Enhancing Performance and Longevity of Multi-radio Multi-channel HetNets through Dynamic Path-assignment

Paulo Alexandre Regis, Suman Bhunia, Shamik Sengupta
Department of Computer Science and Engineering
University of Nevada, Reno, USA, 89557
pregis@nevada.unr.edu, sbhunia@nevada.unr.edu, ssengupta@unr.edu

***Abstract*—In this paper, we present a novel approach to assign paths to data streams in a mission-centric heterogeneous wireless mesh network where nodes have multi-channel radio interfaces. Different than conventional packet-switching routing, our approach assigns the paths using a link utility based on sub-topologies of non-orthogonal frequencies. We use a graph coloring algorithm to calculate the utility of each link. We then propose two path assignment strategies: (i) single path, where each data stream traverse a unique path, and (ii) split-path, where the flows can be divided into multiple paths. Results show there is no best strategy that fits all scenarios. Instead, in different situations, different algorithms are better suited.**

***Keywords*—Heterogeneous, wireless network, flow assignment, energy efficient**

I. Introduction

The last decade has experienced a massive demand for wireless communication thanks to the advancement of hand-held electronic equipment. This advancement coupled with the development of lightweight unmanned autonomous vehicles (UAVs) brings a new era of mobile ad-hoc networking (MANET). The advent of class-1 (micro) and class-2 (small) UAVs and unmanned ground vehicles (UGV) not only increases the trend of 3D wireless mesh networks but also provides a new dimension to next generation wireless networking and service provision. This enables mission-centric operations such as tactical military networks, first responders network in a disaster zone, firefighters, etc., to take place in the absence of deployed infrastructure.

The enhancement in wireless connectivity escalates the demand for data transfer through increased amount of throughput hungry applications such as video monitoring streaming, large files transfer, etc. In a wireless mesh network, a node interferes with many neighbor nodes, even if they are not within communication range. Thus, single radio interfaces may not always be adequate. The priority of mission critical network is to provide uninterrupted communication. To strengthen reliability, these nodes are equipped with multiple heterogeneous radio interfaces that can operate on different spectrum frequencies. One simple example you can carry in your pocket, the smartphone, which can connect to LTE, WiFi, and Bluetooth networks simultaneously. In tactical and disaster zones, devices with multiple interfaces form a heterogeneous network (HetNets) that can provide wireless coverage over a wide area by relaying data over different frequencies, through multiple devices. In Figure 1 we can see an example of HetNet where multi-radio devices are connected to the network using different frequencies, represented by different colors.

The heterogeneous nature of the network poses some serious challenges for resource (links, flow, radio interface) allocation among the devices. Ideally, nodes in the vicinity should either use orthogonal frequencies or schedule the transmission to occur in non-overlapping time intervals (time slots), to avoid interference and carry out simultaneous communications. This time and frequency domain resource allocation becomes very complicated as soon as the devices are equipped with multiple radio interfaces. Only allocating time-frequency blocks does not guarantee optimal performance. In a network with energy constrained devices, increasing the throughput without concerning about the individual power consumption of the nodes can interrupt the connectivity of the entire system. In this paper, we focus on resource allocation by looking at the data demand of the network. To enhance the performance of a HetNet, the resources should be allocated in a way that satisfies flow of data while optimizing available spectrum, neighborhood interference and the longevity of the power constrained devices.

