Efficient Scheduling for Planned Robot Networks

Farid Benbadis, Hicham Khalifé, Catherine Lamy-Bergot
Thales Communications & Security
Gennevilliers - France
{name.surname}@thalesgroup.com

***Abstract*—We propose in this paper a scheduling solution for networks of autonomous robots that exploits the store-and-forward capabilities of nodes during a patrolling mission. Our solution leverages the deterministic behavior of these mobile vehicles in order to construct a connectivity graph for the whole mission that will be exploited to schedule communication requests upon their arrival. Then our scheduling algorithm, based on the capacity demand and the priority level of each demand, is able to place an arriving request in a way to guarantee its delivery deadline. A key feature of our solution stems from its capacity to preempt lower priority ongoing communications in order to create enough space for newly arriving ones with higher priority. Through simulations, we show that by using our solution, we can increase the global conveyed traffic by up to 40% and place up to 10% more high priority traffic.**

I. Introduction

The emergence of new types of automated devices empowered with multiple radio technologies leads to rethinking many communication and networking paradigms. In particular, autonomous robots are arising as new equipment filled with cutting edge technologies and capable to communicate over multiple radio bands and standards. In this category, one can cite autonomous cars, drones or any ground robot as widespread or coming solutions that will disrupt communication procedures known to date. Still, only few initiatives have investigated so far how to take advantage from the numerous new features these devices can embed. Typically, the cost gains that can be obtained from efficiently scheduling communications and movements for a fleet of such robots having to sweep an area are rarely considered. Obviously, this can be of great interest for the operators, in terms of autonomy, communication time, and cost of operation.

These new autonomous agents can either be fully self-governing taking decision and reacting to observed changes or programmed offline to follow a set of tasks or movements in a decided order. The latter case, already largely investigated, for automated public transportation initiatives in smart cities [1] reduces the known risks of fully automated vehicles by operating at dedicated lanes and following pre-planned trajectories. Consequently, exploiting such pre-established path for communication allows to optimize connectivity and throughput, leveraging advanced store-and-forward and scheduling techniques such as Delay Tolerant Networking (DTN) approaches [2]. In practice, using the *a priory* known position of nodes (e.g. through GPS coordinates) to schedule communication allows to better adapt the demand to current and *future* links and network status.

Several research initiatives have already explored robots and autonomous agents path planning as well as collision free navigation solutions [3]. Interestingly, Portugal et al. [4] have demonstrated that cyclic-based patrolling strategies offer the best solutions especially when small number of robots are implicated whereas partitioning strategies are more suitable for larger teams and unbalanced topologies. Particular attention was drawn on defining patrolling strategies that provide efficient/optimal trajectories and dynamic adaptation to steer robots toward an optimal path [5]. Recent proposal have also considered communication constraints in the deployment planning: for instance, Acevedo et al. [6] formalize the perimeter surveillance problem under communications constraints, and the authors solve the defined problem by proposing a path-partition strategy using a frequency-based approach. Mahboubi et al. [7] present an algorithm capable to increase coverage in mobile wireless sensor networks by proposing a strategy that moves mobile nodes in a way to reduce coverage holes. Other initiatives have also investigated the persistent coverage problem in changing environment [8] [9].

Unlike these previously cited initiatives, in this paper we tackle the problem from a different perspective: we consider the patrolling and path planning as already decided, and explore the multiple radio capabilities available in the robots to satisfy a set of communication constraints. This hypothesis on the paths being pre-determined is consistent with use in areas where complete sweeps are required (e.g. disaster areas) and so where a random walk or reactive trajectory determination are not usable approaches.

Our solution leverages store-and-forward concepts in order to offer additional capacity and throughput in this pre-planned robot network. The key idea resides in basing our scheduling decision on the full mission contact graph, extracted as a time-discretized graph that highlights buffering opportunities. Then, over the hence obtained graph, we propose, as our main contribution, a scheduling algorithm that exploits buffering options to propose DTN-ready scheduling. Our approach not only reduces the costs in a way to favor store-and-forward options known to be less costly but also enables moving to other existing paths ongoing communications to increase the number of accepted demands and capacity. This is ensured while guaranteeing the delivery deadline of each communication. Our solution, with its specific features such as flow interruption and resources re-allocation, is compatible with

This work was partially funded by DGA (Direction Générale de l'Armement) under project PEA ICARO.

