 5. The node A is directly connected to B with a quality of 70%, as a result, it assesses the local link with a quality of  $\lfloor \frac{255 \times 70}{100} \rfloor = 178$ . When A receives the announcement message of C forwarded by B, it assesses the quality of the routing path with a quality of  $\lfloor \frac{255 \times 178 \times 245}{255^2} \rfloor = 171$ .

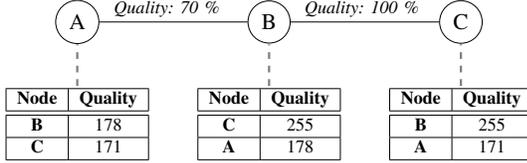


Fig. 5. Assessment of the path quality

We define two mechanisms to enhance the path selection process when a disruption occurs. The first one is triggered when a path is no longer announced, the related quality is decreased. The second one is triggered when the path is no longer announced after two periods, the related routing entry in the routing table is declared unreachable. All packets destined to the destination are then managed by the DTN algorithm.

3) *Integration of the F-ETX metric:* We have developed the Fast ETX (F-ETX) metric for assessing the quality and determine the link state in mobile networks [10]. Unlike other estimators, F-ETX uses a dynamic window size to catch packets, in order to compute the couple of ratios ( $d_f$  and  $d_r$ ). The estimator adapts its reactivity and its accuracy according to the link stability, that makes it suitable for mobile networks. F-ETX provides four types of information through four estimators.

The first estimator gives a short-term assessment of the link quality and is computed as follows:

$$\chi^{LQ} = \frac{1}{(1 - d_f)(1 - d_r)}. \quad (8)$$

The second estimator gives a long-term assessment of the link quality by computing the variation of the current and the previous estimations to track the course of the link quality:

$$\begin{aligned} \Delta_t^{LQ} &= \chi_t^{LQ} - \chi_{t-1}^{LQ}, \\ \chi_t^{Trend} &= \beta \cdot \Delta_t^{LQ} + (1 - \beta) \cdot \chi_{t-1}^{Trend}. \end{aligned} \quad (9)$$

The third estimator computes the link stability by taking into account the absolute and relative size of the windows maintained for computing the couple of ratios. Let  $W_{max}$  the maximum window size,  $W_n$  the current window size and  $W_i$  the  $i^{th}$  the element in the window. The windows maintained to compute the  $d_f$  and  $d_r$  respectively noted  $W^{d_f}$  and  $W^{d_r}$ . The link stability indicator is computed with an EWMA (Exponentially Weighted Moving Average) filter tacking into account the absolute  $\Xi$  and the relative stability  $\xi$ :

$$\begin{aligned} \Xi &= \frac{\sum_{i=1}^{|F_n^{d_f}|} F_i^{d_f} + \sum_{i=1}^{|F_n^{d_r}|} F_i^{d_r}}{2|F_{max}|}, \\ \xi &= \frac{\sum_{i=1}^{|F_n^{d_f}|} F_i^{d_f} + \sum_{i=1}^{|F_n^{d_r}|} F_i^{d_r}}{|F_n^{d_f}| + |F_n^{d_r}|}, \\ \chi_t^{Stab} &= \Xi_t \cdot \gamma + (1 - \gamma) \cdot \xi_t. \end{aligned} \quad (10)$$

The fourth estimator observes the changes in the window sizes maintained for the couple of the ratios. Let  $W$  be a window and  $W_n^t$  its size at time  $t$ . The variation of the reception ratio provided by the window  $W$  at  $t$  is noted  $\Delta_t^{Win}$ . This estimator determines the link as becoming unidirectional if the assessed value becomes negative, and is computed as follows:

$$\begin{aligned} \Delta^{Win} &= \sum_{i=1}^{|F_n|} F_i - \sum_{i=1}^{|F_{n-1}|} F_i, \\ \chi_t^{ULL} &= \chi_{t-1}^{ULL} \cdot \lambda + (1 - \lambda) \cdot \varphi(\Delta_t^{d_f}, \Delta_t^{d_r}), \\ \text{with } \varphi(x, y) &= \begin{cases} -1 & x < 0 \wedge y > 0 \\ 1 & x > 0 \wedge y < 0 \\ 0 & \text{else} \end{cases}. \end{aligned} \quad (11)$$

Each estimator assesses a specific property of a link. Current approaches as proposed Baccour et al. [11] merge all estimations into a single one. However, there is no silver bullet to merge all estimations and provide a reliable information on the link. In [10] we have proposed a framework, depicted figure 6, to integrate the F-ETX to a routing protocol by treating separately each estimation. The stability information is treated in first by the routing algorithm since it determines if a route is valid or not. Then, the algorithm checks if the link is unidirectional by observing the values returned by the fourth estimator and the  $d_f$  ratio. At the end, the algorithm merges short and long-term estimations in order to penalize link with decreasing quality and reward links with an increasing quality.