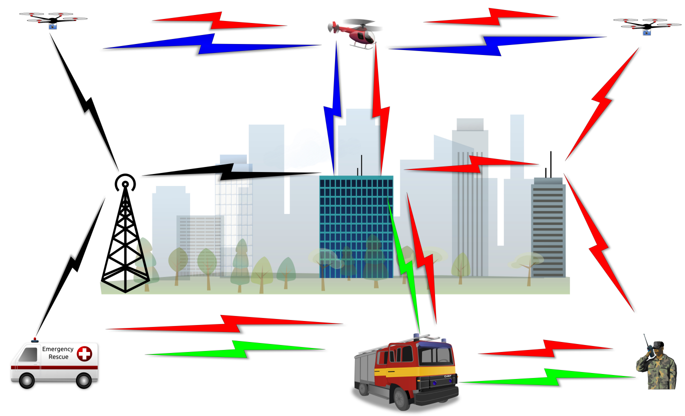


Figure 1: Example application scenario

Resource allocation in wireless ad hoc networks has been studied in many aspects such as joint power allocation and routing, joint routing and dynamic spectrum access, quality of service based resource allocation, energy efficiency, etc [1], [2], [3]. The work in [4] proposes different strategies to allocate power and minimize the bit error rate in multi-hop networks. While in [5] the proposed mechanism tries to maximize the throughput in cognitive radio ad hoc networks with a joint routing and dynamic spectrum access strategy. With the possibility of one node having multiple radio interfaces to access different mediums, this problem becomes even more complex. The work in [6] tackles the problem of resource allocation based on the quality of service requirements, and it considers only homogeneous networks, where a base station serves an arbitrary number of users. Similarly, [7] investigates the importance of energy efficiency in LTE and WiFi systems. The work in [8] investigates the channel assignment problem in wireless mesh networks. The problem is elaborated as a joint routing and channel assignment task, not considering power balance among the nodes. Regarding heterogeneous network resource allocation, [9] tries to maximize the energy efficiency of individual nodes, but not considering interference. [10] proposed a prioritized resource allocation method for heterogeneous networks. The model splits traffic flows into multiple paths to achieve better throughput, but energy efficiency and network longevity are not considered. However, without careful design of flow assignment and novel mechanisms to increase performance and longevity jointly, the very features of mission-centric multi-radio multi-channel HetNets can turn into disadvantages.

In his paper, we focus on resource allocation based on the mission-centric objective as well as longevity and quality of services. We propose a method to distribute the flows in a mission-centric HetNet. To tackle this issue we consider the traffic demand of a device as a flow. We also calculate the utility of wireless links between a pair of neighbors as a function of its capacity. This utility value is then used to decide the path of each flow in the network. In this paper, we propose two algorithms to allocate resources (time slot) to a link. We utilize a divide and conquer approach. Initially, multiple subgraphs are formed for each of orthogonal frequency band. We apply graph coloring algorithms to identify the required time slots for each frequency. Then each of the links is assigned a time slot through the proposed algorithms. The first algorithm considers the data packets from a source to destination can flow through a single path, whereas the second algorithm takes the advantage of split-path routing that allows a stream of data to follow different routes. The method improves the longevity of the network and reduces the problem of node starvation. By balancing the flows and accepting a trade-off with increased delay, our approach is able to enhance the performance when compared to conventional solutions, while maintaining good power balance among the nodes.

The remainder of the paper is organized as follows. Section II describes the system model and assumptions. In Section III we describe the utility model for the network links. Section IV presents the flow assignment strategies devised. Section V describes the simulation experiment setup, the parameters used, and presents and discusses results. Finally, Section VI gives the final remarks and future directions.

II. System model

In this section, we describe the mission-centric system model. We assume the network consists of an arbitrary number of heterogeneous nodes which are capable of adjusting the transmission power. Each node is equipped with multiple heterogeneous radios, capable of transmitting over different frequencies simultaneously [11], [12]. A command and control base (CC) which can be a node in the network or a remote station, decides the flow assignment. The placement of CC is beyond of the scope of this paper. In Figure 1 we see an example of such network the nodes are placed in a grid manner to facilitate the visualization. Each color indicates a different operating frequency. Each node is capable of transmitting in at least one frequency (at least one interface installed). Figure 2 shows the sub-topologies of the example network for different frequencies. A summary of the assumptions is shown next.

- There can be an arbitrary number of *sinks* which are just like the rest of the nodes, except when a traffic is generated, the destination is always one of the *sink* nodes. *Sinks* can also generate traffic, in which case the destination is a different *sink* node.
- Nodes, including $sinks$, can act as relays for all network flows.
- Nodes can generate traffic, $source$ and $destination$ must be different, $destination$ must be a $sink$.
- Nodes have an arbitrary number of radio interfaces.
- Each radio interface in a node allows the node to access a specific frequency.
- Nodes are stationary.