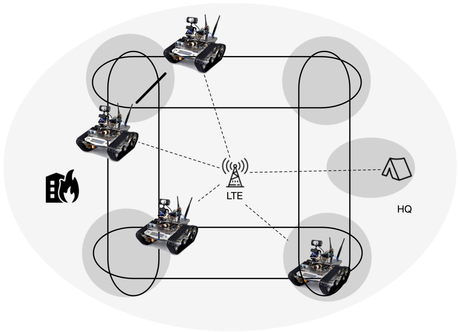


Fig. 1: Investigated use case implying 4 robots sweeping a surveillance zone with pre-planned contact/communication opportunities.

DTN solutions [2] [10], but not part of the paradigm itself.

The remainder of the paper is structured as follow: in Section II, we define a realistic use case for employing autonomous robots and formulate the investigated problem. We then detail our contribution in Section III composed of a contact graph construction and a scheduling algorithm design before validating our solution through simulation in Section IV. Finally, Section V concludes the paper and gives future research perspectives.

II. Case Study

We investigate in this part a representative use case implicating a set of collaborating ground robots deployed to monitor a sensitive zone or a part of a disaster relief operation. To reduce the cost of such operations, and avoid the risk of collisions between platforms, limiting the number of robots is deemed necessary. For this reason, robots in their patrolling mission do not cover the whole operation region but are allocated to a pre-defined zone they continuously monitor. Our robots are equipped with 2 radio interfaces, LTE and device-to-device (D2D), and they are empowered by a GPS receiver. In the cost metric that we will use, D2D connection will have a lower cost than LTE, which relies on an infrastructure, requires a subscription, and needs more transmission power.

A. Scenario Description

In the scenario depicted in Fig. 1, 4 autonomous robots follow an elliptic trajectory in order to monitor the hence covered zone. The D2D interface (WiFi-Direct or ZigBee could be considered) permits direct communication whenever 2 robots meet (refer to the grey zones in Fig. 1). These cheap connections prevent from using expensive LTE communications and can be used in the case of a continuous patrolling and in the absence of important events. More precisely, each of these four robots can be in the D2D neighborhood of two other robots but never at the same time. Also, only one of the four robots enters into direct communication (non-LTE) with the headquarter. Quite logically, such deployment becomes interesting when the autonomous devices meet, create communication opportunities in the dedicated areas where robots are in communication range of each other, and relay messages for each other, by storing then forwarding them.

In this context, because initial position of robots (thanks to their GPS receivers) and their speed are known, one can establish in advance the contact graph for the whole mission duration. For the studied deployment, robots continuously monitor the environment and record/buffer a set of defined operational data. Depending on the assigned mission, such data include registered pictures of particular spots, measured phenomena such as temperature, humidity, and short recorded videos of a part or the whole round of surveillance. Knowing that the sensed data has a validity duration, it should be conveyed to the control center before it becomes obsolete. Typically, sensors information need to be received by the headquarters regularly, for them to be analyzed and action be initiated if needed. Most often, a recorded picture or measured temperature needs to be received before the robot captures and transmits a newer copy of the same information. In some cases, the robot may detect some abnormal condition (e.g. a temperature outside of a predefined range) and assign then a higher priority and lower time to distribute the information.

B. Problem Formulation

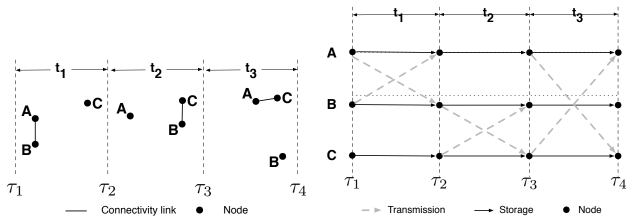
Given a source robot generating different types of content (observed or sensed data) having each a capacity requirement and a validity duration, how to schedule this traffic in a way to satisfy its capacity and delay requirements. From a network perspective such allocation should be made while in the meantime maximizing the network capacity and reducing the link usage costs. In other words, the question at hand can be formulated as follows: should we use the LTE connection or relay messages using multihop device-to-device links to reach the headquarters (or other robots) at the minimum costs? This decision should be made not only based on the considered traffic requirement but also based on costs and global network capacity metrics, and the strategy may differ when priority of the data is high or low.

