



HAL
open science

A Lightweight Fluid Model for Mobile Ad hoc Distributed Systems

Bruno Chianca Ferreira, Guillaume Dufour, Guthemberg Silvestre

► **To cite this version:**

Bruno Chianca Ferreira, Guillaume Dufour, Guthemberg Silvestre. A Lightweight Fluid Model for Mobile Ad hoc Distributed Systems. ISCC2022, Jun 2022, Rhodes, Greece. 10.1109/ISCC55528.2022.9912980 . hal-03668033

HAL Id: hal-03668033

<https://hal.science/hal-03668033>

Submitted on 14 May 2022

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.

A Lightweight Fluid Model for Mobile Ad hoc Distributed Systems

Bruno Chianca Ferreira¹, Guillaume Dufour², and Guthemberg Silvestre³

¹ ENAC/ReSCo, Université de Toulouse, France
`bruno.chianca@enac.fr`

² ONERA/DTIS, Université de Toulouse, France
`guillaume.dufour@onera.fr`

³ ENAC/ReSCo, Université de Toulouse, France
`silvestre@enac.fr`

Abstract. Emerging edge applications introduced new computing time-variant topologies with mobile nodes connected via ad hoc networks. Such topologies are fundamentally different from cloud infrastructures due to the lack of hierarchy and clear network function separation. Sometimes, nodes that are sources are also destinations and routers, hence, creating dynamic flow patterns traversing the network. The latter, thus, can change the average performance of a distributed system, such as throughput and latency. This work introduces an analytical model based on fluid quantities to study data flows of distributed systems in mobile ad hoc networks. Using an approach based on a network of queues with evenly distributed bandwidth over concurrent flows, this lightweight model enables fast, coarse-grained analysis of different distributed systems configurations. They enable the analysis of different topologies, mobility and data flow models with a small footprint. The model was implemented, validated and evaluated with stress workloads to confirm its accuracy.

Keywords: distributed systems, edge, swarm, modelling, mobile, ad hoc, cps

1 Introduction

A Distributed Computing System is a set of individual networked nodes that collaborate to execute a task or execute individual tasks serving a common purpose. Such computing nodes can be static, as in traditional cloud infrastructures [1] or HPC centres, or they can otherwise be mobile [2], which is a configuration more common in Cyber-Physical Systems (CPS), Internet of the Things (IoT) and Mobile Edge Computing (MEC). Mobile nodes add new challenges to the study and design of distributed computing due to its dynamic nature and unreliable network links. Such dynamic behaviour results in continuously-changing routes within the network, rendering the network traffic unpredictable under certain circumstances. This study introduces a new lightweight, analytical model to

investigate the data flow in such mobile architectures, and how certain topologies may hinder the overall system performance. To scale, our approach models message passing as fluid quantities and the topology as a network of connected queues.

1.1 Mobile Computing Challenges and Opportunities

Mobile Computing architectures can be exploited by different types of paradigms and applications. Although the concept is not new [3], the challenges have evolved over time with the adoption of new communication technologies and advancements of the computing capacity [4, 5]. Furthermore, as applications become more data-centric (e.g., edge applications that require processing large amounts of sensor data), the amount of data required to be transferred increases. So, the focus of the challenges shifts to how to make data available to applications whilst maintaining the ability to comply with the performance goals [1, 6]. For latency-sensitive applications, the most common paradigm adopted are the *Edge* and *Fog computing*, where data is preprocessed at the edge servers before being sent to the cloud. In addition to enabling lower latencies by preprocessing data, edge servers can also be leveraged to increase data safety. For instance, a CPS swarm connected via ad hoc networks can be sent on a mission to survey and collect data that must be stored on a distributed store during the mission before offloading to an edge cloudlet [7]. In other applications, collected data must be distributed among all nodes for sensor data fusion [8]. Such swarms can assume different topologies depending on how the mission was planned and evolves, creating challenges and opportunities to tame the network flow, and thus reduce congestion and delay.

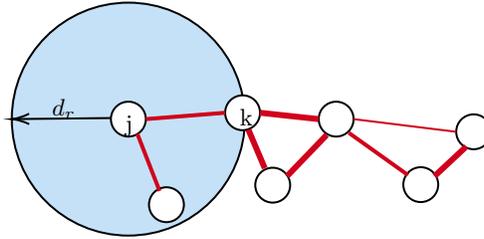


Fig. 1: Topology model representation based in distance between nodes.

Concerning the communication infrastructure, the main characteristics of mobile swarms is how the mobility and network density affect their connectivity [9, 10]. Since the functions of such nodes during their runtime are usually complementary, data flows can be leaving, arriving and traversing the node simultaneously. Due to the limited communication range, as shown in Figure 1 and comprehensively surveyed in [11], communication and routing usually happen in a multi-hop manner. Hence, some nodes, which have a strategic location

in the topology, might experience excessive traversing traffic flow. It is therefore valuable the ability to study the behaviour of the network flows for different topologies and applications. This can be used to prevent unwanted congestions by employing different deployment strategies or employ control laws on the mobility to balance the traffic flow across the topology.

1.2 Traffic Flow in Mobile Distributed Applications

As mentioned in [12] and [13], one of the first steps of modelling a swarm of CPS is to define the deployment topology or self-organizing models. It is, therefore, crucial to have the ability to search several different topologies and path planning alternatives for better traffic flow, which requires a lightweight model for the network employed as a faster than real-time simulator. Mobile nodes have the intrinsic advantage of being able to move during runtime and thus rearrange the topology. By being able to quickly re-evaluate the traffic flow during a mission, the nodes can be potentially rearranged in real-time to improve the performance as done in [14] for connectivity maintenance. Topology control is a popular topic in WSNs [15, 16], and is used as a way to increase network coverage or reduce energy consumption. While discrete network models provide a granularity level, that is usually not required for traffic flow optimization. Hence, models based on fluid mechanics have been proposed as a scalable alternative [17–20]. By assuming a macroscopic point of view of the network flow, such fluid models do not keep granular information about each message. Therefore, they can simulate larger topologies and message flows faster, enabling fast mission planning and opening the possibility of its reevaluation during runtime.

