PROJECT TITLE:
SECURE DEMAND-BASED ROUTING
IN DEVICE-TO- DEVICE COMMUNICATION
ENABLED 5G WIRELESS NETWORKS

by

Fernanda Tovar, Rithika Jaiswal

A project submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Electrical and Computer Engineering and Computer Science

Approved:

Anand Santhanakrishnan Project Advisor	N. Sertac Artan and R. Khalaj Amineh Committee Members

Yoshikazu Saito
Chairman, Department of ECE, SoECS

NEW YORK INSTITUTE OF TECHNOLOGY
New York, NY

2021

Abstract

The next generation of wireless networks promises the support of quality-of-service (QoS) features using Device-to-Device (D2D) communications. Criteria for routing in multi-hop D2D networks include bandwidth, energy, security and load balancing. This project proposes secure and reliable routing algorithms for multi-hop D2D networks. Varying demands and priorities for different source-destination pairs are taken into account to develop a prioritized demand based routing algorithm.

We first derive cut and distance upper bounds to determine the maximum achievable prioritized throughput. We then investigate the scenario when a set of colluding attackers choose to attack one or more of the source-destination pairs and the network expending defense resources to counter the attack. This attack defense scenario is modelled as a min-max optimization problem and the optimal strategies for the attacker and the network are obtained as the saddle point of the problem. We develop theoretical foundations to show that attackers and the network expend their sources in proportion only to each other and not depend on the actual demand or the throughput.

Numerical evaluation shows that the proposed algorithm can achieve as close to 92 to 94% of the maximum achievable throughput dictated by the theoretical bounds we derive. Comparison with existing algorithms show that when routes are limited, the proposed algorithm can achieve 28% to $3\times$ enhancement to the throughput. Under hostile environments (i.e., in the presence of optimized attacks), the proposed algorithm can yield 25% to $4\times$ enhancement to the throughput when the attacker and the network are equally capable. When the attacker is more powerful than the network, the throughput enhancement can be 30% to $4\times$. When the network is more powerful, the proposed algorithm provides identical performance enhancement as if the attack didn't exist.

Acknowledgements

First and foremost, we would like to thank God Almighty for giving us the strength, knowledge, ability, and opportunity to undertake this project and to persevere and complete it.

We would like to express our sincere gratitude to our advisor Prof. Anand Santhanakrishnan for continuous support, for his patience, motivation , enthusiasm, and immense knowledge.

Besides our advisor, we would like to thank the project committee members, Prof. Nabi Sertac Artan and Prof. Reza Khalaj Amineh, for the time they took to review and evaluate our project and provide us invaluable feedback.

Our sincere thank also goes to the Department Chairman, Dr.Yoshikazu Saito, who offered us such a huge opportunity and provided all the support we needed throughout our program.

Most importantly, we would like to express our very profound gratitude to our family, for providing us with unfailing support and continuous encouragement throughout the years of study.

Finally, we thank our friends and all the people who have supported us directly or indirectly to complete our project.

Fernanda Tovar, Rithika Jaiswal

Contents

List of Figures

Chapter 1

Introduction

1.1 Background and Significance

Over the years, wireless data traffic has been increasing due to 1.3 billion smart devices based on 5G technology projected by 2023 [3]. A larger network means that it has to take care of increasing demands, which are not met by traditional approaches [4]. The design of traditional IP routing reduces complexity at the routers, but it also creates an environment that is established. It depends on best effort delivery of packets, and it doesn't ensure the transmission of application data with tolerable delay, jitter or loss [5]. Routing based on flows would limit these issues.

The next generation of wireless networks promises the support of quality-of-service (QoS) features [1]. This routing technique can be labeled as flow-based routing or QoS routing [5]. QoS is determined by a set of resources, which are bandwidth, delay, jitter, and packet loss [1]. QoS routing improves upon the routing model due to the multiple paths to the same destination that can be calculated for different flows. Additionally, traffic can be changed quickly from its current path to an alternate path. Ultimately, there's increased performance with guaranteed bandwidth, fast error recovery, and dynamic load balancing [5].

Device-to-Device (D2D) [4] communication between devices (i.e., cell phones or vehicles) gained popularity in recent times. This is an exciting concept that promises a better performance of devices by allowing nearby user pairs to have direct communication with enhanced throughput, lower energy consumption, coverage, and reduce end-to-end latency [6]. The advantages include a clear and robust traffic path that unloads easily [4], thus making QoS based routing more important.

D2D has created many opportunities in peer-to-peer services and location-based applications (i.e., public safety services, connected vehicles, cellular offloading and multi-hop relaying). Also, it is ideal for local data services through uni-cast, multi cast, and broadcast

mechanisms [7]. For example, Facebook [8] is a social proximity-based application that suggest friends based on current location. Another example is the Google Chromecast [9] streaming service that forms clusters and multi-casts data within a cluster [10].

Mobile adhoc networks (MANETs) with D2D capabilities optimize this growing market of fast and automatic communication [11]. The characteristics of MANETs include self-organization, constraint resource availability, frequent changes in topology, multi-hop routing, scalability, and limited security [12]. The metrics for route selection include energy consumption, node mobility, and link bandwidth. Since the nodes in MANETs are resource-constrained, the routes between source-destination pairs must be optimized to enhance the battery life and link utilization [13].

1.2 Motivation and Proposed Research

Criteria for routing in multi-hop networks include bandwidth, energy, security and load balancing. This project aims to develop secure and reliable routing algorithms for multi-hop D2D networks. In addition to these routing algorithms, varying demands for different users and from different sources will be taken into account. The Multi-Commodity Flow Problem (MCFP) will be explored to develop the algorithms for multi-hop D2D networks. MCFP is considered a difficult mathematical programming problem especially when there is a large number of commodities [14]. Examples include the general integral MCFP problem and the two commodities integral flow problem [15].

We deploy MCFP as our foundation, to maximize prioritized flow from all source-destination pairs, with minimized cost, while meeting capacity constraints. The novelty of our research is the focus on maximizing prioritized information flow without being biased only to high priority traffic. In our approach, we solve the Max Concurrent Flow Problem with enhanced security.

Our model can be depicted as a home network with different family members that have different demands from the internet as shown in Fig. 1.1. A commodity/service should be transported from their respective sources to their respective destinations, to meet the demands of the destinations. Some services like downloading an official document may have higher priority compared to watching a Netflix or an Instagram video [16].

Each link in the network has a limited capacity (i.e., bandwidth) which has to be shared by *all commodities / flows* incident on the link. There is a cost in terms of energy or security for every link. The goal of this project is to develop a routing algorithm that optimally meets

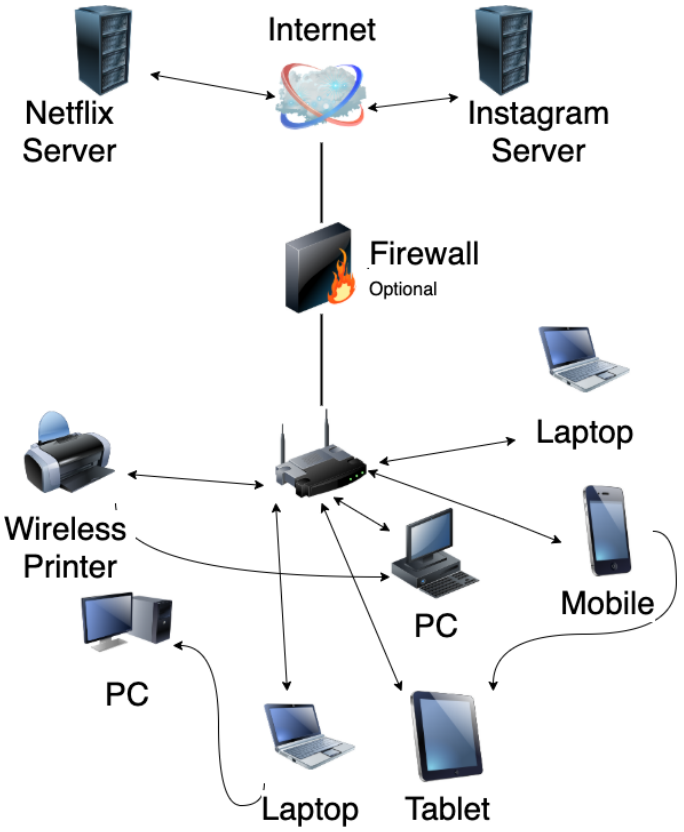


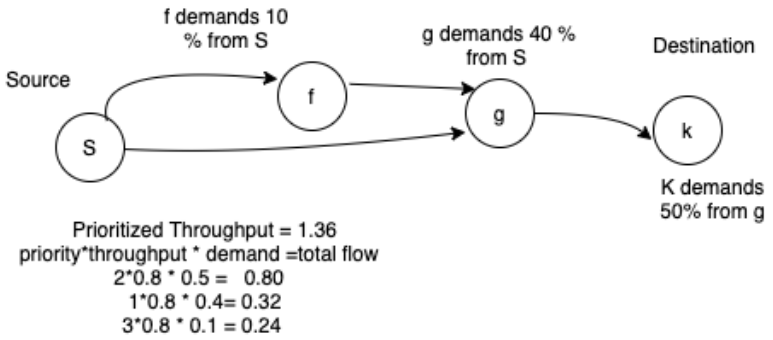
FIGURE 1.1: Home Network with different destinations and sources

the demands of the destinations based on their priorities, while keeping into consideration bandwidth and security requirements[1]. Current approaches in the literature deploy a greedy approach of making $Z(i,j) = 1$, for the highest priority demand and then $Z(i,j) = 2$ the next highest priority demand and so on [17]. Other approaches include either meeting 100% of some demands and 0% of the rest (called " All or Nothing") [1],[18] or meeting the *same fraction of all demands* [2] which yields larger average throughput compared to [1],[18].

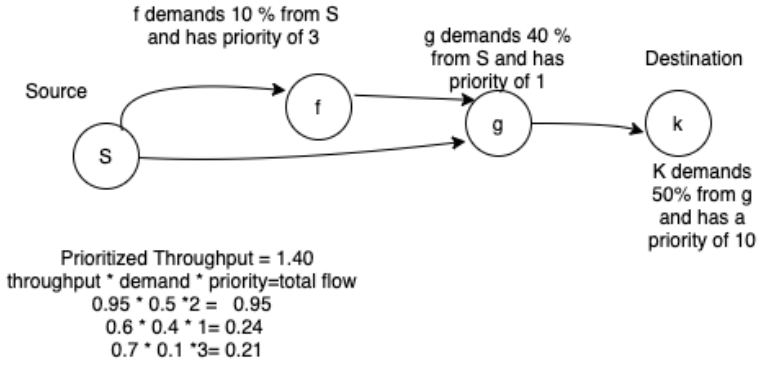
However, meeting equal fraction of all demands does not guarantee optimal throughput performance. As an example, consider the network shown in 1.2. By implementing the algorithm in [2], an average throughput of 0.8 is obtained. We tried an alternate heuristic and find that a weighted throughput of 1.36, i.e., an enhancement of $\frac{1.36-0.8}{0.8} \times 100 = 70\%$ can be obtained. **We use this result as a motivation to develop a theory grounded routing algorithm to provide enhanced throughput and study its impact on security**.

We propose a multi-commodity flow maximization algorithm to meet the demands of source-destination pairs in the network. We prioritized the demands based on the type of traffic. Our algorithm then maximizes the prioritized information flow in the network. We first derive

[1]Energy is posed as a consequence of the proposed mechanisms as against a constraint.