### B. The DTN routing algorithm

When nodes cross sporadically each other, MANET routing strategies become inefficient. In this case, DTN approaches appear as the best. We have developed a routing DTN algorithm to support the MANET algorithm, when any paths are available. In order to perform an optimal routing, we map the Q-learning algorithm to the routing problem. In addition,

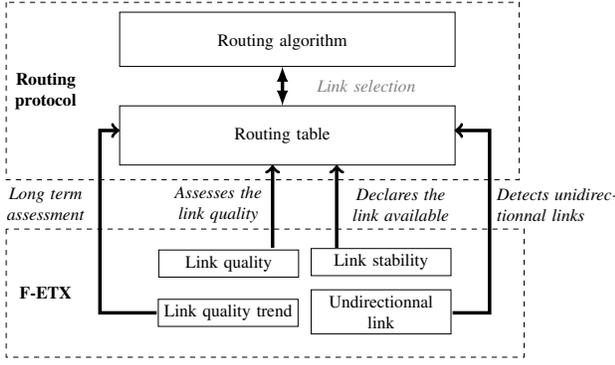


Fig. 6. Framework suggested by Bindel et al. [10]

the protocol quantifies meeting occurrences and uses packet prioritization and a replication mechanism to increase the Packet Delivery Ratio (PDR) and reduce the end-to-end delay.

1) *Routing algorithm*: Our DTN algorithm relies on the Q-learning algorithm. We define the state in the Q-learning as a pair of nodes  $S(a, b)$  such as a node  $a$  holds a packet and is in contact with  $b$ . Routing decisions are mapped to a couple of actions: either transmitting the packet to the node or keeping it. We define also two rewards,  $r_t$  the reward related to the transmission and  $r_k$  related to the storing action.

Rewards are the key points of the Q-learning algorithm, since they determine the behavior of the learning agent. Rewards must take into account the capacity of a node to cross the desired destination and its lifetime determining its capacity to hold a packet. The end-to-end delay can be estimated according to the meeting occurrences between nodes. Such information is given by contact metrics, described in the section III-B2. Let  $n$  a node holding a packet to a desired destination  $d$  and meeting a node  $m$ . The node  $n$  compares its meeting interval with  $d$  and that provided by  $m$ , denoted  $P(n, d)$  and  $P(m, d)$  respectively. The reward function related to the capacity of a node to cross the destination is computed as follows:

$$D(n, m) = \begin{cases} 0 & P(n, d) < P(m, d) \\ -0.5 & P(n, d) = P(m, d) \\ -1 & P(n, d) > P(m, d) \end{cases} \quad (12)$$

We consider also in the routing decision the residual energy of a node. Let a node  $n$ , the reward function related to the residual energy is computed as follows:

$$E(n) = - \left( 1 - \frac{E_{res}(n)}{E_{init}(n)} \right), \quad (13)$$

with  $E_{res}(n)$  and  $E_{init}(n)$  are the residual energy and the initial energy of the node  $n$ , respectively. Combining equations 12 and 13, we define the reward  $r_t$  related to the packet transmission from a node  $n$  to a node  $m$  as:

$$r_t = \alpha E(m) + \beta D(n, m), \quad (14)$$

with  $\alpha$  and  $\beta$  weighting the reward functions. Beside, the reward  $r_k$  related to the packet storing by a node  $n$  is computed as follows:

$$r_k = \alpha E(n) + \beta D(n, m). \quad (15)$$

2) *Metrics*: Assessing meeting occurrences gives information on the capacity of a node to cross a desired destination. Our DTN routing algorithms relies on this information to determine if a packet has to be transmitted or stored. We have extended prediction contact techniques used in [12] by developing four metrics presented in the following paragraphs.