1.3 Contributions

The main contribution of this work is the proposed lightweight, analytical model for distributed computing in mobile ad hoc networks. Such a model can be used to study swarm robotics deployments aimed at simulating network flows and finding events that can hinder the performance of mobile distributed systems. Due to its modularity, the model can potentially be extended to study routing or MAC protocols.

1.4 Outline

In the following Section 2, the paper defines the proposed analytical fluid model, and in the subsequent Section 3 it defines, based in existing state-of-the-art approaches, a detailed system model that allows applications to be modelled and tested. The proposed model's accuracy is later validated in Section 4 in two ways:

- Modelling a data flow model and implementing on a network simulator (OM-Net++), a mobile ad hoc computing emulator (MACE) and in the proposed analytical model.

- Modelling the steady state performance of a bottleneck node and comparing the results with the fluid model proposed in [18].

The model is then used to verify the ability to extract useful metrics for studying distributed systems such as throughput and latency, by comparing stress loads on different topologies. Later, it is used to investigate the behaviour, during the whole simulation runtime, of factors that might hinder the performance of a specific topology, by exposing transient queue levels and average latency in bottleneck nodes. Finally, the scalability of the model is assessed. The results are presented in Section 5 and Table 1 lists most of the notations used in the paper to aid understanding.

Table 1: Notations

Symbol	Description
δt	Time-step
n	Number of nodes in the topology
d_{jk}	Distance between nodes j and k
d_r	Radio communication range
\mathcal{R}	Routing table
S_j	Set of nodes serving data flows for a particular node j
$O\mathcal{R}_j$	Set of nodes being served as next hop for a particular node j
t_{mp}	Medium propagation latency
t_s	Serialization latency
t_{od}	End-to-end latency
\mathcal{P}	Position of a node
\mathcal{A}	Attraction position of a node
δ_p	Position variation for a node
\mathcal{V}	Velocity of a node
\mathcal{L}_a	Limiting area for a mobility model
λ	Flow creation frequency
α	Age of a message
p	Position of a message
Q_j^k	Queue at $p = j$ for messages with destination $e = k$
\mathcal{D}_j^k	Drainage of queue at $p = j$ for messages with destination $e = k$
a_j^k	Arrival of queue at $p = j$ for messages with destination $e = k$
f	Average flow size
ρ	System load

2 Analytical Fluid Model

In this section a fluid model which allows modelling with accuracy the message passing in the network is introduced. The model can be implemented together with the models from Section 3 to build a simulator. In this model, each node in the network is abstracted as a queue, and messages are abstracted as fluid quantities flowing through them.

2.1 Message Queues

In order to model the flow through the topology, each transferred message is abstracted as $\mathcal{M}(a(t), p(t), e)$. Each abstracted message has an accumulated

travel delay, henceforth called age $\alpha(t)$ with $\alpha \in \mathbb{R}^+$, which will be detailed in Section 2.2; a position $p(t)$ with $p \in \mathbb{N}, 0 \leq p < n$, that represents in which node the message is currently located; and the end node e with $e \in \mathbb{N}, 0 \leq e < n$. When a message is flowing from one node to another, they are placed at an output queue as a corresponding amount of data. In each node, the queue evolves during execution and data leaves the queue at a rate according to the available bandwidth connecting the two nodes. When data is located at more than one non-empty output queue, they share the available bandwidth. Each queue \mathcal{Q} is a set with $|\mathcal{Q}| = n$, and each element represents a destination node $\mathcal{Q}_j = \{\mathcal{Q}_j^0, \mathcal{Q}_j^1, \mathcal{Q}_j^2, \dots, \mathcal{Q}_j^{n-1}\}$, while the total queue level at a node j given by Equation 1:

$$\mathcal{Q}_j(t) = \sum_{i=0}^{n-1} \mathcal{Q}_j^i(t) \text{ with } p(t) = j \text{ and } e \neq j \quad (1)$$

Therefore, when a message is transmitted, it is added to a queue that drains at its maximum rate $\tilde{B}_{jk}(t)$, with $\tilde{B}_{jk} \in \mathbb{R}^+$, which is the available bandwidth between j and k . Assuming that every destination is equiprobable, the individual queue drainage at j for messages with $e = k$ is therefore given by:

$$\mathcal{D}_j^k(t) = \begin{cases} -\frac{\mathcal{Q}_j^k(t)}{\mathcal{Q}_j(t)} \tilde{B}_{jk}(t), & \text{if } \mathcal{Q}_j^k(t) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Since the system is discretized with time increments δt , $\tilde{B}_{jk} = \tilde{B}_{jk} \delta t$. Each queue is increased if there is flow arriving from other nodes with a rate higher than the possible outgoing flow. The arrival at j 's queue with destination k , i.e. j is a routing node from a node i towards k , is defined by Equation 3.

$$a_j^k(t) = \begin{cases} \sum_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{\mathcal{Q}_i^j(t)}{\mathcal{Q}_i(t)} \tilde{B}_{ij}(t), & \text{if } \mathcal{Q}_i^j(t) > 0; \mathcal{R}_{i,k}(t) = j \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