(a) Existing Approach [2]



(b) Our proposed heuristic

FIGURE 1.2: Example to Motivate our Research

cut and distance upper bounds to determine the maximum achievable prioritized throughput. We then investigate the scenario when a set of colluding attackers choose to attack one or more of the source-destination pairs and the network expending defense resources to counter the attack. This attack defense scenario is modelled as a min-max optimization problem and the optimal strategies for the attacker and the network are obtained as the saddle point of the problem. We develop theoretical foundations to show that attackers and the network expend their sources in proportion only to each other and not depend on the actual demand or the throughput.

Numerical evaluation shows that the proposed algorithm can achieve as close to 92 to 94% of the maximum achievable throughput dictated by the theoretical bounds we derive. Comparison with existing algorithms show that when routes are limited, the proposed algorithm can achieve 28% to 3× enhancement to the throughput. Under hostile environments (i.e., in the presence of optimized attacks), the proposed algorithm can yield 25% to 4× enhancement to the throughput when the attacker and the network are equally capable. When the attacker is more powerful than the network, the throughput enhancement can be 30% to 4×. When the network

is more powerful, the proposed algorithm provides identical performance enhancement as if the attack didn't exist.

The rest of the project report is organized as follows. Chapter 2 describes the current art in this research topic and the research gaps. The fundamentals of the Single Commodity Flow Problem (SCFP) and Multi-Commodity Flow Problem (MCFP) are provided in Chapter 3. In Chapter 4, we detail the theoretical foundations developed in this research that lead to an optimal routing algorithm. Also in the same chapter, we discuss the mechanisms to incorporate security and energy efficiency requirements. Numerical results are discussed in Chapter 5 and conclusions are drawn in Chapter 6.

Chapter 2

Literature Review

The literature review is divided into four sections, which include Approaches to the Multi-commodity Flow Problem (Section 2.1), Energy Conservation and Bandwidth Consumption (Section 2.2), Prioritization (Section 2.3), and Real Life Applications (Section 2.4).

2.1 Approaches to the Multi-commodity Flow Problem

In [1], Chika used a general maximum flow algorithm [1] to find a feasible path for bandwidth aggregation. Their purpose was to maximize resource usage on all paths in networks. Shahrokhi and Matula [2] focused on the Maximum Concurrent Flow problem, which is a subset of the Multi-Commodity Flow Problem. The principal result of this paper is an appropriately modified polynomial-time approximation version of the flow rerouting algorithm for the Max Concurrent Flow Problem allowing arbitrary demands with uniform capacity.

A similar formulation was provided by Nahabedian [19]. The author discussed shipping multiple commodities simultaneously through a network so that the total flow over each edge does not exceed the capacity of that edge. The author solved the concurrent flow problem for each commodity, with a demand d, and involved finding the maximum fraction z, such that z of each commodity's demand can be feasibly shipped through the network. Madry [20] realized the importance to compute an approximate solution in a faster manner than to compute an optimal one. Although all the problems defined can be solved optimally in polynomial time by formulating them as linear programs, they wanted to obtain faster approximation schemes for various versions of the Multi-Commodity Flow Problem.

Dai et al [21] has discussed solving Multi-Commodity Flow Problems.They presented four new methods: none-objective method, edgepath method, shortest-path method, and the widest-path method. They solve the price problems that are not only linear programs, but also problems

[1] Description of Single and Multi-Commodity flow is provided in Chapter 3

for finding shortest paths. Therefore, two methods, finding shortest paths with commodities (SPC) and finding shortest paths with sources (SPS), were proposed. Depending on the commodities either SPC or SPS is used.

MCFP is applied on Software Defined Network (SDN), Farrugia et al [22] developed a routing algorithm, based on a multi-objective genetic algorithm (MOGA) to increase the overall network utilization by using the information at the SDN controller. In another paper by Farrugia et al [23], they improved the performance by introducing an Evolutionary Algorithm (EA), to further enhance network efficiency. Cheng [24] introduced Multi-Commodity Flow Problem with local constrains described as Maximum Multi-Commodity Flow Problem with local requirement (MMFP-LD). The effectiveness, complexity and approximation measures of the designed approximation algorithm are discussed.

2.2 Energy Conservation and Bandwidth Consumption

Yang et al [25] studied network energy saving under demand changes and delay constraints. They developed a non-linear integer programming routing adjustment to minimize energy consumption. Curry [26] considered a data routing plan in a manner that maximizes the lifetime of a wireless sensor network (WSN). Lu et al [27] studied the problem on how to route and schedule inter-data-center bulk data transfers to minimize the energy-cost for geo-distributed data centers. This problem is modeled as a Min-Cost Multi-Commodity Flow Problem and an efficient two-stage optimization method is developed to solve it. As the interest in hybrid energy systems is increasing, Uiterkamp et al [28] presented a planning-based Multi-Commodity energy management system that scales linearly with the number of commodities and connected devices. This approach is applied to balance the three phases in a low voltage grid while charging a fleet of electric vehicles.

Multi-Commodity has applications in today's internet and consumes a large amount of bandwidth resource. Zhang et al [29] introduced a two-level decomposition (TLD) model for Multi-Commodity multi-cast. This model can achieve minimum-cost traffic engineering in the presence of uncontrolled traffic as network coding is allowed. Samani and Wang [30] proposed Max Stream, a SDN-based flow maximization framework. Here Multi-Commodity Flow Problem is introduced with an objective to maximize the number of streaming sessions to improve the providers revenue (quantity of the service) and also maximizing bandwidth provisioning for QoS enhancement (quality of the service).

2.3 Prioritization

Khanal et al [17] focused on commodity priority, where commodities are transshipped from the respective sources to the corresponding sinks without violating capacity constraints on the arcs. They apply this to real people (critically insured, minor insured, old aged people and pregnant women or with babies) and match them in respect to their according destinations (equipped hospitals, health centers, care centers and safe shelters). Therefore, if more than one commodity is entering an arc, then commodity with first priority enters first. Following that, if the flow of the first commodity is strictly less than the arc capacity, then commodity of the second priority enters and so on. So prioritization of sources automatically prioritize the sink in same order. Seliuchenko et al [31] proposed a model of the extended Multi-Commodity that allows to achieve compromise in optimal quality of service and overall network channel utilization. A new parameter called relative priority is introduced along with two extra parameters that considers users priority and sensitivity of the flow to the redirection degree.

Utility maximization is a major priority of energy consumers participating in peer-to-peer energy trading and sharing (P2P-ETS). Jogunola et al [32] introduced a Multi-Commodity algorithm with dual-optimization of energy and communication resources in P2P-ETS. This algorithm maximizes the global utility of consumers with fair resource allocation and minimizes energy generation cost and communication delay. The multi-robot path planning problem has been an area of interest for years. Araki et al [33] proposed a framework for path planning of the vehicles that can both fly and drive. It included two algorithms, one with priority planning with safe interval path planning and the other with Multi-Commodity Network Flow ILP, to accommodate multi-modal locomotion.

2.4 Real life Applications

Behrisch and Erdmann [34] used route assignment to apply it to a traffic situation in order to make it convenient for people to travel in an efficient manner. Al-Mayouf et al [35] also provided an improvement for traffic by making the routes more efficient and safer in Vehicular ad hoc networks (VANETS). In this real life example, there's a closer look on another type of transportation, railway systems. Whitman et al [36] emphasized the need for Multi-Commodity Network Flow models that response well to link closures and to finding an alternative path. They accomplish this by modifying the classic minimum-cost Multi-Commodity Flow optimization framework to relax the constraint that all demand must be met and to remove the cost criteria from the objective function. For their optimization model, they used Gurobi optimization

software. We decided to base our code model on Gurobi due to its ability to incorporate Multi-Commodity Flow Problems.

The last real life example is closely similar to our network model. The application is described by Nahabedian [19] for Video on demand (VOD) service, which is provided by most cable companies in which a user can request a video which is then downloaded directly to a home TV or PC. To formulate this as a minimum cost Multi-Commodity Flow Problem, the nodes of G will be the homes with VOD service, the servers storing the videos, and any intermediate relay nodes in the network. The edges of the network will be the cables connecting these nodes. The flow in this network is constantly changing as users request new videos. Every time a user requests a video, that user's home becomes a sink, and every home and server with that video becomes a source. The major difference however is that, the cable network has large available bandwidth, and hence, the capacities on the edges are very large.

The current literature has no foundations for theory grounded maximization of prioritized information flow and their impact on optimized attacks, which is developed in this research.

Chapter 3

Multi-commodity Flow Problem (MCFP)

Here, we introduce first the basics of Single-Commodity Flow Problems to then enable better understanding of Multi-commodity Flow Problems, which is the basis of our research.

3.1 Single-Commodity Flow Problem (SCFP)

Consider a network $N(V, E, s, t, C)$ as shown in Fig.3.1. The set of vertices V and a set of directed edges E with source s, delivering a commodity to a sink t. Associated with each edge E, is a capacity $C(E)$, which is a non-negative real number [19]. The objective is to deliver maximum amount of commodity from the source to sink while following flow conservation and capacity constraint.

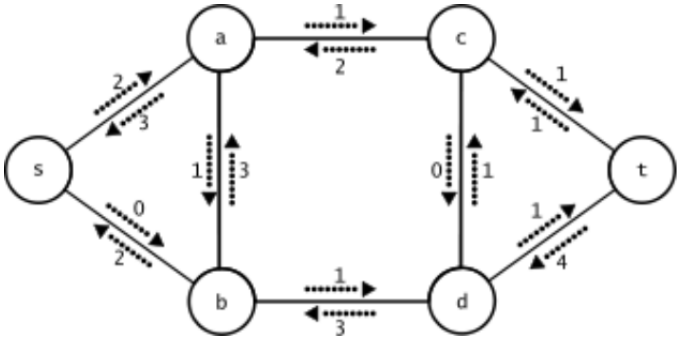


FIGURE 3.1: Single Commodity Flow Problem Network

In SCFP, conservation of flow means the total flow leaving a node must be equal to the total flow entering that node (for all nodes except the for the source and sink). The capacity constraint mandates that flow on any edge is feasible if it does not exceed the capacity of the edge. This problem can be formulated as a linear programming (LP) problem, maximizing a linear function subject to a set of linear constraints. However, some special features of the LP problem specific to the SCFP enabled a more efficient solution. Ford and Fulkerson proved an algorithm that satisfies the max-flow min-cut theorem [19]. Edmonds and Karp, and push-relabel are other algorithms that solve Single-Commodity Maximum Flow Problem [1]. Lastly,

the minimum cost flow's objective is to find a flow of a given value d (demand) from s to t , which has a minimum cost [19]. Therefore, costs can be included to minimize the product of cost and the flow.

3.2 The General MCFP Problem

The MCFP's goal is to find a feasible maximum flow from multiple commodities and/or multiple source and sink pairs. For each commodity there is a demand that needs to be transported from each source-sink pair [19]. In order to solve the MCFP, there are two necessary constraints. The first is the travel demand, which means that all the commodities need to be transported to their respective destinations. The flow conservation constraint is now modified for each commodity. This is because a node could be a source or a sink for one commodity, but be an intermediate for another one. The first constraint is essentially the sum of a set of Single-Commodity Flow Problems [16]. The second is the edge capacity constraint. This means that the flow on each edge can not exceed flow capacity. Multi-Commodity Flow Problem is defined as a given graph with a non-negative capacity. The graph consists of many commodity pairs k and there is a source and sink for each commodity (j). $f_j(u, v)$ is the flow from vertex u to v of commodity j. The aggregate flow $f(u, v)$ on edge (u, v) is the sum of all commodity flows, $f(u, v) = \sum_{j=1}^{k} f_j(u, v)$ and cannot exceed the capacity of edge (u, v) [1] as seen in fig. 3.2.