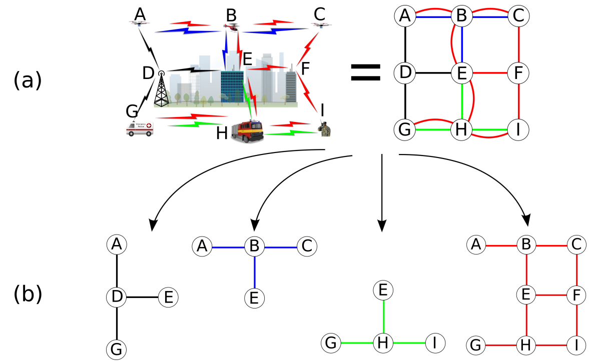


Figure 2: Sub topologies

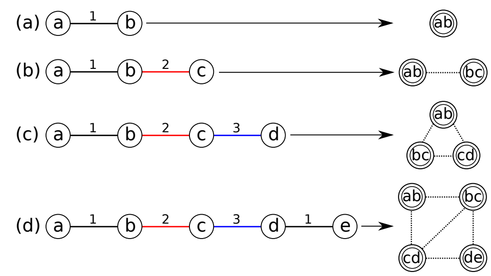


Figure 3: Interference model. Numbers represent time slots.

A. Flow model

Since the network is deployed to accomplish a certain mission, any node can generator traffic. We consider three types of traffic [13], [14]:

- **Streaming**: video and/or voice streaming, highest data-rate requirement
- **Imagery**: image file transfers, data-rate requirement only lower than $Streaming$
- **Sensor**: small data collected from sensors, lowest traffic data-rate requirement

Each flow has a source, destination, and a requirement. A node may perform more than one task at the same time where the flow to the *sink* will have an aggregated requirement of all the traffic generated at the source.

B. Interference model

From the complex network topology, we construct the interference graph for each frequency. Our model assumes a binary interference model [15], where a link simply interferes or not with the other links. A link interferes with another link if the distance is less or equal to three hops, this assures both links can be active at the same time. Although our work assumes this simple model, it can be further extended to utilize more complex ones. Figure 3 illustrates the model. From top to bottom we progressively include a new node to show when two links can be active at the same time. Links $a-b$ and $c-d$ cannot happen at the same time because there is a link between nodes b and c, thus if $c-d$ is active, b can receive what c is transmitting. The same does not apply to links $a-b$ and $d-e$, since there is no constraint between their nodes. Each topology is converted to its corresponding interference

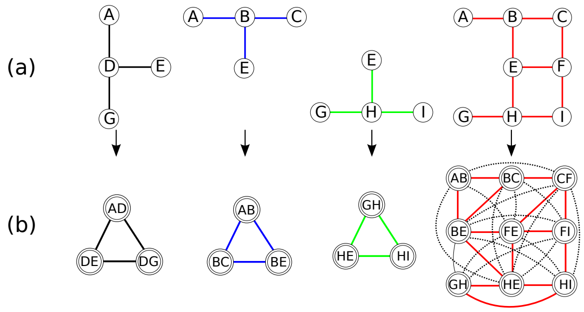


Figure 4: Building the interference graph

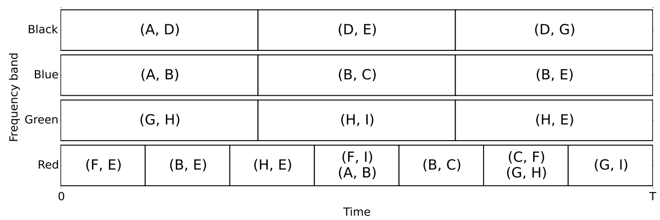


Figure 5: Assigned time slots to links

graph. The nodes in the interference graph (the double circles) are equivalent to edges in the topology graph, and the links (dotted lines) represent the constraints between links in the topology.

III. Utility model

After the network is divided into sub-topologies for different frequencies, time division multiplexing is used to avoid interference. Each link is assigned a time slot in which it can be active. The utility of a link is modeled as a function of channel capacity and time slots assignment. A time slot is the assigned interval in which it can communicate. The channel capacity is considered as Shanon capacity. $C = Blog_2(1+\frac{S}{N})$, where C is in bits per second, B is the bandwidth in hertz, and S and N are the signal and noise powers, respectively.

The number of required time slots in a sub-topology graph varies depending on the interference constraint graph, exemplified in Figure 4. To find the number of time slots necessary in each sub-topology we use a graph coloring algorithm. Each time slot is mapped to a different color, and the minimum chromatic number for frequency f (denoted by τ_f) of the interference graph is equivalent to the number of time slots necessary for that particular frequency. The link utility is defined as $U = \frac{C}{\tau_f}$, where U is measured in bits per second since we divide the channel capacity C into equally sized time slots.

After generating the interference graph with the constraints, we run the minimum degree ordering greedy algorithm [16] to find a feasible chromatic number of the interference graph in Figure 4b. Since graph coloring is an NP-complete problem [17], this may not yield the global optimum solution. Each color found with the algorithm represents a time slot in the x-axis of Figure 5.