If the receiving node is not the final destination, its queue will receive a positive arrival flow with the same ratio as the previous hops send. Therefore, each individual queue level derivative is:

$$\begin{aligned} \mathcal{Q}_j^k(t) &= -\frac{\mathcal{Q}_j^k(t)}{\mathcal{Q}_j(t)} \tilde{B}_{jk}(t) \mathbb{1}_{\mathcal{Q}_j^k(t) > 0} \mathbb{1}_{j \neq k} + \\ &+ \sum_{\substack{i=0 \\ i \neq j}}^{n-1} \frac{\mathcal{Q}_i^k(t)}{\mathcal{Q}_i(t)} \tilde{B}_{ij}(t) \mathbb{1}_{\mathcal{Q}_i^k(t) > 0} \mathbb{1}_{\mathcal{R}_{i,k}(t) = j} \end{aligned} \quad (4)$$

By observing Equation 4 for the case when $j = k$, i.e. data arriving at destination node, the rate change of the queue can be only $\mathcal{Q}_j^k(t) \geq 0$. At the

end of the simulation, the value of $\mathcal{Q}_j^k(t_h)$ when $j = k$ will be the total data received at node j . These equations permit observing, during the simulation execution, the queues' dynamics.

2.2 Modelling Average Dynamic Communication Latency

As mentioned in Section 2.1, each message \mathcal{M} has an associated parameter $\alpha(t)$, which represents the evolution of the message's age. The age, an abstraction for end-to-end delay of a message, is incremented every time the latter is being transferred or waiting in queue before arriving at the final destination. Since messages are not simulated individually in the proposed model, the average age will be modelled by using momentum conservation. Thus, it can be inferred at any moment, what is the actual expected end-to-end delay taking into consideration not only the current queue levels in each node, but also the accumulated latency from previous hops. For each individual queue \mathcal{Q}_i^k , there is an associated age α_i^k , where $\alpha \in \mathbb{R}^+$, $0 < \alpha < t_h$, and the evolution of the age in each queue is modelled by:

$$\begin{aligned}
 & [(\alpha_j^k(t - \delta t) + \delta t)(\mathcal{Q}_j^k(t - \delta t) - d_j^k(t - \delta t)) + \\
 & + \sum_{\substack{i=0 \\ i \neq j \\ \mathcal{H}_i^k(t)=j}}^{n-1} (\alpha_i^k(t - \delta t) + \delta t)d_i^k(t - \delta t)] - \alpha_j^k(t)\mathcal{Q}_j^k(t) = 0 \quad (5)
 \end{aligned}$$

Since, besides the current age, all the elements of the equation are known, $\alpha_j^k(t)$ can be isolated and calculated at each time step.

3 Mobile Ad Hoc System Model

This section introduces a model for mobile distributed systems by modelling its auxiliary subcomponents composed of: the network, routing, latency, mobility and message flow models. The models discussed in this section are needed to enable the implementation of the fluid model proposed in Section 2 as a mobile distributed system simulator.

3.1 The Network Model

Unlike infrastructure-based wireless topologies, ad hoc networks do not rely on a central entity to control connectivity, and nodes are connected when in radio reach from each other. These network topologies, which are composed by wireless mobile nodes, are modelled as an undirected graph $T = (V, E)$. The set of vertices N is composed by n nodes, where N_i , with $i \in \mathbb{N}$ and $0 \leq i < n$, and the set of edges E is composed by the edges $e = (N_j, N_k)$, where the distance $d_{jk} \leq d_r$, being d_r the communication range. The latter is defined by the wireless

radio technology modelled, which allow a maximum bandwidth $B \in \mathbb{R}^+$ defined in Mbps.

For simplicity, there is no membership control and every node in the vicinity of one of the nodes participating in the connected graph T , is henceforth considered a participating node. In order to establish connectivity between the nodes, the Euclidean distance between them is calculated, and when $d_{jk} \leq d_r$, the nodes can communicate with one-hop distance. When $d_{jk} > d_r$, the nodes can potentially still communicate with every other node in the same connected component via multi-hop routing.

3.2 The Multi-Hop Routing Model

There are many routing protocols applicable to mobile ad hoc systems [21], and for this study, a proactive, greedy, location-based routing protocol was modelled inspired by [22], since all the required metrics are simulated. Given the number of nodes n participating in the topology, with $n \in \mathbb{Z}$, the routing table \mathcal{R} is a matrix with dimensions $n \times n$. The line index i of \mathcal{R} represents a node in the topology, while each column j represents a destination node from i , $\forall(i, j) < n$. Therefore, the value of $\mathcal{R}_{i,j}$ represents the next hop data currently in i with destination j . In this model, to update the routing table, the system periodically evaluates the current position of the nodes and runs the Dijkstra algorithm [23] to calculate the shortest path for each origin/destination pair. The table is therefore updated with the next hop towards every destination that yields the shortest path.

3.3 Bandwidth

The bandwidth provided by the MAC/PHY technologies of a wireless stack is considered to be fully available for a node. However, considering a technology where all connections share the same channel such as some versions of IEEE802.11, this system bandwidth is shared among the sources injecting data into the node and all the other nodes receiving data from the former. Considering the set of nodes serving data flows to node j , S_j , and the set of nodes $O_{\mathcal{R}}$ being served as next hop $\mathcal{R}_{j,n}$ for node j with n destinations, the bandwidth available for the link between nodes j and k as seen at node j is:

$$\tilde{B}_{jk}(t) = \frac{B}{|S_j(t)| + |O_{\mathcal{R}}(t)|} \quad (6)$$

However, the receiving node k will eventually be served by other nodes and serve other nodes as well. So, the actual bandwidth available \tilde{B}_{jk} might be lower than what is expected by only observing j ; i.e., every link will be limited by the lowest available bandwidth between two nodes. Considering any destinations n being served by k :

$$\tilde{B}_{jk}(t) = \begin{cases} \tilde{B}_{jk}(t), & \text{if } \tilde{B}_{kn}(t) > \tilde{B}_{jk}(t) \\ \tilde{B}_{kn}(t), & \text{otherwise} \end{cases} \quad (7)$$