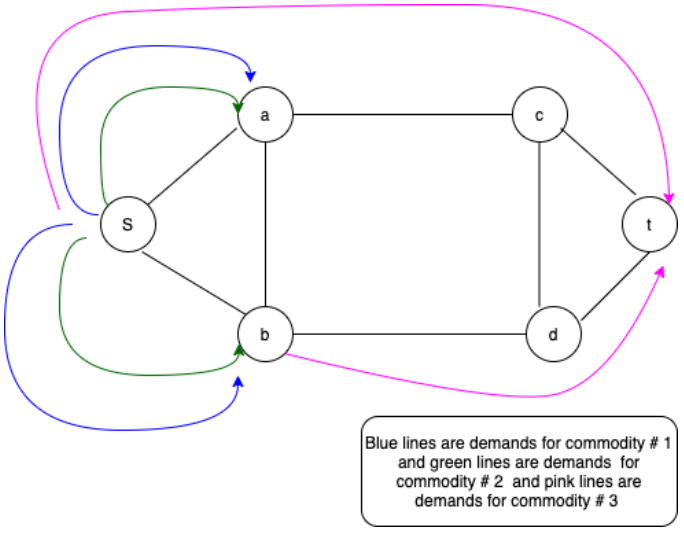


FIGURE 3.2: The multiple commodities using the same arc in a MCFP

In general, the Multi-Commodity Flow Problem, although may seem as simple as solving different Single-Commodity Flow Problems, in fact it is NP-complete. [19]. This is because of the interaction between commodities [16]. In general, MCFP are SCFP with bundle constraints,

tie together by arc capacity constraints [15].There are more reasons why MCFP is more complex. In SCFP, flows can cancel each other out if they are in opposite directions while with different commodities, they don't cancel each other out. For example, if the flow from an edge (u, v) from u to v is five units and one side has three units leaving then it would be five minus three, which is two units remaining. In SCFP, the max-flow min-cut theorem that guarantees that the max flow equals to the min cut, but such property doesn't hold for MCFP. Thirdly, there is total unimodularity of the constraint matrix in the SCFP-LP formulation guarantees that optimal integer-valued flows can be found by linear programming if the node supplies/demands and arc capacities are integers, but this integral property can not be extended to MCFP [15].

There are two different formulations of the LP, Node-arc formulation and Arc-chain formulation [19]. The node-arc form of the MCFP problem is derived from the simple SCFP formulation. In 1958, Ford and Fulkerson suggested that the arc-path form of the min-cost MCFP problem is based on the fact that any network flow solution can be broken up into path flows and cycle flows. Any arc flow vector can be expressed by simple path flows, assuming that no cycle has a negative cost. The node-arc form works for smaller size problems, but the arc-path form (w/ column generation techniques) works better for large-scale MCFP problems [15]. A pseudo-code for maximum Multi-Commodity Flow Problem can be found in [1].

3.3 Some Other Types of MCFP

Some other forms of MCFP are maximum MCFP, Max Concurrent Flow, and min-cost MCFP.

The Max Multi-Commodity Flow Problem: is a directed graph with capacities. There is a source-sink terminal pair of commodity j and there is an amount of flow between that pair. The objective is the maximize the sum of the flows for all commodities between their respective sources and destinations without exceeding the capacity of any edge.

The Max Concurrent Flow Problem: is also a directed graph with capacities. Here, every destination j has a demand $D(i, j)$ from source i. The network aims to meet a fraction z of the demands of *all* source destination pairs (i.e., throughput) and maximize Z [1, 2].

The Min-Cost MCFP: finds the flow assignment satisfying the demands of all commodities with minimum cost without violating the capacity constraints on all edges, with a cost associated with a unit flow on each edge [15].

Chapter 4

Problem Formulation and Theoretical Analysis

We consider a network $N(V, E, C)$ of n nodes and m directed edges. The set of all vertices of the network, N is $V = \{v_1, v_2, \cdots, v_n\}$. The network consists of a set of directed edges, $E = \{e_1, e_2, \cdots, e_m\}$, where edge e_k = (v_{i_k}, v_{j_k}) represents a directed edge from vertex, v_{i_k}, to vertex, v_{j_k}. $C : E \to \mathcal{R}^+$ is a capacity function, such that for any $e \in E$, $C(e)$, called the capacity of edge, e, is a non negative real number. Vertex v_j has a demand $D(i, j)$ from vertex v_i with priority $P(i, j)$. Due to the limited availability of resources, the network meets a fraction $Z(i, j)$, of the demand $D(i, j)$, ($0 \leq Z(i, j) \leq 1$, for all i, j) as seen in fig. 1.2(b).

Information in the network is maximized when the total prioritized flow in the network is maximum. In other words, when $\sum_{i,j} Z(i, j)D(i, j)h(i, j)$ is maximum. Therefore, we solve the problem:

$$\text{Maximize } \tilde{Z} = \max\left(\frac{\sum_{i,j} Z(i, j)D(i, j)h(i, j)}{\sum_{i,j} D(i, j)h(i, j)}\right) \tag{4.1}$$

subject to the constraints,

$$\sum_{i,j} f_{ij}(e) \leq C(e), \forall e \in E, \tag{4.2}$$

and

$$\forall j, \sum_k f_{jk}^{(i)} - \sum_l f_{lj}^{(i)} = \begin{cases} -Z(i, j)D(i, j) & j \neq i, \\ \sum_k Z(i, k)D(i, k) & j = i. \end{cases} \tag{4.3}$$

Constraint (4.2) is called the *capacity constraint*. The flow, $f_{ij}(e)$, represents all flows from vertex v_i, to vertex v_j, that flows through edge e. Thus, $\sum_{i,j} f_{ij}(e)$ represents *all flows* through edge e. The capacity constraint (4.2) therefore ensures that the flow on any edge *never* exceeds the capacity of the edge.

Constraint (4.3) is called the *flow conservation constraint*. In (4.3), $f_{jk}^{(i)}$ represent flow *originating* at vertex v_i, that leave vertex v_j, towards vertex v_k. Similarly, $f_{lj}^{(i)}$ represent flow *originating* at vertex v_i, that enters vertex v_l, from vertex v_k. The flow conservation constraint

(4.3) therefore ensures that nodes only take in those flows from any source, that are intended for them. They also ensure that nodes only add those flows in the network, that are being demanded of them. Otherwise, they just pass on the flows neither adding their own nor deleting/consuming them.

4.1 Theoretical Foundations for Throughput Enhancement

While linear programming simplex algorithms [37] can be used to solve (4.1) subject to (4.2) and (4.3) , the structure of the problem can be exploited to develop results that can provide more efficient algorithms. We first define a *distance function*(4.1), *shortest path* (4.2) and *diameter* (4.3) as in [2].

Definition 4.1. Let $d : E \rightarrow \mathcal{R}^+$ be a **distance function**, so that $\forall e \in E$, $d(e) \geq 0$. For any path P with edges $e_{i_1}, e_{i_2}, \cdots, e_{i_k}$, $d(p) \stackrel{\triangle}{=} \sum_{e \in P} d(e) = \sum_{j=1}^{K} d(e_{i_j})$.

Definition 4.2. Let P^*_{ij} be the path from vertex, v_i to vertex, v_j such that $d(P^*_{i,j}) \leq d(P_{ij})$ for all other paths, P_{ij} from vertex, v_i to vertex, v_j. Then $P^*_{i,j}$ is called the **shortest path** [38] from v_i to v_j and $\hat{d}(i,j) \stackrel{\triangle}{=} d(P^*_{i,j})$ is called the **minimum distance** [39] from v_i to v_j.

Definition 4.3. [38] In a network, $N(V, E, C)$, let $\hat{d}(i,j)$ be the minimum distance from vertex, v_i to vertex, v_j. Then the **diameter** of the network, N ,$dia(N)$, is defined as $dia(N) \stackrel{\triangle}{=} \max_{vi,vj,\in V} d(i,j)$.

We use Definitions 4.1-4.3 to propose the following theorem.

Theorem 4.4. ***Distance Upper Bound***

For a network, $N(V, E, C)$ with demand $D(i,j)$ for vertex v_j, from vertex v_i, having priority $h(i,j)$ and throughput $Z(i,j)$,

$$\tilde{Z} \leq \frac{\sum d(e)C(e)}{dia(N)\sum_{i,j} h(i,j)D(i,j)} \tag{4.4}$$

Proof. Let $\tilde{f}(p)$ be the prioritized flow on path P. Let $\mathbf{P}$ be the set of all paths from any pairs of vertices in V and $\mathbf{P}_e$ be the set of all paths containing edge e. Then,

$$\begin{aligned}
\textstyle\sum_{p\in\mathbf{P}} d(p)\tilde{f}(p) \leq \sum_{p\in\mathbf{P}} d(p)f(p) &= \textstyle\sum_{p\in\mathbf{P}}\sum_{e\in\mathbf{P}} d(e)f(p) \\
&= \textstyle\sum_{e\in\mathbf{E}}\sum_{p\in\mathbf{P_e}} d(e)f(p) \\
&= \textstyle\sum_{e\in\mathbf{E}} d(e)\sum_{p\in\mathbf{P_e}} f(p) \\
&= \textstyle\sum_{e\in\mathbf{E}} d(e)f(e) \\
&\leq \textstyle\sum_{e\in\mathbf{E}} d(e)c(e).
\end{aligned}$$

Let $\mathbf{P}_{ij}$ be the set of all paths from v_i to v_j. Then,

$$\begin{aligned}
\max\textstyle\sum_{p\in\mathbf{P}} d(p)\tilde{f}(p) &= \max\textstyle\sum_{i,j}\sum_{p\in\mathbf{P_{i,j}}} d(p)\tilde{f}(p) \\
&\geq \max\textstyle\sum_{i,j}\hat{d}(i,j)\sum_{p\in\mathbf{P_{i,j}}}\tilde{f}(p) \\
&= \max\textstyle\sum_{i,j}\hat{d}(i,j)f(i,j) \\
&= \max\textstyle\sum_{i,j}\hat{d}(i,j)f(i,j) \\
&= \max\textstyle\sum_{i,j}\hat{d}(i,j)Z(i,j)D(i,j)h(i,j) \\
&= dia(N)\textstyle\sum_{i,j} Z(i,j)D(i,j)h(i,j)
\end{aligned} \tag{4.5}$$

$$\text{Therefore, } \tilde{Z}\sum h(i,j)D(i,j)Z(i,j) \leq \frac{\sum_{e\in\mathbf{E}} d(e)c(e)}{dia(N)}, \tag{4.6}$$

$$\text{i.e, } \tilde{Z} \leq \frac{\sum d(e)C(e)}{dia(N)\sum_{i,j} h(i,j)D(i,j)} \tag{4.7}$$

□

We develop an alternate bound called cut upper bound theorem 4.9 for which we provide the following definitions:

Definition 4.5. [38] A **cut** from v_i to v_j, $(A_{i,j}, \overline{A}_{i,j})$ is defined as a partition of the vertex set, V, such that $v_i \in A_{i,j}$ and $v_j \in \overline{A}_{i,j}$, $A_{i,j} \subset V$, $\overline{A}_{i,j}$= $V \backslash A_{i,j}$.