IV. Flow assignment

In this Section, we propose two algorithms. The difference relies on the possibility of dividing a flow into multiple paths. The main idea behind the development of these algorithms is that by distributing all the traffic in the network in an evenly manner among the links of the network, the system would not suffer from bottlenecks. This happens when a single or a small subset of the nodes are overwhelmed with most of the load, causing the bottleneck nodes to run out of energy before the rest, leading to failures and degraded performance.

We devise two distinct algorithms with different goals in mind. The single path algorithm aims to provide resources to as maximum number of flows as possible, and the split-path strategy tries to provide resources to as much data-rate requirement as possible, regardless of the number of flows it represents.

A. *Single path strategy*

As the name suggests, in this strategy, a flow is assigned only one path between source and destination. It aims to provide resources to the most number of flows as possible. The algorithm tries to first assign path to the flows whose source is farthest from destination. The reasoning behind it is to try and spread the traffic load among the most links as possible and maintain the balance of quality among all flows. After assigning a path to a flow,the utility of the links in that path get updated each iteration. Then, the flow with the longest shortest path is selected first, the flows with shorter paths will eventually be forced to use a longer one in later iterations. In Algorithm 1 we can see the steps taken when trying to assign paths to the flows in the network. The input for the algorithm are the edge list and their respective utility, and the list of flows that need to be assigned. The output is a list of paths, one for each flow.

Algorithm 1: Single path balancing

Input: Edge list $\mathcal{E}$, edge utility list $\mathcal{U}$, flow list $\mathcal{F}$
Output: path per flow

$\mathcal{F}_a \leftarrow \emptyset$; $\mathcal{F}_u \leftarrow \emptyset$
while $\mathcal{F} \neq \emptyset$ **do**
  for $r \in \mathcal{F}$ **do**
    $l_i \leftarrow$ `FindPath`(r)
  $r_{lsp} \leftarrow argmax_i(l)$
  if $r_{lsp} = \infty$ **then**
    $\mathcal{F}_u = \mathcal{F}_u \cup \{r_{lsp}\}$
  else
    for $e \in path(r_{lsp})$ **do**
      $U_e \leftarrow U_e - r_{lsp}$
    $\mathcal{F}_a = \mathcal{F}_a \cup \{r_{lsp}\}$
  $\mathcal{F} \leftarrow \mathcal{F} \backslash r_{lsp}$

The algorithm takes two empty lists, $\mathcal{F}_a$ and $\mathcal{F}_u$, representing the flows that can be assigned, and the flows that cannot, respectively. Throughout the algorithm, flows are added to one of the two lists and removed from the input list $\mathcal{F}$. The procedure iterates until the input list becomes empty. In the iteration, The first step taken is to find the shortest path for each flow, with the utility of each edge being its weight. This enforces the lower utility links to be used first, which consequently translates to lower power consumption, since these links need much less power to achieve the same transmission range as the higher frequency ones. Next, the algorithm selects the flow with the longest shortest path (r_{lsp}). If the flow is unreachable, it is placed in the $\mathcal{F}_u$ list. If the path exists, then the utility of each link (U_e) in that path is reduced by the requirement of the flow (r_{lsp}). Then the flow is added to the $\mathcal{F}_a$ list, and removed from $\mathcal{F}$. The function $FindPath(r)$, does more than just finding the shortest path. It verifies if the path is feasible by comparing the utility of each edge to the requirement of the flow. If the shortest path is not feasible, the second shortest path is selected, and so on and so forth. If no path is feasible, then it returns infinity, which leads to the flow being placed in the corresponding $\mathcal{F}_u$ list.

We compare five variations of our algorithms. Each variation has a different strategy, but the underlying algorithm is the same.

i) *Longest shortest path, lower data rate link first* (LSP_LF): it uses Algorithm 1 as described before.
ii) *Shorter shortest path, higher data rate link first* (SSP_HF): a selfish strategy based on Algorithm 1. Selects the flow with the shorter shortest path first, based on the fastest path possible.
iii) *Longest shortest path, higher data rate link first* (LSP_HF): it finds the fastest path possible for the longest shortest path, it is based on Algorithm 1.
iv) *Longest shortest path, multi-path flow* (LSP_MP): it uses Algorithm 2 as it is originally described.
v) *Longest shortest path, hop count as weight* (LSP_HC): uses Algorithm 1 as the base algorithm, but assumes uniform weight when finding a feasible path, it is used as a benchmark scenario, since most networking protocols use hop count for path metric.