3.4 Latency Model

There are two main drivers for delays on a distributed system: the communication and the computation delays, and this work focuses on the former. It is expected that a multi-hop communication scheme has an impact on the overall network latency due to an accumulation of factors that add delays at each hop:

- Medium Propagation - Delay added due to the propagation of information in the medium. In this work, radio waves in the air are considered and their velocity approximated as $C = 299.792.458$ m/s, so the latency $t_{mp} \in \mathbb{R}^+$ is therefore $t_{mp}(t)_{jk} = d_{jk}/C$.
- Serialization - Delay added for actually transferring serialized data. Considering the current available bandwidth between the nodes $\tilde{B}_{jk}(t) \in \mathbb{R}^+$ measured in *Mbps* and the amount of data to be transferred $\mathcal{D} \in \mathbb{R}^+$ measured in *MiB*, the latency $t_s \in \mathbb{R}^+$ is therefore $t_s(t)_{jk} = \frac{\mathcal{D}}{(\tilde{B}_{jk}(t)/8)}$.

The total end-to-end expected latency from origin(o) to destination(d) $t_{od}(t) \in \mathbb{R}^+$ is therefore:

$$t_{od}(t) = \sum_{\substack{j_0=o \\ j_n=\mathcal{R}_{j,k}^{n-1}(t) \\ k_n=\mathcal{R}_{j,k}^n(t)}}^{k_n=d} t_{mp}(t)_{j_n k_n} + t_s(t)_{j_n k_n} \quad (8)$$

Additional delays added by hardware computation and due to medium access layers are not considered since they are dependent on specific hardware and chosen MAC protocol. The end-to-end latency can be estimated based on the path followed and each hop's delays estimations [24, 25].

3.5 Mobility Models

Since this study proposes the observation of distributed systems deployed in dynamic networks with time-varying topologies, the introduction of mobility models is required. With them, the nodes' positions periodically change and the state of other models are updated accordingly by recalculating distances between nodes. The following mobility models are implemented for the evaluation in Section 5. A comprehensive list of mobility models that are useful to ad hoc computing, in particular when dealing with FANETs, can be found in [21].

Attraction Model The attraction model was created to emulate the situation where a node is given a command to keep the position on a specific coordinate but it is unable to execute the command exactly due to external disturbances, such as lateral wind, consequently the node fails to keep its expected position $\mathcal{P}(t) \in \mathbb{R}^3$. To model disturbances, the attraction coordinates $\mathcal{A}(t) \in \mathbb{R}^3$, being a set $[x, y, z]$, has each Cartesian coordinate frequently modified by a random,

bounded variation $\epsilon(t) \in \mathbb{R}^3$ at a pre-defined frequency. At each time step, the position is changed by a $\delta_p(t) = \delta t/V_p(t)(\mathcal{A}_p(t) - \mathcal{P}_p(t))$ based on a pre-defined velocity $\mathcal{V} \in \mathbb{R}^+$, with $\delta_p(t) \in \mathbb{R}^3$ being a set $[x, y, z]$. It is, hence, calculated by Equation 9 and, as a proportional controller, the closer the node gets to the attraction point, the slower it moves.

$$\mathcal{P}(t+1) = [x(t) + \delta_{p_x}(t), y(t) + \delta_{p_y}(t), z(t) + \delta_{p_z}(t)] \quad (9)$$

Random Walk Model When the goal is to have a mobility model that allows the nodes to move freely in a pre-defined area, the Random Walk model is more suitable. It mimics applications where nodes do not have pre-defined mobility paths, or when they have but the scheduling of nodes in such paths is stochastic. In such a model, a limiting arena $\mathcal{L}_a \in \mathbb{R}^3$ is pre-configured, and an initial position $\mathcal{P}(0) = [x_0, y_0, z_0]$ is defined in runtime. A random velocity $\mathcal{V}_0 \in \mathbb{R}$ is also defined in runtime and it characterizes the initial direction given to each node. At each simulation time-step, the position is updated by a $\delta_p(t)$ (distance equivalent to the velocity and the time-step) and calculated as shown in Equation 9. When there is a collision of a node with a wall, the velocity has its signal inverted in the axis of the wall hit. As a consequence, the nodes keep bouncing inside the arena, but for simplicity, collisions between nodes are ignored.

3.6 Data-Flow Model

This data-flow model describes how the generated messages are distributed among the participating nodes during runtime. It mainly simulates the behaviour of messages being created by applications in one node and transferred to other nodes participating in the algorithm of the distributed application.

- Round Robin - each node becomes the source of traffic per round, and the source node is changed with every pre-defined service period defined by λ .
- Stochastic - the source node is selected randomly at each round according to a Poisson distribution. The flows are served following a pre-defined λ .
- Fixed - in this model, the source node (selected based on experiment objectives) is fixed during all the execution, and the destination is defined by a set of nodes. When left empty, all nodes will serve as destination. The data flows are served following a pre-defined λ .

4 Model Validation

To validate the behaviour of the proposed fluid model, a scenario where two source nodes inject data flows towards two different destination nodes via a bottleneck, as shown in Figure 2a, was simulated with the model, a simulator (OMNet++) and an emulator (MACE). First, periodic flows of 1kB with arrival rate $\lambda = 1$ from S_0 to D_0 and S_1 to D_1 . After 5 seconds, a second periodic flow of same size but arrival rates $\lambda = 0.1$ from S_0 to D_0 and $\lambda = 0.5$ from S_1 to D_1 is introduced.