Definition 4.6. [38] The **flow** of a cut $A_{i,j}, \overline{A}_{i,j}$, $f(A_{i,j}, \overline{A}_{i,j})$, is defined as

$$f(A_{i,j}, \overline{A}_{i,j}) \triangleq \sum_{\substack{e=(v_k,v_l)\in E \\ v_k\in A_{i,j}, v_l\in\overline{A}_{i,j}}} f(e). \tag{4.8}$$

Definition 4.7. [38] The **capacity** of a cut, $(A_{i,j}, \overline{A}_{i,j})$, $C(A_{i,j}, \overline{A}_{i,j})$, is defined as

$$C(A_{i,j}, \overline{A}_{i,j}) \triangleq \sum_{\substack{e=(v_k,v_l)\in E \\ v_k\in A_{i,j}, v_l\in \overline{A}_{i,j}}} C(e).$$

Theorem 4.8. ***Cut Upper Bound***

$$\tilde{Z} \leq \frac{\sum_{i,j} \min C(A_{i,j}, \overline{A}_{i,j}) h(i,j)}{\sum_{i,j} D(i,j) h(i,j)} \tag{4.9}$$

Proof.

$$\begin{aligned} \tilde{f}(P_{i,j}) &= Z(i,j)D(i,j)h(i,j) \\ &= f(A_{i,j}, \overline{A}_{i,j})h(i,j), \forall A(i,j) \\ &\leq C(A_{i,j}, \overline{A}_{i,j})h(i,j), \forall A(i,j). \end{aligned} \tag{4.10}$$

$$\begin{aligned} \max Z(i,j)D(i,j)h(i,j) &\leq \min_{A_{i,j}} C(A_{i,j}, \overline{A}_{i,j})h(i,j), \\ \max \textstyle\sum_{i,j} Z(i,j)D(i,j)h(i,j) &\leq \textstyle\sum_{i,j} \min C(A_{i,j}, \overline{A}_{i,j})h(i,j) \\ \text{i.e.,} \tilde{Z} &\leq \frac{\sum_{i,j} \min C(A_{i,j}, \overline{A}_{i,j})h(i,j)}{\sum_{i,j} D(i,j)h(i,j)}. \end{aligned} \tag{4.11}$$

□

Cut upper bound is similar to max-flow-min-cut theorem [37] for Single Commodity Flows. However for the MCFP, this bound may not be achieved. Developing an optimal algorithm to numerically evaluate our proposed algorithm is a challenging task. We use the off-the-shelf linear constrained optimization tool provided by Gurobi [40].

4.2 Security Enhancement

In this section, we discuss security aspects of the proposed prioritized demand based routing algorithm. For the following attack model, an attacker (or a set of colluding attackers) attack different source-destination pairs and the network defends these by its defense mechanisms [1].

The probability of successful attack on a source-destination pair k, p_k depends on the amount of attacking resources a_k and defending resources d_k, for the source-destination pair k. Here a_k could depend on the number of attackers, type of the attack and strength of the attack.

[1] Exact attack and defense mechanisms are beyond the scope of this research

Similarly, d_k depends on the strength and complexity of the security algorithm and the network's authentication and authorization mechanisms.

The interplay between a_k, d_k and p_k is explained as follows [41]. Let the ability of a drug to destroy a disease be X_1 and let X_2 denote the immunity parameter of the subject to the drug. Let the event Y=0 denote the survival of a subject when a drug is used on the subject and let the event Y=1 denote the death. Let X= $[X_1\ X_2]^T$. Let $\underline{\beta}$ = $[\beta_1\ -\beta_2]^T$ be the vector of regression parameters. The negative sign for β_2 indicates that the dose and the immunity act against each other. If $P_r\{Y$=1$\}$=1=p=1-$P_r\{Y$=0$\}$. Then according to the dose-response-immunity model,

$$\log it(p) = ln\Big(\frac{p}{1-p}\Big) = \underline{\beta}^T X. \tag{4.12}$$

Hence, from the dose-response immunity model the probability of successfully attacking the source-destination pair k, p_k, can be obtained from (4.12) as

$$p_k = \frac{a_k^{\beta_1}}{a_k^{\beta_1} + d_k^{\beta_2}}. \tag{4.13}$$

In general, the values β_1 and β_2 in (4.13) denote the attacker's capability and the network's defending capability. The net utility obtained by the attacker by attacking the flow from source-destination pair, k, which is also the net impact experienced by the network if the attacker attacks source- destination pair k, E_k, can be written as

$$E_k = U_k p_k = U_k\Big(\frac{a_k^{\beta_1}}{a_k^{\beta_1} + d_k^{\beta_2}}\Big), \tag{4.14}$$

where U_k = prioritized throughput obtained by source destination pair k, (i.e., $Z_k D_k h_k$). The expected impact on the network can then be written as

$$E = \sum_{k\in\mathcal{A}} E_k = \sum_{k\in\mathcal{A}} U_k\Big(\frac{a_k^{\beta_1}}{a_k^{\beta_1} + d_k^{\beta_2}}\Big), \tag{4.15}$$

where $\mathcal{A}$ is the set of source-destination pairs attacked by the attacker to assign defense resources d_k, for source destination pair k. The strategy for the attacker is allocating a_k amount of resources to attack source destination pair k. Mathematically, the strategy for the network is the vector, $\mathbf{d} = \begin{bmatrix} d_1 & d_2 & d_3 & \cdots & d_N \end{bmatrix}^T$ and that for the attacker is the vector, $\mathbf{a} = \begin{bmatrix} a_1 & a_2 & a_3 & \cdots & a_N \end{bmatrix}^T$. Note that the values of a_k are zero for $k \notin \mathcal{A}$. The optimal

strategy of the attacker is the vector $\mathbf{a}$ that solves the optimization problem

$$\max_{A,\mathcal{A}\subset N} \sum_{k\in\mathcal{A}} U_k\Big(\frac{a_k^{\beta_1}}{a_k^{\beta_1}+d_k^{\beta_2}}\Big), \tag{4.16}$$

subject to the constraints,

$$\sum_{k\in N} a_k \leq A, \tag{4.17}$$

where A is the total amount of attacking resources available to the attacker. The utility for the network is the negative of the attacker because the network loses in the form of impact, whatever the attacker gains by attacking source-destination pair k. In other words, the attacker and the network play a zero-sum game [42]. Thus, the optimal strategy for the network can be determined by solving the optimization problem.

$$\min_{\mathbf{d}} \max_{\mathbf{a},\mathcal{A}\subseteq N} \sum_{k\in\mathcal{A}} U_k\Big(\frac{a_k^{\beta_1}}{a_k^{\beta_1}+d_k^{\beta_2}}\Big), \tag{4.18}$$

subject to the constraints,

$$\sum_{i} d_k \leq D, \tag{4.19}$$

where D is the total amount of defending resources available to the network.

We develop Lemmas 4.9 and 4.10 to obtain optimal strategies for the network and the attacker.

Lemma 4.9. *The attacker obtains maximum utility only when it uses all the attacking resources, i.e., constraint (4.17) is met with equality.*

Proof. Consider a strategy $\tilde{\mathbf{a}} = \begin{bmatrix} \tilde{a}_1 & \tilde{a}_2 & \tilde{a}_3 & \cdots \tilde{a}_N \end{bmatrix}$, where for any chosen subset $\mathcal{A}$ of size s, $\tilde{A} = \sum_{k\in\mathcal{A}} \tilde{a}_k < A$. Let $\phi = A - \tilde{A}$. Note that $\phi > 0$. Let the utility obtained by the attacker under strategy $\tilde{\mathbf{a}}$ be $\tilde{U}$. Consider a strategy $\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \cdots \hat{a}_N \end{bmatrix}^T$, where, for any subset $\mathcal{A}$, $\hat{a}_k = \tilde{a_k} + \frac{\phi}{A}$, $\forall\ k \in \mathcal{A}$ and $\hat{a_{k'}} = 0$, $\forall k' \notin \mathcal{A}$. Note that $\sum_{k\in\mathcal{A}} \hat{a}_k = A$ and $\hat{a}_k > \tilde{a}_k$, $\forall\ k$ and hence,

$$\sum_{k\in\mathcal{A}} U_k\left(\frac{\hat{a}_k^{\beta_1}}{\hat{a}_k^{\beta_1}+d_k^{\beta_2}}\right) > \sum_{k\in\mathcal{A}} U_k\left(\frac{\tilde{a}_k^{\beta_1}}{\tilde{a}_k^{\beta_1}+d_k^{\beta_2}}\right).$$

Thus, $\tilde{\mathbf{a}}$ is a sub-optimal strategy □

Lemma 4.10. *The network perceives minimum impact only when it deploys all its defense resources, i. e., constraint (4.19) is met with equality.*

Proof. Consider a strategy $\tilde{\mathbf{d}} = \begin{bmatrix} \tilde{d}_1 & \tilde{d}_2 & \tilde{d}_3 & \cdots \tilde{d}_N \end{bmatrix}$, where for any chosen subset $\tilde{D} = \sum d_k \tilde{<} D$. Let $\chi = D - \tilde{D}$. Note that $\chi > 0$. Let the utility obtained by the attacker under strategy $\tilde{\mathbf{d}}$ be $\tilde{U}$.
Consider the strategy $\hat{\mathbf{d}} = \begin{bmatrix} \hat{d}_1 & \hat{d}_2 & \hat{d}_3 & \cdots \hat{d}_N & c \end{bmatrix}$, where, $\hat{d}_k = \tilde{d}_k + \frac{\chi}{N}$ and $\hat{d_{k'}} = 0$, $\forall k' \notin \mathcal{A}$ Note that $\sum \hat{d}_k = D$ and $\hat{d}_k > \tilde{d}_k, \forall\, k$ and hence,

$$\max_{\mathbf{a}} \sum_{k \in \mathcal{A}_s} U_k \left(\frac{a_k^{\beta_1}}{a_k^{\beta_1} + \hat{d}_k^{\beta_2}} \right) < \max_{\mathbf{a}} \sum_{k \in \mathcal{A}_s} U_k \left(\frac{a_k^{\beta_1}}{a_k^{\beta_1} + \tilde{d}_k^{\beta_2}} \right).$$

Thus, $\tilde{\mathbf{d}}$ is a sub-optimal strategy for the network. □

The following theorem provides a method for the attackers to decide which source- destination pairs to attack.