III. Proposed Solution

A. Time-Discritized Graph

We use TDG (time-discretized graph) [11] to represent the dynamicity of our topology over time. In a TDG, a path between two vertices represents a possible space-time route between two nodes. A time-discretized graph is similar in essence to time-independent graphs proposed by Hay et al [12]. In both approaches, edges may be associated with different metrics to incorporate delay, storage, and transmission constraints. Nevertheless, while time-independent graphs require the pruning of some contacts between nodes, TDG does not and keeps a complete representation of the nodes and their contacts.

Let us now detail, through a simple example, how any contact patterns of a wireless network can be transformed into a TDG. Consider a wireless network composed of three nodes A, B, and C, as illustrated in Fig. 2a which shows the evolution of contacts between all pairs of nodes. In this figure,



(a) Example of topology evolution. (b) Resulting contact graph.

Fig. 2: An example of topology evolution and its corresponding graph representation. Note that there are two different potential routes between A and C, namely $A_{\tau_1} \to A_{\tau_2} \to A_{\tau_3} \to C_{\tau_4}$ and $A_{\tau_1} \to B_{\tau_2} \to C_{\tau_3}$.

nodes are represented by black dots and a solid line is drawn between two nodes when they are under radio coverage of one another and can communicate. We assume that contacts offer bidirectional connectivity. In the depicted example, A and B are in contact during t_1, B and C are in contact during t_2, and A and C are in contact during t_3. Based on this configuration, if A needs to transfer a message to C, it has the option to send it through B or wait until it meets C. The choice is based on the result of the forwarding strategy which considers the communication requirements (*e.g.* bandwidth, delay, latency), the available capacity between the source and the destination, and the cost of the communication which depends on the chosen radio technology.

To turn contact patterns between nodes from wireless connectivity into a TDG, we first discretize time into intervals of varying durations. Each time interval represents a period of time during which the topology of the network remains unchanged. The idea is to represent each node at each time step with a vertice, and that an edge between two vertices represents either a node storing the data or a forwarding opportunity between two nodes.

The resulting time-discretized graph $G = (V, E)$ represents all available routes univoquely – any path in the graph represents a feasible wireless route, and, conversely, all feasible wireless routes can be represented as paths in the graph.

Two types of links are used in Fig. 2b, each of them associated with a step in the communication process:

- **Forwarding step.** Gray dashed arrow represents contact between two nodes, i.e. transmission opportunity.
- **Storage step.** Black solid arrow corresponds to the case where a node stores data for the time interval.

B. Scheduling Algorithm

Once the time-discretized graph of the complete mission established at the beginning of the operation, a specifically designed algorithm is instantiated at every arriving communication request. This algorithm constitutes the key contribution of our work by scheduling arriving demands while leveraging D2D capabilities of the deployed robots. Our strategy, in addition to exploiting the future contacts from the TDG to handle traffic requests based on their delivery deadline, offers the possibility of manipulating lower priority demands by either delaying them in time or deleting them. This is done to accept those of a higher priority. As a consequence, such operation increases greatly the total capacity of the network while reducing costs.

Algorithm 1 Schedule new arriving flow demand D

Input: G // Weighted network connectivity graph
$X_D^{T,P}$ // Arriving demand for communication X with delivery deadline T_X and priority P_X
Output: G // Updated network Graph
τ_X // Estimated delivery time of demand $X_D^{T,P}$

1: τ_X = min_cost_dtn($G, X_D^{T,P}$)
2: **if** $\tau_X > T_X$ **then**
3:   sort(P_i) // increasing order
4:   **for** i **do**
5:     Remove(R_i) // Remaining capacity of demand i
6:     **if** $X_D^{T,P}$ and R_i can be guaranteed **then**
7:       min_cost_dtn($G, X_D^{T,P}$)
8:       min_cost_dtn($G, i_D^{T,P}$)
9:       **return** G, τ_X, τ_i
10:     **else**
11:       **if** (only $X_D^{T,P}$ can be guaranteed && removable_selected == false && $P_i < P_X$) **then**
12:         removable = i
13:         removable_selected = true
14:       **end if**
15:     **end if**
16:   **end for**
17:   **if** removable_selected **then**
18:     Remove(removable)
19:     τ_X = min_cost_dtn($G, X_D^{T,P}$)
20:     **return** G, τ_X
21:   **else**
22:     FAIL()
23:     **return** G
24:   **end if**
25: **end if**

More formally, every communication X is designated by its capacity demand D_X, its deadline T_X and its priority level P_X. Consequently, demand X is noted $X_D^{T,P}$.