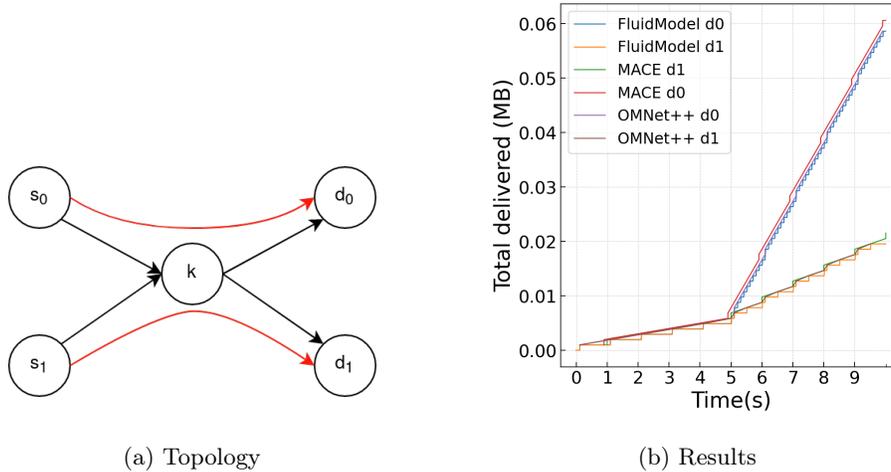


Fig. 2: Model validation with a network emulator and a simulator

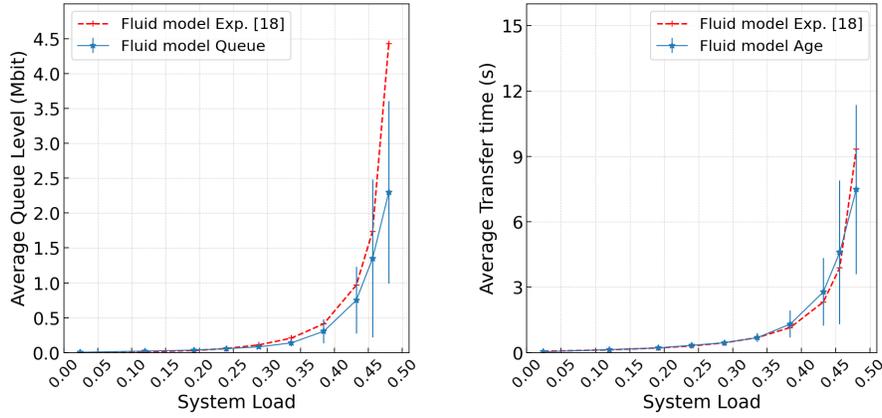
Figure 2b shows the evolution of the total delivered data for each tool and destination node, showing that even having fluid quantities that mix in the bottleneck node, it is possible to achieve results with the same precision as a discrete simulator without the need of representing each network packet. In [19], the author proposed a fluid model for representing the behaviour of a bottleneck node in an ad hoc wireless network. Their model, which aims in estimating steady-state values, was implemented and validated with a discrete network simulator. The same scenario and models described in the paper were implemented and compared to our model, achieving similar results as shown in Figure 3 for an exponential distribution of the mean flow size.

In this scenario, mean flow size $f = 0.12Mbit$ and system capacity $C = 5Mbps^4$ are fixed, while the mean flow arrival increases to achieve a system load $\rho = (\lambda f)/C$ close to 0.5, which is the maximum stable load due to the fact that each arriving flow must be served twice before reaching the destination.

5 Implementation and Evaluation

In this section, the described models' implementations are used to evaluate and illustrate its ability to study data flow in mobile ad hoc distributed systems. Some stress-test traffic is injected with the dataflow models described in Section 3.6 so that the topologies reveal potential hotspots, and the impact they have on hindering the overall system's performance. In order to evaluate the effect of the data flow, the injected traffic is gradually incremented through the network which is configured as in Table 2 and runs in a regular *Intel i7-8550U* home computer with 16GB of DDR4 RAM. The evaluation then continues in subsections

⁴ Approximate net capacity of a IEEE802.11b link



(a) Bottleneck Queue Stability Limit

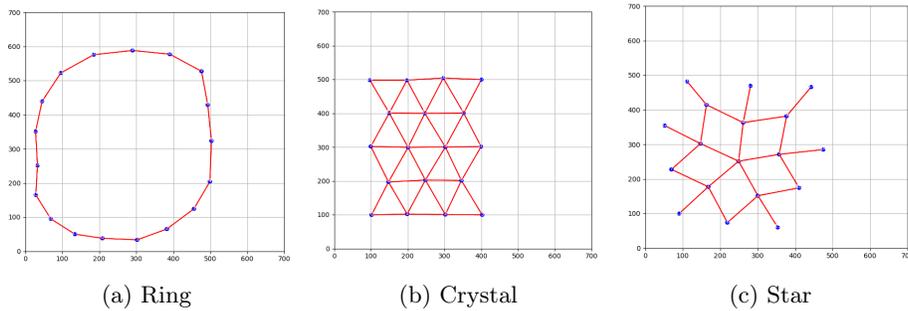
(b) Average Transfer Time

Fig. 3: Model validation for average steady state queue level and transfer time for arbitrary particle during system load saturation.

that expose different features, starting from congestion throughout latency and scalability. The evaluation was run with the three different⁵ topologies illustrated in Figure 4.