a) *Average contacts metric*: The assessment of average contacts,  $\rho$ , gives an information on the percentage of nodes encountered during a time interval. Let  $n_T$  the total number of cross nodes, computed from the time interval  $T$  and  $N$  the number of cross nodes from a set of time intervals. If we assume that the contacts occurrences are independents,  $\rho$  is given by:

$$\rho = \frac{n_T}{N} \times 100. \quad (16)$$

We pay an attention to the interpretation of this metric, which can be only considered at the end of the second interval. Indeed,  $\rho$  is always fixed at 100% at the end of the first interval.

b) *Long-term contacts metric*: This metric assesses the average number of long-term meetings with a node. It uses also the EWMA filter to give more importance to recent values than older, avoiding a lag effect on the assessment. Let  $Plg_{u,v}^t$  the average number of long-term meeting computed from a set of time intervals  $T$ , each meeting occurrences between nodes  $u$  and  $v$  during a period  $t$  is denoted  $n_t^u$  such as:

$$Plg_{u,v}^t = \alpha \cdot n_t^u + (1 - \alpha)n_{t-1}^u. \quad (17)$$

To avoid a lag effect,  $\alpha$  must be  $0 < \alpha < 0.5$  in order to give more importance to the newest values than older.

c) *Recent contacts metric*: We are now interested to assess recent contact occurrences. This third metric computes from a Short Time Interval (STI) ( $< T$ ) the average number of short-term meetings. Let  $Pr_{v,u}^t$  the average number of short-term meetings between the couple of nodes  $u$  and  $v$  at time  $t$ . Such as  $n^{u,v}$  is the number of exclusive meeting occurrences during a time interval, so  $Pr_{v,u}^t$  is computed as follows:

$$Pr_{v,u}^t = \frac{n^{u,v}}{n}. \quad (18)$$

d) *Indirect contacts metric*: The last metric assesses the average number of indirect short-term meetings. We assume that if a node encounters often two nodes  $v$  and  $u$  during the same time interval, there is a high probability that this couple of the nodes cross each other during the same time interval. Let  $Pi_{v,w,u}^t$  the average number of indirect short-term meetings computed by the node  $v$  at time  $t$  between  $u$  and  $w$ . This information is computed from the last time intervals, by counting the number of meetings observed by  $v$  with the couple of nodes  $u$  and  $w$  and denoted  $P_{v,w,u}^i = \frac{n_{u,w}}{n}$ .

Therefore, the average number of indirect short-term meetings between  $v$  and  $u$  is computed as follows:

$$Pi_{v,u}^t = \max_{rc} \{Pi_{v,rc,u}^t\}. \quad (19)$$

We consider only the most recently node crossed by  $v$  and denoted  $rc$ .

3) *Queuing mechanism*: Our protocol determines a packet transmission priority in order to reduce the end-to-end delay. A queuing mechanism has been implemented, each queue is dedicated to a destination node. Each  $p$  packet stored in a queue  $q$ , is ranked according to its deadline  $t_d$  which decreases periodically. We use a scheduling algorithm to determine for each one, the number of packets to be transmitted. Our algorithm relies on the Weighted Round Robin (WRR) to compute the transmission priority related to each queue. Let  $q$  a queue with a size  $n$  and  $q_p^{t_d}$  the deadline  $t_d$  of the packet  $p$ , the weight related to the queue,  $W(q)$  is computed following the equation 20.

$$W(q) = q_0^{t_d} \sum_{p=0}^n q_p^{t_d}. \quad (20)$$

The weight determines the priority level of the queue, the one with the lowest weight is the most served by the scheduling algorithm. Our algorithm inspects the weight of each queue in order to determine the number of packets to be transmitted. The algorithm 3 details each step.

---

**Algorithm 3:** Scheduling algorithm

---

**Data:**  $F$ , list of queues  
**Data:**  $C$ , list of queue's weight  
**Data:**  $n_f$ , size of the list  $f$   
**Data:**  $w_{f,p}$ , weight related to the packet  $p$  of the list  $f$

```

1 for  $i \leftarrow 1$  to  $n_F$  do
2    $C(i) \leftarrow 0$ 
3   for  $j \leftarrow 1$ , to  $n_i$  do
4      $C(i) \leftarrow C(i) + w_{i,j}$ 
5   end
6    $C(i) \leftarrow C(i) \times w_{i,0}$ 
7   if  $C(i) > Cmax$  then
8      $Cmax \leftarrow C(i)$ 
9   end
10 end
11 for  $i \leftarrow 1$  to  $n_C$  do
12   /* Determine the number of packets
      transmitted */
13    $TransmitPacket(F_i, \frac{Cmax}{C_i})$ 
14 end
```

---

The packet transmission is then managed by the DTN routing algorithm, which determines the rewards related to the packet transmission or its storing. If the algorithm keeps a packet, all related to the same queue are not served during the meeting time with the neighbor. At last, all packets destined to a connected neighbor are directly sent.