First over the previously established graph G, our algorithm searches for an available path that minimizes the cost for the arriving demand (min_cost_dtn($G, X_D^{T,P}$). Our algorithm returns the estimated delivery time τ_X and if this deadline is lower than T_X, the demand deadline, it exits successfully. Otherwise, we go through all already scheduled communications from the lowest priority to the highest, looking for a case where both demands can be satisfied by delaying an older demand with lower priority while ensuring that the two deadlines are guaranteed (lines 7 to 9 of Algorithm 1). This is possible if the low priority requests we are moving have been placed so that they end much sooner than the deadline requested by the system. Thus, moving them to a later delivery allows accepting other requests while respecting the deadline of both demands.

If moving already scheduled traffic does not allow to accept $X_D^{T,P}$, the second option our algorithm implements aims at replacing a communication of a lower priority by the arriving one. However, to optimize the complexity of our algorithm this operation is cleverly done during the previous same iteration. To do so, during the same loop we flag communications of lower priority that, when removed, would allow enough room for $X_D^{T,P}$ (lines 11 to 13 of Algorithm 1). Consequently, if at the end of the iteration the *removable_selected* flag is true for a traffic, then the indicated demand is completely removed and $X_D^{T,P}$ accepted (lines 17 to 20).

Finally, if no other communication can be moved or a lower priority replaced, our algorithm fails to schedule the arriving requests and the $X_D^{T,P}$ is irrevocably dropped.

IV. Validation

The presented algorithm has been applied to the use case presented in Fig. 1, with simulation environment and parameters detailed hereafter.

A. Simulation Environment Description

The four nodes were simulated moving on an elliptical loop. The average movement speed of each node is equivalent to 10 m/s and the total distance of each ellipse is 3 km. Each of the four nodes meets two other nodes at the intersection of the ellipses of their respective trajectories. The coverage radius of the used D2D radio is set to $100m$, which gives a duration of contact between the nodes of 3 min at each encounter. During its movement along its ellipse, one of the four nodes crosses the headquarters and can communicate directly with it.

All the nodes involved in this scenario can also communicate via LTE. We assume that connectivity to the eNodeB is always available for all the nodes, regardless of their position. In our evaluation, we varied the D2D throughput relatively to LTE's throughput. The evaluated values vary from 4 to 14 in steps of 2 (4 means that LTE is four times faster than D2D).

We generate a number of communications that we try to place in our network, using our scheduling algorithm, while respecting their constraints. Each communication, generated randomly, is composed of a source, a destination, an instant of arrival, a delivery deadline (time elapsed between instant of arrival and deadline is the delay of conveying data), a size, and a priority level. This priority level is used by our placement algorithm described in Sec III-B, in order to remove low priority demands to make room for high priority ones. A communication is considered placed if one is able to find enough resources to respect its size demand between the source at its time of arrival and the destination no later than the deadline.

B. Simulation Results

For all of our simulations, we have generated randomly 200 communication requests that we aim to schedule with our algorithm. As already explained, communications can take place between any pair of nodes, including robots and headquarter. The capacity of the simulated LTE is a factor of D2D capacity and ranges from 4 to 14 in steps of 2. Simulated LTE usage costs are 10 and 100 times higher than D2D. Our communication scheduling algorithm has been performed in two scenarios: one where nodes have store-and-forward (*S&F*) capabilities, and one where they have not, i.e. requiring an end-to-end connection to communicate (*No S&F*).