Table 2: Network Configuration

Bandwidth:	54 Mbps
Time Horizon:	100 s
Processing Latency:	1 ms
Route update interval:	1 s



(a) Ring

(b) Crystal

(c) Star

Fig. 4: Topology samples used for experimentation.

⁵ With a fixed number of 18 nodes.

5.1 Data Throughput

Even though a mobile ad hoc topology can implement any type of distributed application, as long as the application requires that data transmission is associated with the execution of the required task or function, the introduction of network latency shall hinder the system performance. In applications that require data transfer, the data needs to be copied or moved before the computation associated with such data can start. Some protocols optimize the data usage to avoid unnecessary traffic flow, but reducing unwanted latencies will allow the system to transfer more data in the same amount of time.

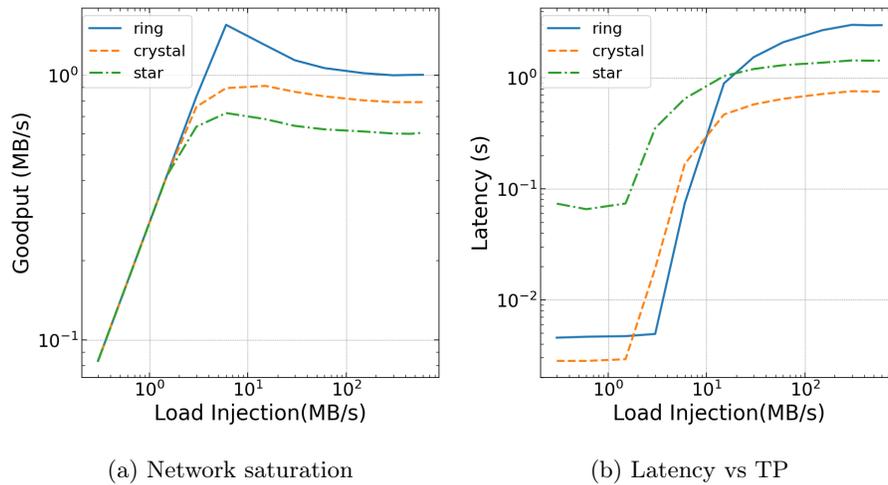


Fig. 5: Network saturation while injecting 3MB, *Round Robin* w/ variable frequency.

In Figure 5a, the goodput, which is the useful data delivered at final destination, is represented as it grows together with the injected load until saturation occurs. Also, the topologies saturate at different levels of traffic, and with different steady-state levels of goodput. Such saturation can also be observed in the latency increase as shown in Figure 5b. It is noticeable that regarding latencies, the topologies behave differently. This is specially evident in the ring topology, where the nodes do not have alternative routes, which results in higher latencies due to queue formation.

5.2 Congestion

As mentioned in 1.1, when running distributed applications in ad hoc networks, unbalances are likely to occur in some nodes as a consequence of the multi-hop traffic flow. When running topologies such as illustrated in Figure 4, some end-to-end preferred routes will be established. As a consequence, due to

the data flow’s characteristics, some nodes will have more data traversing it than others. Such nodes are more prone to congestions, which might increase the end-to-end latency and as consequence reduce the performance as shown in Section 5.1. Since the first goal is to observe the formation of hotspots in the topologies, the queues in each node are modelled boundless and no queue management algorithm is put in place. In Figure 6a each node was offloading 0.5 MB of data with the other nodes in the topology from Figure 4b in a round robin scheduling and interval of 100ms.

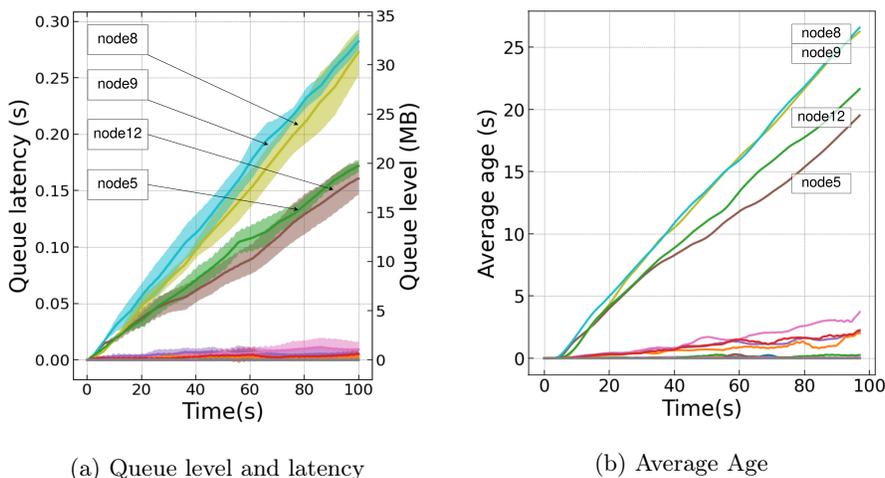


Fig. 6: Stress load injection of 0.5MB, *Round Robin* ($\lambda = 10$).

It is possible to observe that four nodes were preferred routes of the data flow and allowed the formation of queues. The shadows in the plots represent the standard deviation of the collected data, because since the mobility imposed on the nodes has a random component to their behaviour, each simulation results will differ slightly.