Theorem 4.11. *If $\exists \mathcal{A}_s \subset \mathcal{A}$ such that $\sum_{k \in \mathcal{A}_s} U_k > \sum \_ k' \notin \mathcal{A}_s U_{k'}$ then $a_{k'} = 0, \forall k' \notin \mathcal{A}_s$. That is, the attacker puts all attacking resources on only source-destination pairs that experience significant weighted throughput. In other words, the attacker does not waste resources on source-destination pairs that experience low priority and low throughput flows.*

Proof. Let the attacker deploy a strategy $\tilde{\mathbf{a}} = \begin{bmatrix} \tilde{a}_1 & \tilde{a}_2 & \tilde{a}_3 & \cdots \tilde{a}_N \end{bmatrix}^T$, where $\tilde{a}_{k'} > 0$, $k' \notin \mathcal{A}_s$. Let $\tilde{A} \triangleq \sum_{k' \notin \mathcal{A}_s} \tilde{a}_{k'}$. The utility obtained by the attacker under this strategy is

$$\tilde{E} = \sum_{k \in \mathcal{A}_s} \tilde{E}_k + \sum_{k' \notin \mathcal{A}_s} \tilde{E}_{k'} \leq \sum_{k \in \mathcal{A}_s} \tilde{E}_k + \sum_{k' \notin \mathcal{A}_s} U_{k'}.$$

Consider the strategy, $\hat{\mathbf{a}} = \begin{bmatrix} \hat{a}_1 & \hat{a}_2 & \hat{a}_3 & \cdots \hat{a}_N \end{bmatrix}^T$ where $\hat{a}_{k'} = 0 \; \forall \; k' \notin \mathcal{A}_s$ for some $i' \in \mathcal{A}_s$, $\hat{a}_i = \tilde{a}_i + \tilde{A}$ and $\forall\; i' \in \mathcal{A}_s$, $j' \neq i'$, $\hat{s}_{i'} = \tilde{s}_{i'}$. Thus the attacker deploys all its sensing resources and hence, obeys Lemma 4.9.

Let the impact on source-destination pair i according to this strategy be $\hat{E}_i$. Note that $\hat{E}_{i'} = \tilde{E}_{i'}, \forall\, i' \notin \mathcal{A}_s$ and $\hat{E}_i > \tilde{E}_i$. The utility obtained by the attacker for this strategy is

$$\begin{aligned} \hat{E} &= \sum_{\substack{k' \in \mathcal{A}_s \\ k' \neq k}} \hat{E}_{k'} + E_k \\ &> \sum_{\substack{k' \in \mathcal{A}_s \\ k' \neq j}} \tilde{E}_{k'} + \sum_{k \notin \mathcal{A}_s} U_k \\ &> \tilde{E}. \end{aligned} \tag{4.20}$$

In the above, the second step follows from the hypothesis while the third follows from (4.20). Thus $\tilde{\mathbf{a}}$ is a sub-optimal strategy for the attacker. □

Using Theorem 4.11, we develop the following theorem that helps the network plan its defense mechanism.

Theorem 4.12. *If the subset $\mathcal{A}_s$ is as defined in Theorem 4.11 then $d_{K'} = 0\, \forall\, K' \notin \mathcal{A}_s$ However if $d_{k'} = 0$ then $d_l^{\beta_2}$ is replaced by $a_l^{\beta_1} + (d_l + d_0)^{\beta_2}$, which is lesser than E_l in (4.22). Reducing the max min in (4.18), and hence a better strategy for the network. Theorem 4.12 indicates that the network will not waste its resources defending source-destination pairs that won't be attacked.*

$$E_l = \left(\frac{U_l a_l^{\beta_1}}{a_l^{\beta_1} + (d_l + d_0)^{\beta_2}} \right) \tag{4.21}$$

Proof. Since $a_{K'} = 0\, \forall\, K' \notin \mathcal{A}_s$ (Theorem 4.12, $E_{K'} = 0$ (from (4.14)). Let $d_{K'} = d_0 \neq 0$. Then $\exists\, l \in \mathcal{A}_s$ so that

$$E_l = \left(\frac{U_l a_l^{\beta_1}}{a_l^{\beta_1} + d_l^{\beta_2}} \right) \tag{4.22}$$

□

Another interpretation of Theorem 4.12 is that if the network is aware that the attacker has no incentive in attacking a particular source-destination pair, the network will not waste its resources defending that source-destination pair. The following theorem provides the proper allocation of attack and defense resources.

Theorem 4.13. *For all source-destination pairs, $k \in \mathcal{A}_s$, the attacker allocates attacking resource a_k, and the network allocates defense resource, d_k, so that*

$$\frac{a_k}{d_k} = \frac{a}{d}, \forall k. \tag{4.23}$$

Proof. Let the equilibrium strategies be $\underline{\mathbf{a}} = [\underline{a}_i]_{i\in\mathcal{A}_s}$ and $\underline{\mathbf{d}} = [\underline{d}_i]_{i\in\mathcal{A}_s}$ for the attacker and the network, respectively. Thus $\underline{\mathbf{a}}$ is a solution to the optimization problem (4.16) subject to the constraints (4.17) with $\mathbf{d} = \underline{\mathbf{d}}$. Similarly, $\underline{\mathbf{d}}$ is a solution to the optimization problem (4.18) subject to the constraints (4.19) with $\mathbf{a} = \underline{\mathbf{a}}$. From Lemmas 4.9 and 4.10, constraints (4.17) and (4.19) are met with equality. Thus, writing the Lagrangian for the equality constrained optimization problem specified by (4.16) and (4.17) and applying the first order necessary conditions, we obtain

$$\left(\frac{\underline{a}_i^{\beta_1} + \underline{d}_i^{\beta_2}}{\underline{a}_j^{\beta_1} + \underline{d}_j^{\beta_2}}\right)^2 = \frac{\underline{d}_i^{\beta_2}\underline{a}_i^{\beta_1-1}}{\underline{d}_j^{\beta_2}\underline{a}_j^{\beta_1-1}}, \forall i, j \in \mathcal{A}_s \tag{4.24}$$

Similarly, writing the Lagrangian for the equality constrained optimization problem specified by (4.18) and (4.19) and applying the first order necessary conditions,

$$\left(\frac{\underline{a}_i^{\beta_1} + \underline{d}_i^{\beta_2}}{\underline{a}_j^{\beta_1} + \underline{d}_j^{\beta_2}}\right)^2 = \frac{\underline{d}_i^{\beta_2-1}\underline{a}_i^{\beta_1}}{\underline{d}_j^{\beta_2-1}\underline{a}_j^{\beta_1}}, \forall i, j \in \mathcal{A}_s \tag{4.25}$$

From (4.24) and (4.25),

$$\frac{\underline{a}_i}{\underline{d}_i} = \frac{\underline{a}_j}{\underline{d}_j}, \forall i, j \in \mathcal{A}_s \tag{4.26}$$

Applying the constraints (4.17) and (4.19) and Lemmas 4.9 and 4.10 to (4.26), $\frac{\underline{a}_i}{\underline{d}_i} = \frac{A}{D}$, $\forall$ $i \in \mathcal{A}_s$. □

Theorem 4.13 suggest that the attack and defense resources are proportionally adjusted, for the chosen targeted source destination pairs. This may sound counter intuitive at the outset, because the attacker and the network are expected to put more resources on source-destination pairs with higher $Z_k D_k h_k$. However, Theorem 4.13 shows that once the sets of target source-destination pairs are decided (according to Theorem 4.11 and Theorem 4.12), the attacker and the network match each other's efforts and not concerned about the values of the prioritized throughput. This is because, just adding more resources does not ensure higher success for the attacker, if the network matches up its defense, according to (4.23), because the probability of successful attack on high prioritized throughput pairs is reduced, thereby reducing overall gain of the attacker.

Chapter 5

Results and Discussion

We now study the performance of the prioritized demand-based routing algorithm (Chapter 4.1) and its security enhancements (Chapter 4.2). The basic network and its related parameters are provided in Section 5.1 and experimental setup is described in Section 5.2. Numerical evaluations are presented in Sections 5.3-5.5.

5.1 Network and parameters

We consider a typical 30 nodes 5G D2D network [43]. A node is considered to be a source with 16% probability (i.e., about 5 nodes in a 30 node network) [44]. A node is a sink with 60% probability (i.e., about 18 nodes in a 30 node network) and the rest of the nodes are intermediate nodes [44]. We consider 5 types of priorities for the traffic based on the Internet Engineering Task Force (IETF) Differentiated Services (Diff Serv) [45]. These include broadband video service class (Netflix, live Zoom, etc), high throughput service class (like downloads), low latency service classes (like social media and online retail), telephony service class (constant bit rate like VoIP) and standard service class (other applications).

The priority for these service classes are taken to be based on Type of Service (ToS) values of the differentiated services code point (DSCP) [46]. Thus, broad band service video class gets a priority of 5, high throughput gets a priority of 4, telephony gets 3, low latency gets 2 and standard gets 1. To keep them between 0 and 1 we normalize by making 0 priority of 5 in a set of priorities $5, 4, 3, 2, 1$ as $5/(5+4+3+2+1) = \frac{1}{3}$. The percentage of these traffic is taken according to the 5G specifications [47] by which standard traffic forms 40%, low latency forms 20%, telephony and high throughput for 15% each and broadband video forms 10%.

We consider three scenarios:

1. **Uniform:** The prioritized demand is identical for all source-destination pairs.

2. **Dense:** Higher prioritized demands arise for traffic whose source is "dense" (i.e. have more emanating neighbors).

3. **Sparse:** Higher prioritized demands arise for traffic whose source is "sparse" (i.e. have less emanating neighbors).

5.2 Experimental Set Up

Gurobi [40] is a mathematical optimization solver used in conjunction with Python to obtain numerical results. It includes software modules to solve linear programs (linear objective functions with linear constraints, as in the flow maximization approach proposed in Chapter 4.1). We use the objective function in (4.1), subject to constraints, (4.2) and (4.3).

We use this tool to generate 100,000 topologies of random geometric graphs [44]. For the capacities, we use Okamura-Hata-distance model [48] for indoor environments that deploy orthogonal frequency division multiple access (OFDMA), which also translates to link capacities. For comparison we consider two other algorithms, "All or Nothing" [1] and Max Concurrent Flow [2] . In "All or Nothing", 100 % of prioritized demands are met for some traffic (ALL) and rest are completely blocked (NOTHING). In Max Concurrent Flow, there is an equal percentage on all demands, successfully met or provided. We present the numerical results in terms of throughput (Section 5.3), security impact (Section 5.4) and energy trade-off (Section 5.5).

5.3 Throughput Enhancement

Figure 5.1 [1] depicts the throughput performance for the scenarios when prioritized demands are the same from all sources (scenario **uniform** in Section 5.1), when there is more prioritized demand from dense sources (scenario **dense** in Section 5.1) and when there is more demand from sparse sources (scenario **sparse** in Section 5.1).

It is observed from Fig. 5.1(a) that the proposed algorithm and Max Concurrent Flow [2] perform almost identically for the **Uniform** scenario. This is because the prioritized demands are equal and hence, almost equal percentage of all demands will be met. The slight improvement of our algorithm arises from the fact that capacities of different links are different. The "All or Nothing" [1] suffers from low throughputs because many source-destination pairs are blocked, resulting in a low average.

[1] In all figures in this chapter, "All or Nothing" represents meeting 100% of prioritized demand or completely blocking [1]. Max Concurrent Flow [2] meets equal percentage of all demands.

In the **dense** scenario as shown in Fig. 5.1(b), the improvement provided by the proposed algorithm is more significant compared to the Max Concurrent Flow [2]. For instance, the Max Concurrent Flow [2] gives less than 60%, but the proposed algorithm yields 75%. That is about a 25% enhancement ($\frac{75-60}{60} \times 100 = 25\%$). The enhancement is more than 50% compared to "All or Nothing"[1] (increases from 48% to 75%).