1) Satisfied requests and their total cost: The generated requests are placed one after another, depending on their arrival time. Fig. 3 shows, for a capacity factor of 4 and a cost factor of 10 (which means that LTE offers 4 times more capacity for a cost 10 times higher), if a given demand has been placed or not. If a demand has been placed, then it is represented with a vertical bar. The bars show also the cost for satisfied requests. The x axis depicts the id of the request, and the y axis gives the cost. Each request may be represented with two colors; blue if the algorithm has succeeded to place it using store-and-forward capability, and orange if it has been able to place it without store-and-forward capability. For instance, we can observe that requests $13, 14, 15$, and 16 could be placed only if S&F is used, while requests 17 and 18 have not been placed regardless of the used technique.

Another observation is that the number of blue satisfied requests (the ones with store-and-forward) largely dominates the number of the orange ones (non S&F). This result is logical and expected, since by using the S&F feature, we are adding capacity to the network. Moreover, a feature to highlight concerns the cost of placed communications. While the size of blue bars remains stable, the size of the orange ones increases steadily. This phenomenon is due to the fact that the use of store-and-forward option reduces the use of LTE and makes full use of regular encounters between robots and their ability to forward messages to each others. In the case of the network not using store-and-forward, resources are quickly exhausted and the use of LTE is more frequent and necessary, which logically increases the cost of communications.

Besides, one may consider surprising at first glance that few requests, less than 5%, are satisfied in network not using store-and-forward and not satisfied in the network implementing this strategy. Such behavior may seem counter-intuitive, but it is easily explicable. This phenomenon occurs only after several requests are satisfied in S&F scenario while not satisfied in the other scenario. Under such conditions, more resources were consumed in the S&F network than in the non S&F one. In such a situation, a new request can find resources in the second scenario but not in the first.

2) Cost of placed requests: In Fig. 4, we plot the cost/capacity ratio of all the communications that have been successfully placed. This metric clearly reflects another benefit of using the store-and-forward mechanism. Indeed, it allows to investigate the effective price of the capacity added to the network. Interestingly, in addition to allowing more requests, the observed cost is lower than the non store-and-forward case. In other words, our algorithm allows for *cheap capacity*

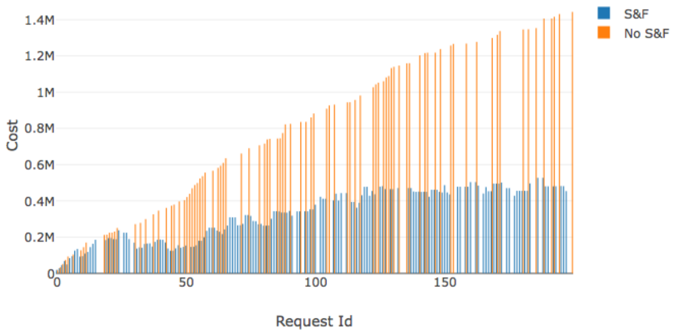


Fig. 3: Placement costs with and without DTN.

increase. This behavior remains valid regardless of the cost and capacity factors. An important result here is that capacity factor has no impact on the cost/capacity ratio in the S&F scenario. This is due to the extra bandwidth gained thanks to S&F strategy, which saves us from consuming expensive resources.

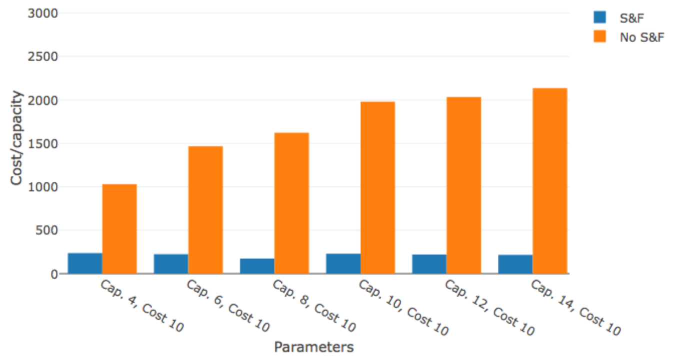


Fig. 4: Ratio cost/capacity of all the requests.