B. Split-path strategy

In the split-path algorithm, a function $FindPaths(r)$ tries to find the shortest path for a flow. If this shortest path can not satisfy the entire requirement, another shortest path route is considered for the remaining data-rate requirement. Finally the number of simultaneous paths needed are considered for the final flow assignments. Similar to the single path strategy, the split-path selects the flows in decreasing order of number of paths. In other words, it selects the flow that needs the highest number of sub-paths to transmit. This strategy aims to serve as much throughput requirement as possible, regardless of the number of flows. In Algorithm 2 we extended the previous algorithm by allowing one flow to have multiple paths towards the destination, dividing the requirement between these paths. The input remains the same, but the output of the algorithm returns a set of paths for each flow. Each path is used to send a fraction of the flow. This fraction is determined by the value of the path, that is the lowest link utility in it (or the remainder of the flow requirement). If the function $FindPaths(r)$ is not able to distribute the flow requirement between all the paths, the flow is then moved to the $\mathcal{F}_u$ set. If a feasible set of paths is found, then the utility of each link is decreased by a fraction of the flow that will travel through it. After the flow being assigned or not, it is removed from the flow set. Differently than Algorithm 1, we select the flow with the larger number of paths first. This approach tries to balance the performance in terms of power consumption as well as delay, by spreading the traffic of the network.

V. PERFORMANCE EVALUATION

We evaluate the system through simulations written in Python programming language. Due to the randomness aspects of the simulation, we draw the values by averaging the results of ten executions in each case. Table I lists the parameters used to calculate the transmission range of each adapter in the simulation. We tried to resemble as much as possible to the real applications. Values for the 60 GHz band, for example, may not reflect the reality since research in this spectrum band is an ongoing effort and no standard is available yet [18].

Algorithm 2: Split-path balancing (LSP_MP)

Input: Edge list $\mathcal{E}$, edge utility list $\mathcal{U}$, flow list $\mathcal{F}$
Output: set of paths per flow

$\mathcal{F}_a \leftarrow \emptyset$; $\mathcal{F}_u \leftarrow \emptyset$
while $\mathcal{F} \neq \emptyset$ **do**
  for $r \in \mathcal{F}$ **do**
    $l_i \leftarrow$ `FindPaths`(r)
  $r_{lsp} \leftarrow argmax_i(l)$
  if $path\_list(r_{lsp}) = \emptyset$ **then**
    $\mathcal{F}_u = \mathcal{F}_u \cup \{r_{lsp}\}$
  else
    for $path \in path\_list(r_{lsp})$ **do**
      for $e \in path)$ **do**
        $U_e \leftarrow U_e - min(value(path), r_{lsp})$
        $r_{lsp} \leftarrow r_{lsp} - min(value(path), r_{lsp})$
    $\mathcal{F}_a = \mathcal{F}_a \cup \{r_{lsp}\}$
  $\mathcal{F} \leftarrow \mathcal{F} \backslash r_{lsp}$

TABLE I: Frequency adapter characteristics

Carrier frequency	Bandwidth	Max Tx power (mW)	Sensitivity (dBm)	Description
850 MHz	24.6 MHz	10	-85	GSM
1.3 GHz	20 MHz	70	-81	Amateur radio
2.4 GHz	20 MHz	80	-80	WiFi 2.4 GHz
5 GHz	20 MHz	100	-79	WiFi 5 GHz
60 GHz	40 MHz	200	-78	mmWave

The total number of nodes in each simulation is 50. The number of radio interfaces in each node is taken from a uniform distributed interval $[1, 5]$. Also, each adapter is chosen randomly from the list in Table I. If a node has more than one adapter, they shall not work on the same frequency. Nodes cannot have 0 adapter. The simulated area is a $10km \times 10km$ square. Nodes are randomly scattered, except for the *sink* in the cases described below. We test the new mechanism in four different cases, each of which corresponds to a different location of the *sink(s)* node(s). The cases are described below and an illustration can be seen in Figure 6.

1) **Origin sink**: only one sink is present, placed at the origin.
2) **Center sink**: only one sink is present, placed at the center $(xmax/2, ymax/2)$.
3) **Random sinks**: half of nodes are *sinks*, randomly placed.
4) **Corner sinks**: four sinks, one in each corner.