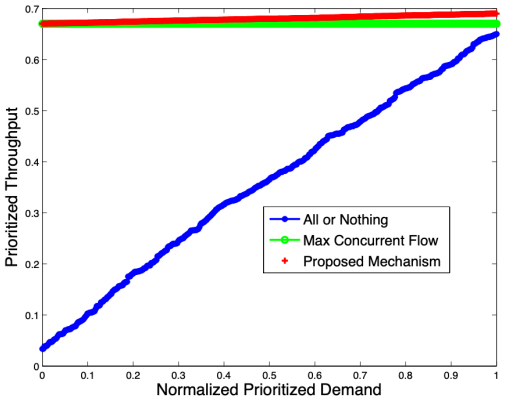
The improvement is even more significant for the **sparse** scenario, as observed from Fig. 5.1(c). Max Concurrent Flow [2] is giving 70% compared to our proposed algorithms that yields 90% (i.e., for an enhancement of 28%). Compared to "All or Nothing" [1], our algorithm provides $3\times$ the throughput. In **sparse** scenario, there are fewer neighbors to each source. Therefore there are fewer alternate routes to send the data. This shows that the proposed algorithm is even more effective when fewer paths are available from the sources. In the next subsection, we discuss the security enhancement emerging out of the mechanisms proposed in Section 4.2.

5.4 Security Performance

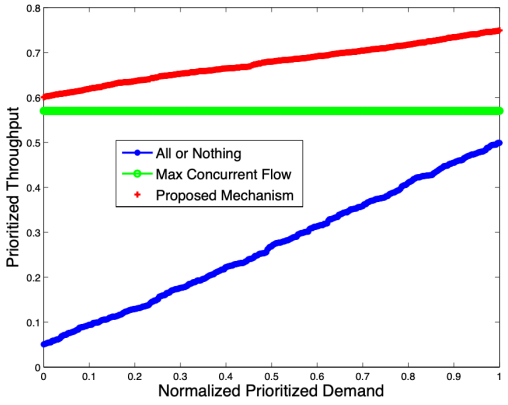
For evaluating the security performance, we consider three scenarios:

1. **Equality:** when the attacker and the network are equally capable (i.e., $\beta_1 = \beta_2$).
2. **Powerful Attacker:** when the attacker has more capability (i.e., $\beta_1 > \beta_2$; we take $\beta_1 = 2\beta_2$).
3. **Weak Attacker:** when the attacker has half the capability (i.e., $\beta_1 < \beta_2$; we take $\beta_1 = \frac{1}{2}\beta_2$).

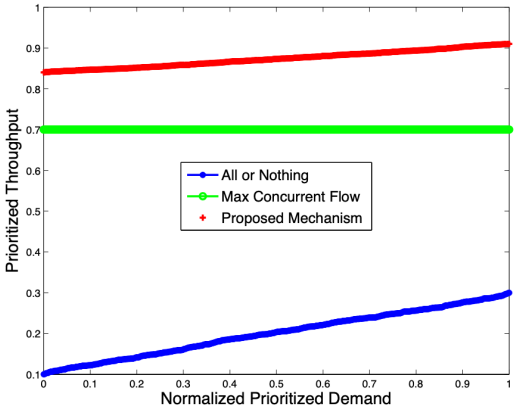
For the sake of fair comparison, we take the amount of available attacking resources, A = the amount of available defending resources, D. The purpose is to study the performance mainly due to the capabilities of the attacker and of the network, as against resources, which can be easily equalized.



(a) **Uniform**

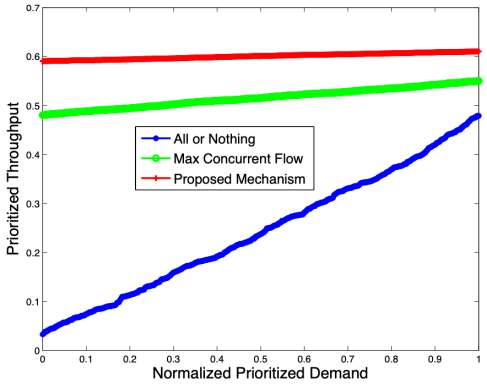


(b) **Dense**

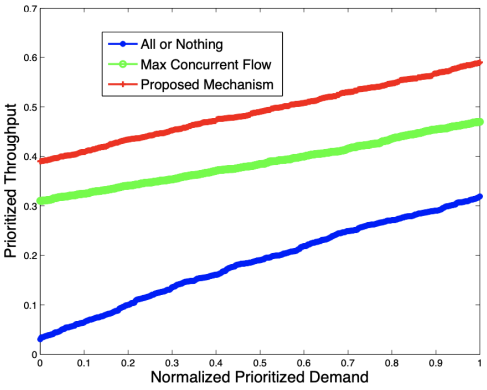


(c) **Sparse**

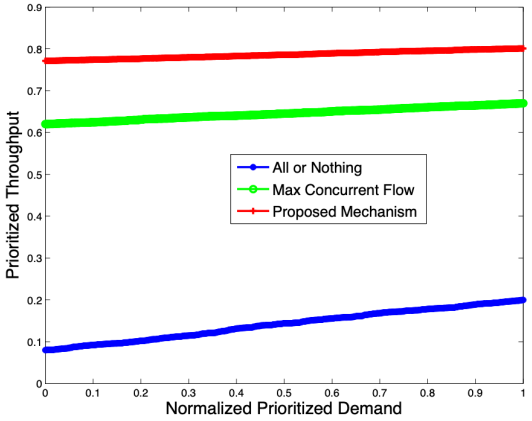
FIGURE 5.1: **Throughput** Performance provided by the proposed algorithm, "All or Nothing" [1] and Max Concurrent Flow [2]. For the **uniform nodes** (Fig. 5.1(a)), the proposed algorithm yields an approximately same result when compared to Max Concurrent Flow [2]. For the **dense nodes** (Fig. 5.1(b)), the proposed algorithm yields 25% enhancement compared to Max Concurrent Flow [2] (increase from 60% to 75%) and more than 50% enhancement (increase from 48% to 75%) compared to "All or Nothing" [1]. For **sparse nodes** (Fig. 5.1(c)), the enhancement is 28% compared to the Max Concurrent Flow and 3× compared to "All or Nothing" [1]. The cut upper bound (Theorem 4.8) yielded 6% (**dense**) to 8% (**sparse**) more than the throughput provided by the proposed algorithm.



(a) **Uniform**



(b) **Dense**



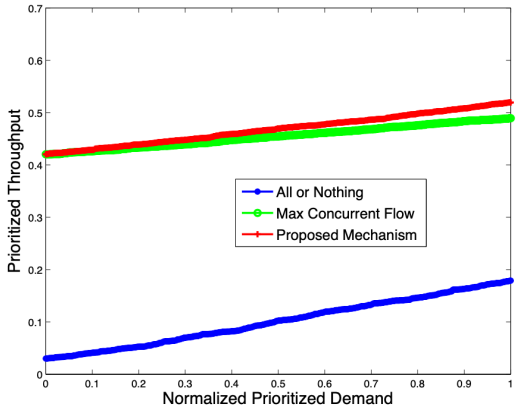
(c) **Sparse**

FIGURE 5.2: **Security** Performance for when the attacker and network are equally capable. For the **uniform** scenario (Fig. 5.2(a)), the proposed algorithm provides 9% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 55% to 60%) and 25% compared to "All or Nothing" [1] (increase from 48% to 60%). For the **dense** scenario (Fig. 5.2(b)), proposed algorithm again yields 25% enhancement in prioritized throughput compared to the Max Concurrent Flow [2] (increase from 48% to 60%) and 2× compared to "All or Nothing" [1] (increase from 30% to 60%). For **sparse** scenario (Fig. 5.2(c)), our algorithm yields 18% enhancement compared to the Max Concurrent Flow [2] (increase from 68% to 80%) and 4× the throughput of "All or Nothing" [1] (increase from 20% to 80%).

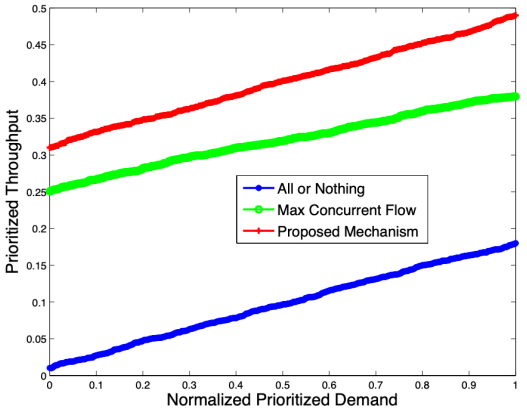
Fig. 5.2 demonstrates the **Security** Performance for when the attacker and network are equally capable (i.e., **equal** scenario mentioned in this section). For the **uniform** scenario (Fig. 5.2(a)), the proposed algorithm provides 9% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 55% to 60%) and 22% compared to "All or Nothing" [1] (increase from 48% to 60%). For the **dense nodes** (Fig. 5.2(b)), the proposed algorithm again yields 20% enhancement in prioritized throughput compared to the Max Concurrent Flow [2] (increase from 56% to 61%) and it's 2× compared to "All or Nothing" [1] (increase from 48% to 61%). For **sparse nodes** (Fig. 5.2(c)), our algorithm yields 18% enhancement compared to the Max Concurrent Flow [2] (increase from 68% to 80%) and 4× the throughput of "All or Nothing" [1] (increase from 20% to 80%).

In Fig. 5.3, the attacker is more capable than the network (**powerful attacker** scenario, mentioned in this section). The **uniform** scenario (Fig. 5.3(a)) for our proposed algorithm provides an 11% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 45% to 50%) and it's 3× compared to the "All or Nothing" [1] (increase from 19% to 50%). The **dense** scenario (Fig. 5.3(b)) shows a 30% enhancement for our proposed algorithm in compared to Max Concurrent Flow [2] (increase from 37% to 48%). Our proposed algorithm is 2.5× compared to "All or Nothing" [1] (increase from 19% to 48%). Lastly, the **sparse** scenario (Fig. 5.3(c)) is 17% compared to the Max Concurrent Flow [2] (increase from 60% to 70%) and 4× compared to the "All or Nothing" [1] (increase from 17% to 70%).

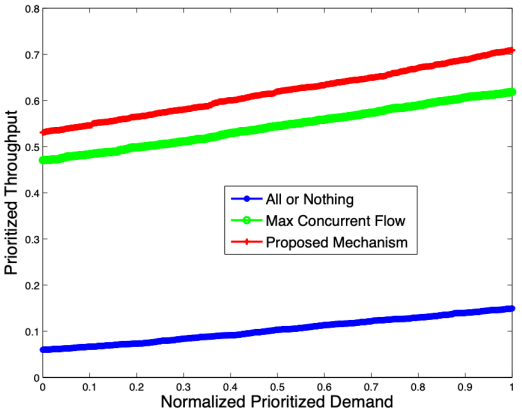
In Fig. 5.4, the attacker is less capable than the network (**weak attacker** scenario, mentioned in this section). The **uniform** scenario (Fig. 5.4(a)) for our proposed algorithm provides an 7% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 60% to 64%) and 28% compared to the "All or Nothing" [1] (increase from 50% to 64%). The **dense** scenario (Fig. 5.4(b)) shows a 11% enhancement for our proposed algorithm in compared to Max Concurrent Flow [2] (increase from 54% to 60%).Our proposed algorithm is 28% compared to "All or Nothing" [1] (increase from 47% to 60%). Lastly, the **sparse** scenario (Fig. 5.4(c)) is 18% compared to the Max Concurrent Flow [2] (increase from 68% to 80%) and 4× compared to the "All or Nothing" [1] (increase from 19% to 80%).