4) *Replication mechanism*: We have developed a replication mechanism based on the packet deadline  $t_d$  in order to reduce the end-to-end delay. We compute the time related to the copy creation according to its deadline. The computation starts as soon as the packet is stored in a queue. Let  $t_c$  the time related to the creation of the copy and computed according to the following equation:

$$t_d = \begin{cases} t + (t_e \times \max(Pr, Pi, Plg)), & Pr \neq 0 \vee Pi \neq 0 \vee Plg \neq 0 \\ t + \frac{t_e}{2}, & else \end{cases}. \quad (21)$$

The basic idea is to increase  $t_c$  if a node knows when it will meet the desired destination. If any knowledge are available about the meeting time with the destination, a copy is created as soon as the half of the deadline has been reached. Finally, if the deadline  $t_d$  is passed the packet lifetime has expired and it is removed from the queue.

#### IV. SIMULATION SETTINGS AND SCENARIOS

In this section, we describe simulation scenarios and detail simulation settings. We used the ns-3 simulator [13] to design networks with realistic signal propagation models and the Bonmotion software [14] to generate mobility scheme retracing the displacement of unmanned vehicles. We define two types of scenarios to assess the performance of our protocol.

##### A. Scenarios to compare hybrid routing protocols

We lead a performance evaluation, by comparing the routing performances of the S-ROGUE and the classical HYMAD routing protocol [7] which has been well investigated in the literature. As HYMAD is not available in ns-3, we set simulation settings and mobility schemes, according to [7]. In the first scenario, 30 nodes move according to the Random Waypoint mobility model (min speed: 0.5m/s, max speed: 1.5m/s, wait time: 2s) during 5000s. In the second scenario, 60 nodes move according to realistic Rollernet trace a highly connected and extremely mobile connectivity traces. Since authors in [7] do not use any propagation models in their simulations, we will use the deterministic Friis propagation model to recreate at least a free space environment.

##### B. Scenarios related to unmanned vehicles

To test our protocol in realistic environments related to unmanned systems, two additional scenarios have been defined. The first one is dedicated to a mission in a free space environment, the second one is dedicated to a mission in an urban city. In the first scenario, all engines evolve on an area of  $1km^2$ , where aerial engines evolve over terrestrial engines at an altitude of 100m. Aerial engines move according to the PPRZM model (PaPaRaZzi Mobility) [15], meanwhile the terrestrial engines move according to the Disaster Mobility Model [16]. In the second scenario, aerial vehicles keep the same mobility, whereas terrestrial vehicles move according to Manhattan Mobility Model with a grid of  $10 \times 10$ . We fix the speed at  $15m.s^{-1}$  for the couple of scenarios. Regarding the propagation model, we fix parameters in order to describe the

TABLE I  
SIGNAL PROPAGATION SETTINGS

Scenario	Pathloss	Shadowing	fading
Free space	Friis (Freq: $2.4 \times 10^9$ , SystemLoss: 1, MinLoss: 0)	None (any buildings)	Nakagami-m (m0: 1.5, m1: 0.75, m2: 0, d1: 80 d2: 320)
Urban	ThreeLogDistance(Exp0: 2.5, Exp1: 5, Exp2: 10, Dist0: 1, Dist1: 75, Dist2: 114)	Building Loss Model (set building at each crossing)	Nakagami-m (m=1) $\rightarrow$ Rayleigh

path loss, shadowing and fading effects as described in table I. Each scenario has been running with three seeds (5155236, 5236, 60) in order to retrace different fast-fading effects.

## V. EVALUATION RESULTS

In this section, some evaluation results are presented. First, the ability of the S-ROGUE and HYMAD protocols to support the network load is analyzed. Second, the robustness of the S-ROGUE is compared to classical MANET and DTN protocols.