5.3 Latency

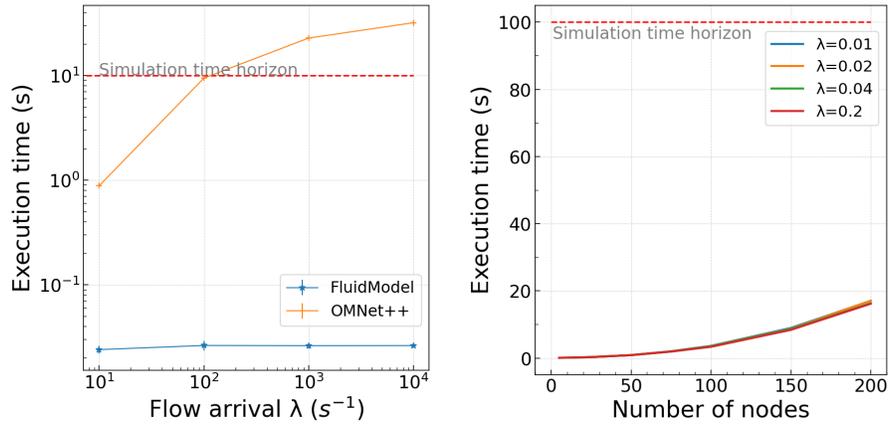
Communication latency is mainly characterized as described in Section 3.4 and Section 2.2. When abstracting unbounded queues in each node, due to the formation of transient or permanent bottlenecks, the network packets will undergo an extra latency besides the one created by the bounded bandwidth available for communication. When waiting in a queue to be served by the communication system, packets are undergoing an unexpected latency that can be estimated by Little’s Law [26]. Figure 6a also illustrates the expected waiting time in each node when using Little’s law. Even though this estimation gives an

insight of additional expected latency, it is incomplete since it takes in consideration only the current state of the system. The model described in Section 2.2 however, since it represents an average latency based on the history of the propagated end-to-end latency, can estimate the latter without the need of measuring each message individually.

Figure 6b shows the average total latency expected for the messages measured in each of the nodes. It is possible to observe the higher magnitude compared with Figure 6a due to the fact that it does not represent the expected added latency by each node’s queue, but the average *age* of each message that has traversed the node.

5.4 Model Scalability Evaluation

One of the main advantages of fluid models compared to discrete models is scalability. The results in Figure 7a, which corresponds to a topology with five nodes, with three of them injecting flows of 1kB with an increasing arrival rate, shows that with OMNet++ the simulation runtime surpasses the time horizon even with slow flow arrival. The runtime with the fluid model is barely influenced by the arrival rate, i.e. the number of network packets.



(a) OMNet++ with 18 nodes and variable injection (b) Model with variable node number and injection

Fig. 7: Scalability evaluation with time horizon of 100s.

Albeit not majorly impacted by the frequency of injected flows, the fluid model has runtime more impacted by the number of nodes in the topology. Figure 7b illustrates, for different arrival rates, a randomly deployed topology with increasing node count. Even though the runtime increases, it remains lower than the simulation time horizon.

6 Related Work and Limitations

In [18] and [19], the authors proposed a fluid flow model for representing a bottleneck being served by multiple sources and serving one node. In [27], the authors expanded the work by modelling resource sharing techniques for performance enhancement. In addition to modelling bottlenecks, our work went further with the ability to represent flows with different routes and mobility, hence being suitable for dynamic networks. Furthermore, our model allows the study of queue and latency dynamics during the whole execution, not only steady-state. The work presented in [17] also introduced a fluid model for ad hoc networks, but with a focus on energy consumption. In the domain of static networks, other works created fluid models based on differential equations, such as in [20] and even the creation of a framework in [28]. This work is also inspired by the fluid model described in [29]. The main limitation, when compared to discrete simulators, is when there is a need to investigate detailed network behaviour and implement different wireless stacks. Our model does not implement detailed contention algorithms, so its overhead must be taken into consideration by using an average usable bandwidth value. The current model is insensible to flow size, so different average flow sizes yield similar results.

7 Conclusion and Future Work

In this paper, a lightweight analytical model was introduced, implemented, validated and tested. Its applicability as a tool for faster than real-time simulations of different deployment topologies for swarms of mobile distributed systems was demonstrated. The model, even running on a personal computer, can simulate hundreds of nodes in less time than the simulation time horizon. With this model, future work will allow research of self-stabilizing topology control mechanisms, optimized node positioning and replica placement schemes for distributed systems. The work can be expanded also for the study of QoS, routing and medium access protocols.

References

1. A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, and J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," *Journal of Systems Architecture*, vol. 98, no. February, pp. 289–330, 2019. [Online]. Available: <https://doi.org/10.1016/j.sysarc.2019.02.009>
2. S. Wang, Y. Zhao, J. Xu, J. Yuan, and C. H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019. [Online]. Available: <https://doi.org/10.1016/j.jpdc.2018.06.008>
3. G. H. Forman and J. Zahorjan, "The Challenges of Mobile Computing," *Computer*, vol. 27, no. 4, pp. 38–47, 1994.