3) Effect of preemption mechanism: Fig. 5 shows two pie charts summarizing the outcome of our scheduling algorithm, taking into consideration the priority of the demands. The red part represents the rate of requests that could not be satisfied, regardless of their priority. The green part shows the high priority placed requests percentage, while the orange indicates the low priority requests rate that have been scheduled. The left chart depicts the results based on our algorithm applying the preemption method that favors high priority requests. The right chart does not consider any priority when placing demands. That is, if an arriving communication can not be placed, it will be discarded, whatever its priority.

Fig. 5 highlights that allowing preemption reduces the number of rejected connections. This increase in the number of accepted demands benefits the high priority traffic (11%). Logically, due to the intrinsic design of our algorithm, higher priority traffic is favored by the scheduling algorithm by preempting lower priority demands. Still, rejected communications can also be high priority requests that were not satisfied due to their size and high claim.

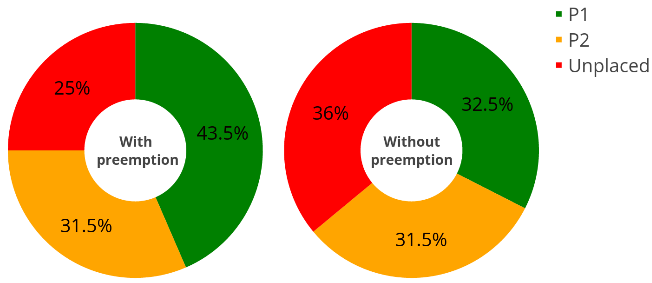


Fig. 5: Rate of accepted high and low priority demands, as well as unaccepted demands.

V. Conclusion

In this paper, we have proposed a novel scheduling algorithm that based on capacity, cost and delivery deadline of arriving demands efficiently exploits store-and-forward capabilities of planned robots networks. Our solution creates a time-discretized graph of the whole patrolling mission, then schedules communications in a way lower priority demands are moved or replaced to leave room for those of a higher importance. Our simulations show 40% more transported traffic thanks to the store-and-forward features, while dividing the cost/capacity ratio by up to 4.

In the future, we plan to investigate cases where deviations from the planned trajectory of the robots occur. In such situations, techniques coming from artificial intelligence could be used to help correct the estimated path.

References

[1] The city of helsinki self-driving bus. https://www.hel.fi/uutiset/en/helsinki/helsinki-self-driving-bus-regular-service.

[2] Kevin Fall. A delay-tolerant network architecture for challenged internets. In *ACM SIGCOMM conference*, 2003.

[3] M. Hoy, A. S. Matveev, and A. V. Savkin. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Cambridge University Press*, 33(3), 2015.

[4] D. Portugal, C. Pippin, R. P. Rocha, and H. Christensen. Finding optimal routes for multi-robot patrolling in generic graphs. In *IEEE Intelligent Robots and Systems (IROS) Proc.*, 2014.

[5] F. Pasqualetti, A. Franchi, and F. Bullo. On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms. *IEEE transactions on Robotics*, 28(3), 2012.

[6] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero. Cooperative perimeter surveillance with a team of mobile robots under communication constraints. In *IEEE Intelligent Robots and Systems (IROS) Proc.*, 2013.

[7] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh. Distributed deployment algorithms for improved coverage in a network of wireless mobile sensors. *IEEE transactions on Industrial Informatics*, 10(1), 2014.

[8] J. M. Palacios-Gasos, E. Montijano, C. Sags, and S. Llorente. Distributed coverage estimation and control for multirobot persistent tasks. *IEEE transactions on Robotics*, 32(6), 2016.

[9] J. M. Palacios-Gasos, Z. Talebpour, E. Montijano, C. Sags, and S. Llorente. Optimal path planning and coverage control for multi-robot persistent coverage in environments with obstacles. In *IEEE Robotics and Automation (ICRA) Proc.*, 2017.

[10] Delay tolerant networking architecture. *IRTF RFC 4838 IETF*, 2007.

[11] P. Tournoux, V. Conan, J. Crowcroft, J. Leguay, M. D. de Amorim, and F. Benbadis. Wardrop equilibrium formulation of resource-constrained DTN routing in public safety networks. In *IEEE MASS Proc.*, 2011.

[12] D. Hay and P. Giaccone. Optimal routing and scheduling for deterministic delay tolerant networks. In *IEEE WONS Proceedings*, 2009.