Each node has a flow probability associated that tells which kind of traffic it generates. There is also the possibility that no event triggered the nodes hardware to collect information.

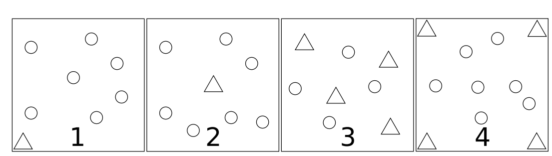


Figure 6: Visualization of *sink* distribution in different cases. Triangle represents sink, and circles the rest

The probabilities and requirements are as follows.

- *Streaming*: 1%, $1Mbps$
- *Imagery*: 10%, $100Kbps$
- *Sensor*: 60%, $1Kbps$
- *No traffic*: 29%, $0bps$

A. *Performance parameters*

i) **Power distribution:** the cumulative distribution of the energy consumption of all the nodes in the network.

ii) **Overall power consumption**: The amount of energy consumed by all nodes in the network, based on the amount of requirement assigned.

iii) **Average delay**: average delay of all the flows, taking in consideration the delay in each node. In the case of multiple paths, the delay of the flow is a weighted average, where the weight is the fraction of the requirement that traverses a path.

iv) **Average hop count**: the average hop count of each assigned flow path. Weighted average when a flow has multiple paths, similar to the previous.

v) **Flow assignment ratio**: the percentage of flows that were assigned a path.

vi) **Requirement assignment ratio**: the percentage of all the flows requirements which were assigned a path.

B. *Results and Discussion*

Figure 7 shows the power cumulative distribution of the nodes in the network for each case. In general, the more the curve approximates to the left axis, the better; it means more nodes are consuming less energy. It shows the percentage of the nodes (y-axis) that consumes up to a certain amount of energy (x-axis). We notice that the position of the sink plays an important role: when the sink is placed at the center, no node consumes more than

The *SSP_HF* and *LSP_HF* variations turned out to be the most inefficient methods in terms of power consumption, because of the greedy strategy to choose the highest data rate path possible. However, even though allocating more flow requirement than any other method, the *LSP_MP* strategy, was able to achieve similar results as the *LSP_LF* and *LSP_HC* in most cases because of the utilization of the lower data rate links before the higher ones when splitting the flow in multiple paths. In case 3 *LSP_MP* consumes more energy, but not more than *LSP_HF*, and *SSP_HF*.

In Figure 8 we can see that *LSP_MP* is able to over 100% more of the total flow requirements than the rest, but this comes to the cost of not assigning multiple number of flows. In fact, the multi-path strategy always assign a lower number of flows because it tries to allocate the flow with the highest number of paths first, which yields the flow with the higher demand.

The power consumption of methods *SSP_HF* and *LSP_HF* turned out as expected. Due to the nature of the strategy they consumed more energy by selecting the highest data-rate paths. *LSP_LF* achieved better results than the traditional hop count strategy, which can be a good option if the trade-off for a higher delay is acceptable.

Each strategy has its advantages and can be useful in different situations. Using *LSP_MP*, for example, is a good option when streaming and imagery is the main type of traffic in the network. While *LSP_LF* can substitute *LSP_HC* to improve the longevity of the network, at the cost of a small decrease in the performance.