4. M. Satyanarayanan, W. Gao, and B. Lucia, "The computing landscape of the 21st century," *HotMobile 2019 - Proceedings of the 20th International Workshop on Mobile Computing Systems and Applications*, pp. 45–50, 2019.
5. M. Satyanarayanan, N. Beckmann, G. A. Lewis, and B. Lucia, "The Role of Edge Offload for Hardware-Accelerated Mobile Devices," *HotMobile 2021 - Proceedings of the 22nd International Workshop on Mobile Computing Systems and Applications*, pp. 22–29, 2021.
6. A. Trivedi, L. Wang, H. Bal, and A. Iosup, "Sharing and caring of data at the edge," in *3rd USENIX Workshop on Hot Topics in Edge Computing (HotEdge 20)*. USENIX Association, Jun. 2020. [Online]. Available: <https://www.usenix.org/conference/hotedge20/presentation/trivedi>
7. P. Gaynor and D. Coore, "Towards distributed wilderness search using a reliable distributed storage device built from a swarm of miniature uavs," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2014, pp. 596–601.
8. M. Cicala, E. D'amato, I. Notaro, and M. Mattei, "Scalable distributed state estimation in UTM context," *Sensors (Switzerland)*, vol. 20, no. 9, pp. 1–20, 2020.
9. H. Zhang, H. Dai, Z. Zhang, and Y. Huang, "Mobile Conductance in Sparse Networks and Mobility-Connectivity Tradeoff," *IEEE Transactions on Wireless Communications*, vol. 15, no. 4, pp. 2954–2965, 2016.
10. Z. Khan, P. Fan, and S. Fang, "On the connectivity of vehicular Ad Hoc Network Under Various Mobility Scenarios," *IEEE Access*, vol. 5, pp. 22 559–22 565, 2017.
11. S. Vashisht, S. Jain, and G. S. Aujla, "MAC protocols for unmanned aerial vehicle ecosystems: Review and challenges," *Computer Communications*, vol. 160, no. April, pp. 443–463, 2020. [Online]. Available: <https://doi.org/10.1016/j.comcom.2020.06.011>
12. M. Brambilla, E. Ferrante, M. Birattari, and M. Dorigo, "Swarm robotics: A review from the swarm engineering perspective," *Swarm Intelligence*, vol. 7, no. 1, pp. 1–41, 2013.
13. A. Bagnato, R. K. Bíró, D. Bonino, C. Pastrone, W. Elmenreich, R. Reiners, M. Schranz, and E. Arnautovic, "Designing swarms of cyber-physical systems: The h2020 cpswarm project: Invited paper," in *Proceedings of the Computing Frontiers Conference*, ser. CF'17. New York, NY, USA: Association for Computing Machinery, 2017, p. 305–312. [Online]. Available: <https://doi.org/10.1145/3075564.3077628>
14. M. Minelli, J. Panerati, M. Kaufmann, C. Ghedini, G. Beltrame, and L. Sabattini, "Self-optimization of resilient topologies for fallible multi-robots," *Robotics and Autonomous Systems*, vol. 124, p. 103384, 2020. [Online]. Available: <https://doi.org/10.1016/j.robot.2019.103384>
15. M. Li, Z. Li, and A. V. Vasilakos, "A survey on topology control in wireless sensor networks: Taxonomy, comparative study, and open issues," *Proceedings of the IEEE*, vol. 101, no. 12, pp. 2538–2557, 2013.
16. R. W. Coutinho, A. Boukerche, L. F. Vieira, and A. A. Loureiro, "Underwater wireless sensor networks: A new challenge for topology control-based systems," *ACM Computing Surveys*, vol. 51, no. 1, 2018.
17. N. Sonenberg and P. G. Taylor, "Networks of interacting stochastic fluid models with infinite and finite buffers," *Queueing Systems*, vol. 92, no. 3-4, pp. 293–322, 2019. [Online]. Available: <https://doi.org/10.1007/s11134-019-09619-w>
18. H. V. D. Berg, M. Mandjes, and F. Roijers, "Performance Modeling of a Bottleneck Node in an IEEE 802 . 11 Ad-Hoc Network," *Ad-Hoc Networks and Wireless*, pp. 321–336, 2006.

19. F. Roijers, H. Van Den Berg, and M. Mandjes, "Fluid-flow modeling of a relay node in an IEEE 802.11 wireless ad-hoc network," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4516 LNCS, pp. 321–324, 2007.
20. M. A. Marsan, M. Garetto, P. Giaccone, E. Leonardi, E. Schiattarella, and A. Tarello, "Using partial differential equations to model TCP mice and elephants in large IP networks," in *Proceedings - IEEE INFOCOM*, 2004.
21. O. S. Oubbati, M. Atiquzzaman, P. Lorenz, M. H. Tareque, and M. S. Hossain, "Routing in flying Ad Hoc networks: Survey, constraints, and future challenge perspectives," *IEEE Access*, vol. 7, pp. 81 057–81 105, 2019.
22. H. Füller, M. Mauve, H. Hartenstein, M. Käsemann, and D. Vollmer, "Mobicom poster: Location-based routing for vehicular ad-hoc networks," *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 7, no. 1, p. 47–49, Jan. 2003. [Online]. Available: <https://doi.org/10.1145/881978.881992>
23. E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, p. 269–271, Dec. 1959. [Online]. Available: <https://doi.org/10.1007/BF01386390>
24. Z. Liang and L. Qilian, "Hop-distance estimation in wireless sensor networks with applications to resources allocation," *EURASIP Journal on Wireless Communications and Networking*, vol. 2007, 05 2007.
25. R. S. Oliver and G. Föhler, "Probabilistic estimation of end-to-end path latency in wireless sensor networks," in *2009 IEEE 6th International Conference on Mobile Adhoc and Sensor Systems*, 2009, pp. 423–431.
26. J. D. C. Little and S. C. Graves, *Little's Law*. Boston, MA: Springer US, 2008, pp. 81–100. [Online]. Available: https://doi.org/10.1007/978-0-387-73699-0_5
27. F. Roijers, H. Van Den Berg, and M. Mandjes, "Performance analysis of differentiated resource-sharing in a wireless ad-hoc network," *Performance Evaluation*, vol. 67, no. 7, pp. 528–547, 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.peva.2010.01.005>
28. M. Bonaventura and R. Castro, "Fluid-flow and packet-level models of data networks unified under a modular/hierarchical framework: Speedups and simplicity, combined," in *Proceedings - Winter Simulation Conference*, 2019.
29. A. De Cecco, "Modélisation Fluide de Réseaux," Theses, UNIVERSITE DE TOULOUSE ; UT3, Jun. 2016. [Online]. Available: <https://hal.archives-ouvertes.fr/tel-01486585>