### A. Network load analysis

We compare the performance of S-ROGUE and HYMAD by observing the related Packet Delivery Ratio (PDR) according to network load. Figures 7a, 7b and 7c show results with the Random Waypoint mobility model and different data traffics. Figure 7d shows results related to the Rollernet scenario. The S-ROGUE protocol appears as the best regardless the scenario. At the end of the first three scenarios, the PDR is the same with the S-ROGUE protocol, close to 0.9 unlike HYMAD, where the PDR decreases when the network load increases. In the last scenario with a real mobility trace, we confirm that S-ROGUE has a better PDR than HYMAD.

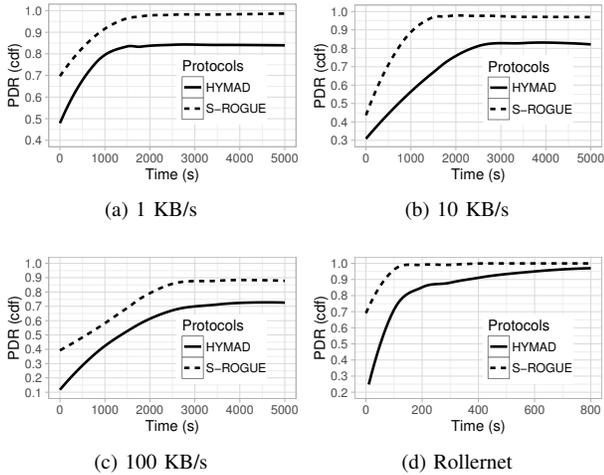


Fig. 7. Cumulative PDR with S-ROGUE and HYMAD

### B. Robustness analysis in realistic simulated environments

We compare the S-ROGUE with classical MANET protocols (AODV and OLSR) and DTN protocols (Spray and Wait with 3 copies authorized and an unlimited version) in term of PDR and end-to-end delay. Figure 8 plots the mean and the

standard error of the mean related to the PDR and the end-to-end delay. Regarding the PDR, S-ROGUE provides the best ratio in both scenarios. Even if the number of nodes impact the network density, giving an advantage to MANET protocols, the environment has a significant impact on the signal quality. As a result, even if the number of paths decreases in the urban scenario, S-ROGUE keeps a high PDR regardless the scenario. About the end-to-end delay, MANET protocols have the lowest delay compared to DTN protocols. However, S-ROGUE comes to reducing the end-to-end delay by exploiting current paths.

## VI. CONCLUSION

In this paper, we have presented a novel routing protocol (S-ROGUE) designed for unmanned systems and using either a MANET or a DTN algorithm according to the network connectivity. The MANET algorithm relies on a proactive approach to discover the local topology and end-to-end paths. It uses also the F-ETX metric which assesses both the quality and the state of local links. We have developed a framework to integrate each estimator in the routing protocol such as, each information is taken into account by the routing process to select the best link. The DTN algorithm relies on a RL technique to perform the best routing action when nodes are crossing. Indeed, it determines if a packet has to be transmitted or kept according to the remaining energy level of a node and its capacity to meet the desired destination. The DTN algorithm uses also a queuing mechanism and a replication technique to reduce the end-to-end delay. S-ROGUE has been compared to a large panel of solutions including MANET, DTN and Hybrid protocols in realistic environments. We have investigated the performance of the S-ROGUE by changing the network load and the number of nodes. Regardless the scenarios, S-ROGUE has appeared as the best routing protocol in term of PDR and end-to-end delay.

However, some issues are still open. Firstly, we did not perform an investigation to identify the bottleneck of our protocol. Secondly, we will investigate online learning algorithms to determine if a better approach is possible.

## REFERENCES

- [1] NIST, *Autonomy Levels for Unmanned Systems (ALFUS) Framework, Volume I: Terminology*, NIST Std., Oct 2008.
- [2] L. Freitag, M. Grund, S. Singh, J. Partan, P. Koski, and K. Ball, "The whoi micro-modem: an acoustic communications and navigation system for multiple platforms," in *Proceedings of OCEANS MTS/IEEE*, Sept 2005, pp. 1086–1092 Vol. 2.
- [3] A. Finn and S. Scheding, *Developments and Challenges for Autonomous Unmanned Vehicles - A Compendium*, ser. Intelligent Systems Reference Library. Springer, 2010, vol. 3.