(a) **Uniform**



(b) **Dense**



(c) **Sparse**

FIGURE 5.3: **Powerful Attacker** Security Performance (i.e., $\beta_1 > \beta_2$; we take $\beta_1 = 2\beta_2$), the attacker is more capable than the network (**powerful attacker** scenario, mentioned in this section). The **uniform** scenario (Fig. 5.3(a)) for our proposed algorithm provides an 11% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 45% to 50%) and 3× compared to the "All or Nothing" [1] (increase from 19% to 50%).The **dense** scenario (Fig. 5.3(b)) shows a 30% enhancement for our proposed algorithm in compared to Max Concurrent Flow [2] (increase from 37% to 48%). Our proposed algorithm is 2.5× compared to "All or Nothing" [1] (increase from 19% to 48%). Lastly, the **sparse** scenario (Fig. 5.3(c)) is 17% enhanced compared to the Max Concurrent Flow [2] (increase from 60% to 70%) and 4× compared to the "All or Nothing" [1] (increase from 17% to 70%).

Comparing the results in the case where there is no attack (Fig. 5.1) to those with attacks in Fig. 5.2 - Fig. 5.4, it is observed that the prioritized throughput decreases in the presence of attacks.This is expected because attacks reduce the networks performance.

Uniform scenario: The prioritized throughput in our algorithm reduces from 70% (Fig. 5.1(a)) to 55% (Fig. 5.2(a)) when the attacker and network are equally capable (**equality** scenario). For the **powerful attacker** scenario, the prioritized throughput further decreases from 70% (Fig. 5.1(a)) to 50% (Fig. 5.3(a)) when the attacker has more capability. For the **weak attacker** scenario when the attacker has less capability, the prioritized throughput's decrease is less than 65% (Fig. 5.4(a)), but still not the same as in (Fig. 5.1(a)).

Dense scenario: The prioritized throughput in our algorithm reduces from 75% (Fig. 5.1(b)) to 60% (Fig. 5.2(b)) when the attacker and network are equally capable (**equality** scenario). For the **powerful attacker** scenario, the prioritized throughput further decreases from 75% (Fig. 5.1(b)) to 48% (Fig. 5.3(b)) when the attacker has more capability. For the **weak attacker** scenario when the attacker has less capability, the prioritized throughput's decrease is less than 58% (Fig. 5.4(b)), but still not the same as in (Fig. 5.1(b)).

Sparse scenario: The prioritized throughput in our algorithm reduces from 90% (Fig. 5.1(c)) to 79% (Fig. 5.2(b)) when the attacker and network are equally capable (**equality** scenario). For the **powerful attacker** scenario, the prioritized throughput further decreases from 90% (Fig. 5.1(c)) to 70% (Fig. 5.3(c)) when the attacker has more capability. For the **weak attacker** scenario when the attacker has less capability, the prioritized throughput's 90% (Fig. 5.4(b)), which is the same in (Fig. 5.1(c)).

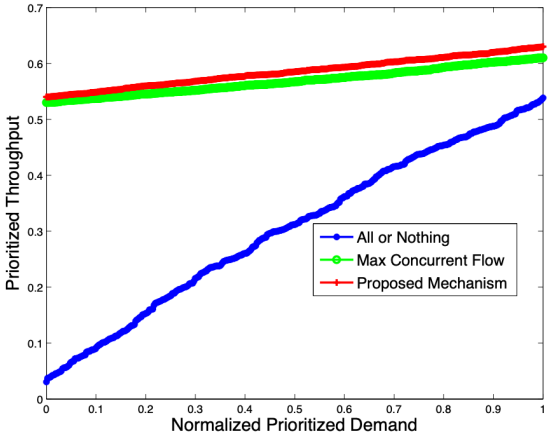
Essentially, numerical results indicate that the proposed algorithm provides significant enhancement to the prioritized throughput when network resources become less available. The throughput enhancement is even more significant in the presence of optimized attacks. This shows that **our proposed algorithm is even more effective under hostile network conditions**.

While Figs. 5.1-5.4 showed the enhancement in the prioritized throughout yielded by our proposed algorithm, we posed a question, "*What is the trade-off or penalty imposed on the network, because of this improvement?*" To answer this question, we study the average energy consumed by the devices, in the following section.

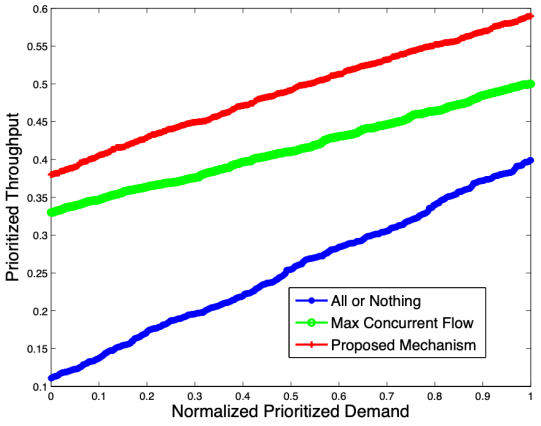
5.5 Energy Trade-off

Fig. 5.5 shows the energy consumption when there is no attack. To calculate energy, we once again used the Okamura-Hata model [48]. For the **uniform** scenario in Fig. 5.5(a), all three algorithms are almost equal energy. The proposed algorithm yields an enhancement of 3% when compared to Max Concurrent Flow (increase from 80% to 82%) and 6% enhancement compared to "All or Nothing" [1] (increase from 77% to 82%). This shows that our algorithm enhances

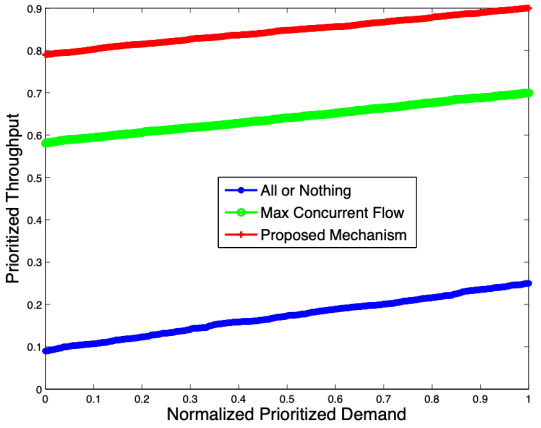
throughput significantly without consuming extra energy. For the **dense** scenario in Fig. 5.5(b), the proposed algorithm yields 7% enhancement compared to Max Concurrent Flow [2] (increase from 75% to 80%) and 14% enhancement compared to "All or Nothing" [1] (increase from 70% to 80%). For **sparse** scenario in Fig. 5.5(c), the enhancement is 23% compared to the Max Concurrent Flow [2] (increase from 70% to 86%) and 43% compared to "All or nothing" [1] (increase from 60% to 86%). Under attack it was observed that all mechanisms consumed almost equal energy. This is because, the attacks reduced the absolute values of the throughput for the proposed as well as existing mechanisms. For the reduced throughputs, the energy consumption from the Okamura-Hata model provided almost identical values.



(a) **Uniform**

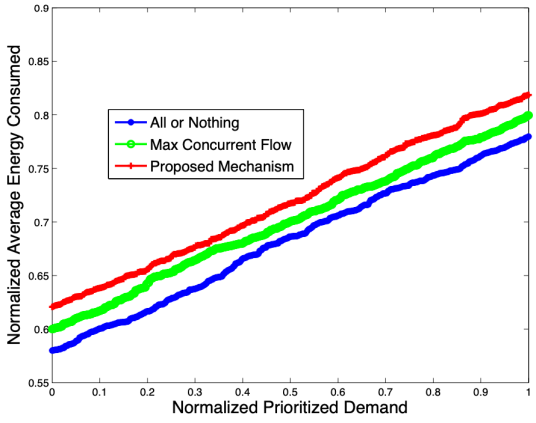


(b) **Dense**

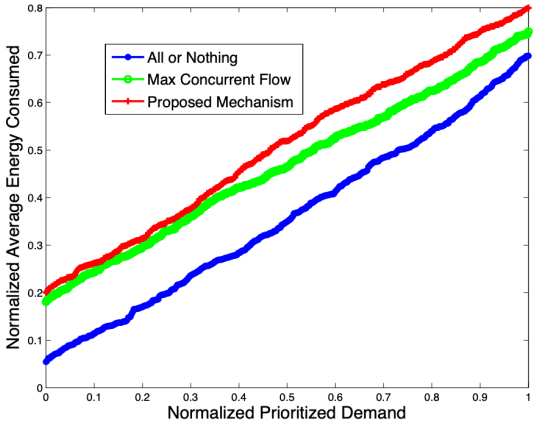


(c) **Sparse**

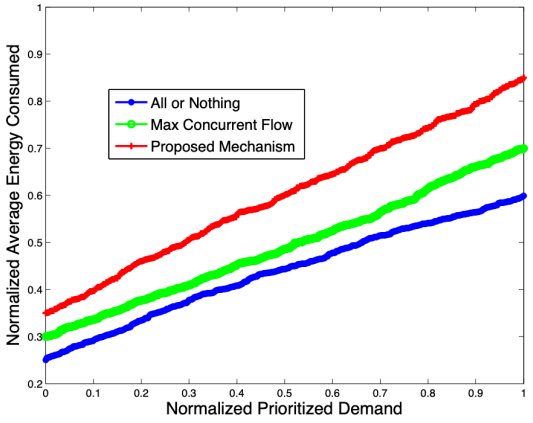
FIGURE 5.4: **Weak Attacker** Security Performance (i.e., $\beta_1 < \beta_2$; we take $\beta_1 = \frac{1}{2}\beta_2$), the attacker is less capable than the network (**weak attacker** scenario, mentioned in this section).The **uniform** scenario (Fig. 5.4(a)) for our proposed algorithm provides an 7% enhancement in prioritized throughput compared to Max Concurrent Flow [2] (increase from 60% to 64%) and 28% compared to the "All or Nothing" [1] (increase from 50% to 64%).The **dense** scenario (Fig. 5.4(b)) shows a 11% enhancement for our proposed algorithm in compared to Max Concurrent Flow [2] (increase from 54% to 60%).Our proposed algorithm is 28% compared to "All or Nothing" [1] (increase from 47% to 60%). Lastly, the **sparse** scenario (Fig. 5.4(c)) is 18% enhanced compared to the Max Concurrent Flow [2] (increase from 68% to 80%) and $4\times$ compared to the "All or Nothing" [1] (increase from 19% to 80%).



(a) **Uniform**



(b) **Dense**



(c) **Sparse**

FIGURE 5.5: **Energy** Performance provided by the proposed algorithm, "All or Nothing" [1] and Max Concurrent Flow [2]. For the **uniform** scenario, the proposed algorithm yields an enhancement of 3% when compared to Max Concurrent Flow (increase from 80% to 82%) and 6% enhancement compared to "All or Nothing" [1] (increase from 77% to 82%). For the **dense** scenario, the proposed algorithm yields 7% enhancement compared to Max Concurrent Flow [2] (increase from 75% to 80%) and 14% enhancement compared to "All or Nothing" [1] (increase from 70% to 80%). For **sparse** scenario, the enhancement is 23% compared to the Max Concurrent Flow [2] (increase from 70% to 86%) and 43% compared to "All or nothing" [1] (increase from 60% to 86%).