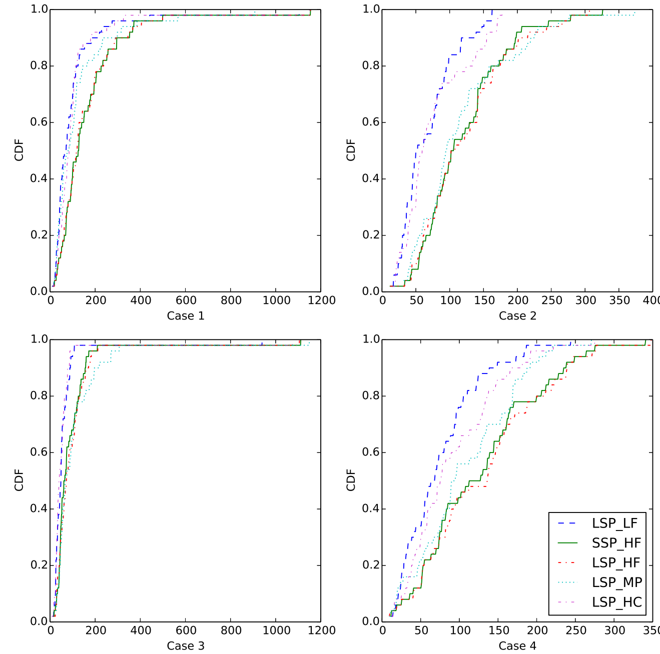


Figure 7: Power cumulative distribution function

VI. Conclusion

In this paper, we proposed a method to distribute the flows in mission-centric ad hoc heterogeneous networks. The methods demonstrated longevity of the network improvement, while avoiding the problem of node starvation. The approach increased the performance when compared to common solutions, and maintained good power balance among the network nodes, by balancing flows and allowing a trade-off with increased delay. The single path method LSP_LF showed better power consumption distribution than the other methods, 17.13% lower than LSP_HC. While the split-path strategy LSP_MP was able to assign more throughput requirement than all single path variations (6.66 times more than LSP_HC). LSP_MP, even assigning more throughput requirements, showed similar power distribution as the SSP_HF variation. We also showed, through different simulated scenarios, that the position and quantity of the $sink$ nodes affect the performance of the network.

References

[1] Y. Zhang, C. Lee, D. Niyato, and P. Wang, "Auction approaches for resource allocation in wireless systems: A survey," *IEEE Communications surveys & tutorials*, vol. 15, no. 3, pp. 1020–1041, 2013.

[2] D. Feng, C. Jiang, G. Lim, L. J. Cimini, G. Feng, and G. Y. Li, "A survey of energy-efficient wireless communications," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 167–178, 2013.

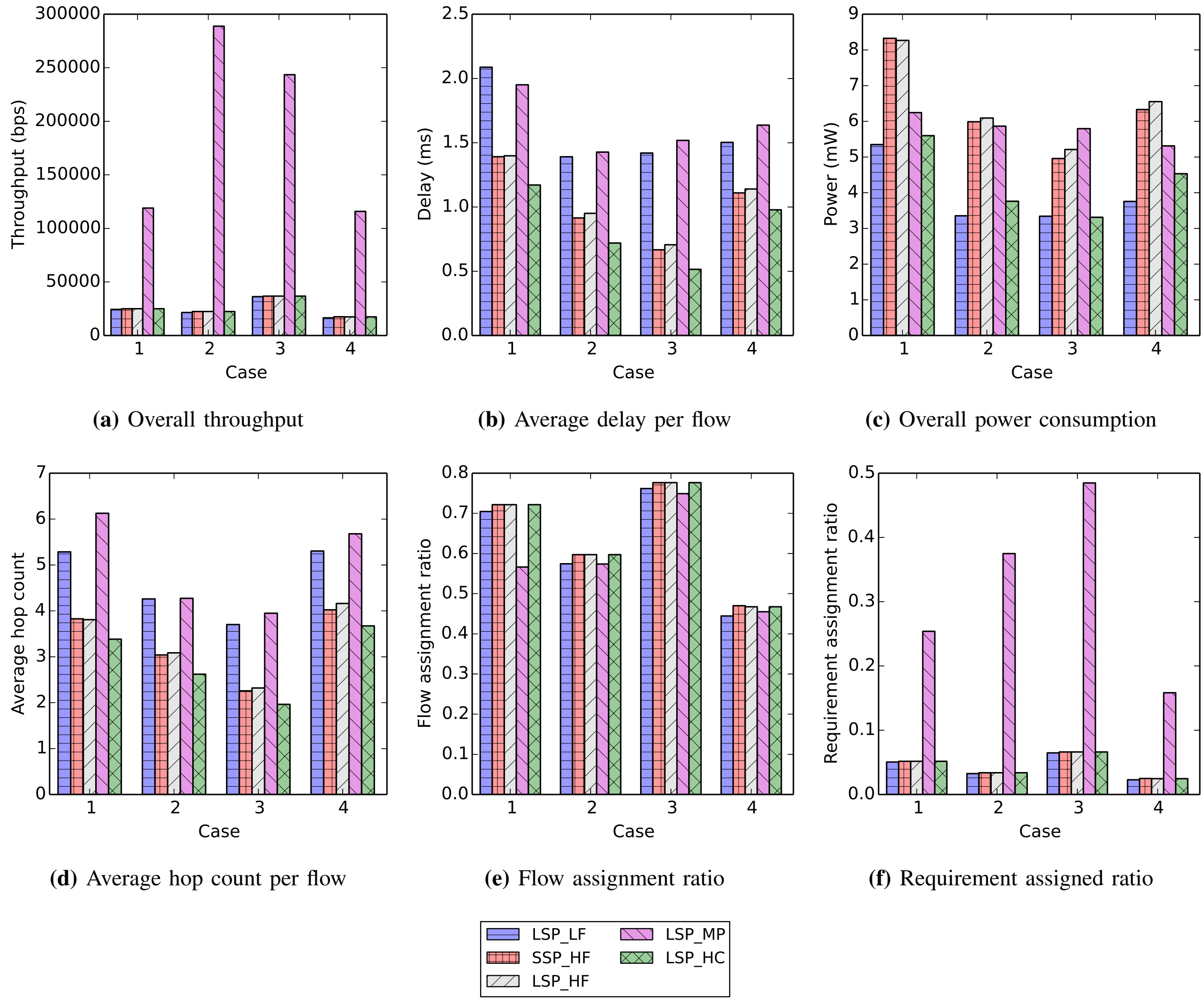
[3] N. Chakchouk, "A survey on opportunistic routing in wireless communication networks," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2214–2241, 2015.