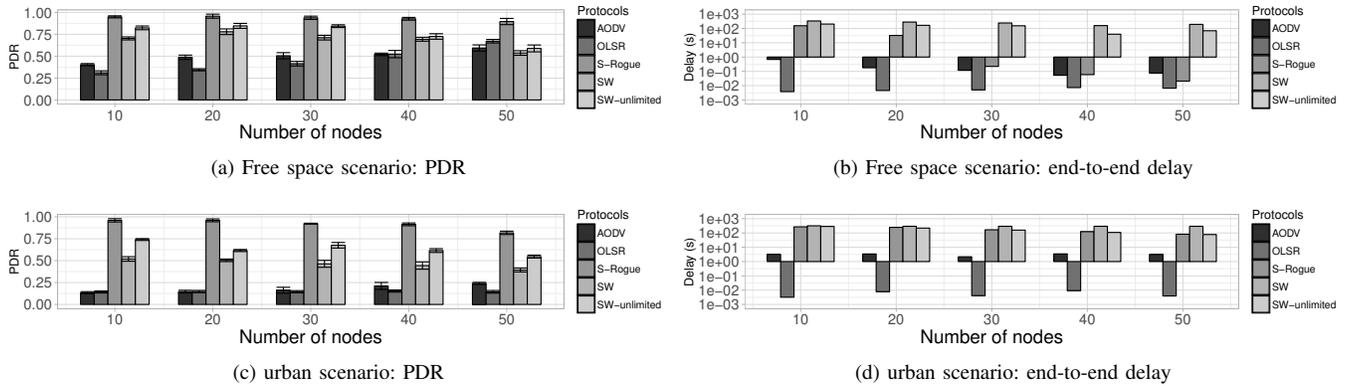


Fig. 8. Impact of the number of nodes on routing performance in different environments

- [4] L. Delosieres and S. Nadjm-Tehrani, "Batman store-and-forward: The best of the two worlds," in *Pervasive Computing and Communications Workshops, IEEE International Conference on*, 2012, pp. 721–727.
- [5] C. Raffelsberger and H. Hellwagner, "A hybrid MANET-DTN routing scheme for emergency response scenarios," in *IEEE PerCom Workshops '13*, J. Indulska and C. Bisdikian, Eds., Mar 2013, pp. 505–510.
- [6] C. Kretschmer, S. Ruhrop, and C. Schindelbauer, "Dt-dymo: Delay-tolerant dynamic manet on-demand routing," in *IEEE ICDCS Workshops '09*, June 2009, pp. 493–498.
- [7] J. Whitbeck and V. Conan, "Hymad: Hybrid dtn-manet routing for dense and highly dynamic wireless networks," *Computer Communications*, vol. 33, no. 13, pp. 1483 – 1492, 2010.
- [8] S. R. Azzuhri, H. Ahmad, M. Portmann, I. Ahmedy, and R. Pathak, "An efficient hybrid manet-dtn routing scheme for olsr," *Wireless Personal Communications*, vol. 89, no. 4, pp. 1335–1354, 2016.
- [9] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*, 1st ed. Cambridge, MA, USA: MIT Press, 1998.
- [10] S. Bindel, S. Chaumette, and B. Hilt, "F-ETX: a predictive link state estimator for mobile networks," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 2, no. 7, Jun. 2016.
- [11] N. Baccour, A. Koubâa, H. Youssef, and M. Alves, "Reliable link quality estimation in low-power wireless networks and its impact on tree-routing," *Ad Hoc Netw.*, vol. 27, no. C, pp. 1–25, Apr. 2015.
- [12] X. F. Guo and M. C. Chan, "Plankton: An efficient dtn routing algorithm," in *2013 IEEE International Conference on Sensing, Communications and Networking (SECON)*, June 2013, pp. 550–558.
- [13] G. F. Riley and T. R. Henderson, *The ns-3 Network Simulator*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 15–34.
- [14] N. Aschenbruck, R. Ernst, E. Gerhards-Padilla, and M. Schwamborn, "Bonnmotion: A mobility scenario generation and analysis tool," in *Proceedings of SIMUTools '10*. Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, 2010, pp. 51:1–51:10.
- [15] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A Mobility Model For UAV Ad hoc Network," in *ICUAS 2014*, Orlando, United States, May 2014, pp. pp 383–388.
- [16] N. Aschenbruck, E. Gerhards-Padilla, M. Gerharz, M. Frank, and P. Martini, "Modelling mobility in disaster area scenarios," in *Proceedings of MSWiM*. New York, NY, USA: ACM, 2007, pp. 4–12.