Chapter 6
Conclusion

We proposed a multi-commodity flow based secure and reliable routing algorithm for multi-hop D2D networks. Varying demands and priorities for different source-destination pairs were taken into account to develop a prioritized demand based routing algorithm. We derived cut and distance upper bounds to determine the maximum achievable prioritized throughput. We then investigated the scenario when a set of colluding attackers choose to attack one or more of the source-destination pairs and the network expended defence resources to counter the attack. This attack defense scenario was modelled as a min-max optimization problem and the optimal strategies for the attacker and the network were obtained as the saddle point of the problem. We showed that the attackers and network expended their sources in proportion only to each other and not based on the actual demand or the throughput.

Numerical evaluation showed that the proposed algorithm achieved as close to 92 to 94% of the maximum achievable throughput dictated by the theoretical bounds we derived. Comparison with existing algorithms showed that when routes were limited, the proposed algorithm achieved 28% to $3\times$ enhancement to the throughput. Under hostile environments (i.e., in the presence of optimized attacks), the proposed algorithm yielded 25% to $4\times$ enhancement to the throughput when the attacker and the network were equally capable. When the attacker was more powerful than the network, the throughput enhancement was 30% to $4\times$. When the network was more powerful, the proposed algorithm provided identical performance enhancement as if the attack didn't exist. Essentially, numerical results indicated that **our proposed algorithm is even more effective under hostile network conditions**.

The optimization problems developed in this project can be applied to any targeted network attack scenario. For example, in the Amazon Web Services (AWS) using containerization [49], our security model can be used to identify containers most likely to be attacked by an attacker and enhance their security accordingly. Our multi-commodity flow algorithm can be adopted to determine the amount of data storage in different containers.

Future enhancements to our research include improving the distance upper bound to incorporate energy efficiency. Another extension of our project is the development of reduced complexity polynomial time algorithms to exploit the nature of the problem.

Bibliography

[1] S. CHIKA, "Using maximum and maximum concurrent multicommodity flows for qos routing in the internet," pp. 1–37, July 2017. [Online]. Available: https://shareok.org/handle/11244/8132

[2] F. Shahrokhi and D. W. Matula, "The maximum concurrent flow problem," *J. ACM*, vol. 37, no. 2, p. 318–334, April 1990. [Online]. Available: https://doi.org/10.1145/77600.77620

[3] N. G., "How many iot devices are there in 2021? [all you need to know]," August 2021. [Online]. Available: https://techjury.net/blog/how-many-iot-devices-are-there/#gref

[4] J. Park, "Fast and energy efficient multihop d2d routing scheme," *International Journal of Distributed Sensor Networks*, vol. 12, no. 5, pp. 1–9, May 2016.

[5] J. Casella, "An analysis of flow-based routing," pp. 4–76, 2011. [Online]. Available: https://scholarworks.rit.edu/theses/533

[6] Y. Zhang, Y. Yang, and L. Dai, "Energy efficiency maximization for device-to-device communication underlaying cellular networks on multiple bands," *IEEE Access*, vol. 4, pp. 7682–7691, 2016.

[7] F. Rebecchi, L. Valerio, R. Bruno, V. Conan, M. D. de Amorim, and A. Passarella, "A joint multicast/d2d learning-based approach to lte traffic offloading," *Elsevier Computer Communications*, vol. 72, pp. 26–37, 2015.

[8] "Facebook." [Online]. Available: https://www.facebook.com/

[9] "Google." [Online]. Available: https://store.google.com/product/chromecast?hl=en-US

[10] M. Alnakhli, S. Anand, and R. Chandramouli, "Joint spectrum and energy efficiency in device to device communication enabled wireless networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 2, pp. 217–225, 2017.

[11] M.Archana, T.Dineshkumar, and V.Uma, "Manets for 5g communication networking," *International Journal of Engineering Research and Technology (IJERT)*, vol. 8, 2020.

[12] S. Corson and J. Macker, "Rfc2501: Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," 1999.

[13] P. Shailaja, C. G. Rao, and A.Nagaraju, "A parametric oriented research on routing algorithms in mobile adhoc networks," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 1, pp. 4116–4126, Nov. 2019.

[14] K. Salimifard and S. Bigharaz, "The multi-commodity network flow problem: State of the art classification, applications and solutions methods," *International Journal on Operations Research*, vol. 8, April 2020. [Online]. Available: https://doi.org/10.1007/s12351-020-00564-8

[15] I.-L. Wang, "Multicommodity network flows: A survey, part i:applications and formulations," *International Journal of Operations Research*, vol. 15, no. 4, pp. 145–153, Dec. 2018.

[16] W. DAI, J. ZHANG, and X. SUN, "On solving multi-commodity flow problems: An experimental evaluation," *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1481–1492, Aug. 2017.

[17] D. Khanal, U. Pyakurel, and T. Dhamala, "Prioritized multi-commodity flow model and algorithm," 12 2020.

[18] C. Chekhuri, S. Khanna, and F. B. Shepherd, "The All-or-Nothing multicommodity flow problem," *ACM Symposium on Theory of Computing*, June 2004.

[19] A. Nahabedian, "A primal-dual approximation algorithm for the concurrent flow problem," pp. 1–41, April 2010. [Online]. Available: https://web.wpi.edu/Pubs/ETD/Available/etd-042910-160853/unrestricted/nahabedian.pdf

[20] A. Madry, "Faster approximation schemes for fractional multicommodity flow problems via dynamic graph algorithms," in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, ser. STOC '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 121–130. [Online]. Available: https://doi.org/10.1145/1806689.1806708

[21] W. Dai, X. Sun, and S. Wandelt, "Finding feasible solutions for multi-commodity flow problems," in *2016 35th Chinese Control Conference (CCC)*, 2016, pp. 2878–2883.

[22] N. Farrugia, J. A. Briffa, and V. Buttigieg, "An evolutionary multipath routing algorithm using sdn," in *2018 9th International Conference on the Network of the Future (NOF)*, 2018, pp. 1–8.

[23] ——, "Solving the multi-commodity flow problem using a multi-objective genetic algorithm," in *2019 IEEE Congress on Evolutionary Computation (CEC)*, 2019, pp. 2816–2823.

[24] C. Cheng, "A approximation algorithm to solve the maximum multicommodity flow problem with local dominant," in *2019 3rd International Conference on Electronic Information Technology and Computer Engineering (EITCE)*, 2019, pp. 988–992.

[25] M. Zhang, W. Yang, S. Gao, and W. Xu, "Network energy-saving adjustment routing under changing demands: Models and algorithms," *IEEE Access*, vol. 8, pp. 90 676–90 685, 2020.

[26] R. M. Curry, "Mathematical models and algorithms for network flow problems arising in wireless sensor network applications," pp. 1–51, Aug. 2018. [Online]. Available: https://tigerprints.clemson.edu/all_dissertations/2226

[27] X. Lu, F. Kong, X. Liu, J. Yin, Q. Xiang, and H. Yu, "Bulk savings for bulk transfers: Minimizing the energy-cost for geo-distributed data centers," *IEEE Transactions on Cloud Computing*, vol. 8, no. 1, pp. 73–85, 2020.

[28] M. H. H. Schoot Uiterkamp, G. Hoogsteen, M. E. T. Gerards, J. L. Hurink, and G. J. M. Smit, "Multi-commodity support in profile steering," in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, 2017, pp. 1–6.

[29] J. Zhang, X. Zhang, and M. Sun, "Two-level decomposition for multi-commodity multicast traffic engineering," in *2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, 2017, pp. 1–2.

[30] A. Samani and M. Wang, "Maxstream: Sdn-based flow maximization for video streaming with qos enhancement," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN)*, 2018, pp. 287–290.

[31] M. Seliuchenko, O. Lavriv, O. Panchenko, and V. Pashkevych, "Enhanced multi-commodity flow model for qos-aware routing in sdn," in *2016 International Conference Radio Electronics Info Communications (UkrMiCo)*, 2016, pp. 1–3.

[32] O. Jogunola, B. Adebisi, K. Anoh, A. Ikpehai, M. Hammoudeh, and G. Harris, "Multi-commodity optimisation of peer-to-peer energy trading resources in smart grid," *Journal of Modern Power Systems and Clean Energy*, pp. 1–13, 2021.

[33] B. Araki, J. Strang, S. Pohorecky, C. Qiu, T. Naegeli, and D. Rus, "Multi-robot path planning for a swarm of robots that can both fly and drive," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 5575–5582.

[34] M. Behrisch and J. Erdmann, "Route estimation based on network flow maximization," *EPiC Series in Engineering*, vol. 2, pp. 173–182, May 2018. [Online]. Available: https://elib.dlr.de/120854/

[35] Y. R. B. Al-Mayouf, M. Ismail, N. F. Abdullah, A. W. A. Wahab, O. A. Mahdi, S. Khan, and K.-K. R. Choo, "Efficient and stable routing algorithm based on user mobility and node density in urban vehicular network," *PLOS ONE*, pp. 1–24, Nov. 2016.

[36] M. G. Whitman, K. Barker, J. Johansson, and M. Darayi, "Component importance for multi-commodity networks," *Comput. Ind. Eng.*, vol. 112, no. C, p. 274–288, Oct. 2017. [Online]. Available: https://doi.org/10.1016/j.cie.2017.08.004

[37] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[38] R. k. Ahuja, T. L. Magnanti, and J. B. Orlin, *Network Flows*. Pearson, 1993. [Online]. Available: http://cs.yazd.ac.ir/hasheminezhad/STSCS4R1.pdf

[39] R. Trudeau, *Introduction to Graph Theory*, ser. Dover Books on Mathematics. Dover Pub., 1993. [Online]. Available: https://books.google.com/books?id=NunuAAAAMAAJ

[40] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2021. [Online]. Available: https://www.gurobi.com

[41] S. Anand, S. Sengupta, and R. Chandramouli, "An attack-defense game theoretic analysis of multi-band wireless covert timing networks," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.

[42] H. Gintis, "Game theory evolving, second edition a problem-centered introduction to modeling strategic interaction," *Game Theory Evolving: A Problem-Centered Introduction to Modeling Strategic Interaction Second Edition*, 01 2009.

[43] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5g wireless networks: A comprehensive survey," *IEEE Communications Surveys and Tutorials*, vol. 18, pp. 1–1, 02 2016.

[44] B. Bollobás, *Random graphs*, 2nd ed., ser. Cambridge studies in advanced mathematics. Cambridge University Press, 2001, no. 73.

[45] F. Baker, J. Babiarz, and K. H. Chan, "Configuration Guidelines for DiffServ Service Classes," RFC 4594, Aug. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4594.txt

[46] C. Publishing, "The DiffServ model, differentiated services code point (DSCP), per-hop behavior (PHB) (classification, marking, and NBAR)." [Online]. Available: http://what-when-how.com/ccnp-ont-exam-certification-guide\/the-diffserv-model-differentiated-services-code-point-dscp-and\-per-hop-behavior-phb-classification-marking-and-nbar/

[47] S. G, F. D, and V. P, "Towards traffic identification and modeling for 5G application use-cases," *MDPI Electronics*, vol. 9, no. 4, Apr. 2020.

[48] T. Rappaport, *Wireless communications: Principles and practice*, 2nd ed., ser. Prentice Hall communications engineering and emerging technologies series. Prentice Hall, 2002, includes bibliographical references and index.

[49] [Online]. Available: https://www.datavail.com/solutions/app-development-integration/containerization-services/