[4] S. Gupta and R. Bose, "Joint power allocation and routing optimization in ber constrained multihop wireless networks," in *Communications (NCC), 2013 National Conference on*, pp. 1–5, Feb 2013.

[5] L. Ding, T. Melodia, S. N. Batalama, J. D. Matyjas, and M. J. Medley, "Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1969–1979, 2010.

[6] M. Fadel, A. S. Ibrahim, and H. Elgebaly, "Qos-aware multi-rat resource allocation with minimum transmit power in multiuser ofdm system," in *2012 IEEE Globecom Workshops*, pp. 670–675, IEEE, 2012.

[7] S. Navaratnarajah, A. Saeed, M. Dianati, and M. A. Imran, "Energy



(a) Overall throughput **(b)** Average delay per flow **(c)** Overall power consumption

(d) Average hop count per flow **(e)** Flow assignment ratio **(f)** Requirement assigned ratio

Figure 8: Performance results

efficiency in heterogeneous wireless access networks," *IEEE Wireless Communications*, vol. 20, pp. 37–43, October 2013.

[8] J. J. Gálvez and P. M. Ruiz, "Efficient rate allocation, routing and channel assignment in wireless mesh networks supporting dynamic traffic flows," *Ad Hoc Networks*, vol. 11, no. 6, pp. 1765–1781, 2013.

[9] G. Yu, Y. Jiang, L. Xu, and G. Y. Li, "Multi-objective energy-efficient resource allocation for multi-rat heterogeneous networks," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 10, pp. 2118–2127, 2015.

[10] M. Gerasimenko, D. Moltchanov, R. Florea, N. Himayat, S. Andreev, and Y. Koucheryavy, "Prioritized centrally-controlled resource allocation in integrated multi-rat hetnets," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–7, IEEE, 2015.

[11] M. Dillinger, K. Madani, and N. Alonistioti, *Software defined radio: Architectures, systems and functions*. John Wiley & Sons, 2005.

[12] Y. Choi, H. Kim, S.-w. Han, and Y. Han, "Joint resource allocation for parallel multi-radio access in heterogeneous wireless networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3324–3329, 2010.

[13] T. Mountassir, B. Nassereddine, A. Haqiq, and J. Mountassir, "On optimizing nodes deployment in multi-radio multi-channel wireless mesh networks," in *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, 2012.

[14] C. M. Viegas, F. Vasques, and P. Portugal, "Evaluating the impact of uncontrolled traffic sources upon real-time communication in ieee 802.11 s mesh networks," in *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, pp. 106–111, IEEE, 2014.

[15] A. Iyer, C. Rosenberg, and A. Karnik, "What is the right model for wireless channel interference?," *IEEE Transactions on Wireless Communications*, vol. 8, no. 5, pp. 2662–2671, 2009.

[16] D. J. Welsh and M. B. Powell, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, vol. 10, no. 1, pp. 85–86, 1967.

[17] I. Holyer, "The np-completeness of edge-coloring," *SIAM Journal on computing*, vol. 10, no. 4, pp. 718–720, 1981.

[18] T. O. Olwal, K. Djouani, and A. M. Kurien, "A survey of resource management towards 5g radio access networks," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2016.