TOWSON UNIVERSITY
OFFICE OF GRADUATE STUDIES

A Bloom Filter Based Dual-Layer Routing Scheme in Large-Scale Mobile Networks

by

Weichao Gao

A Thesis

Presented to the faculty of

Towson University in partial fulfillment

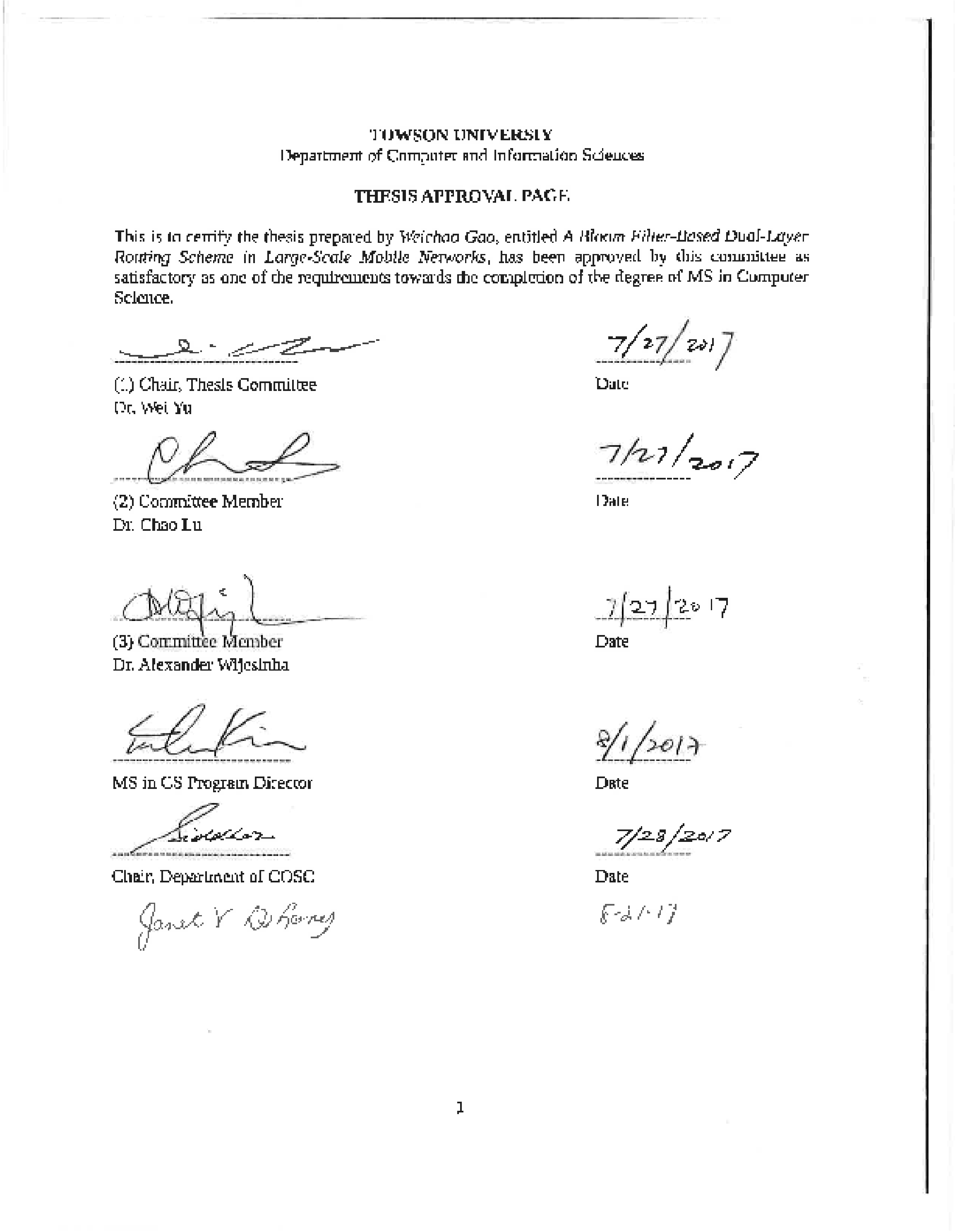
of the requirements for the degree

Master of Science

Department of Computer and Information Sciences

Towson University Towson, Maryland 21252

(August, 2017)



TOWSON UNIVERSITY
Department of Computer and Information Sciences

THESIS APPROVAL PAGE

This is to certify the thesis prepared by *Weichao Gao*, entitled *A Bloom Filter-Based Dual-Layer Routing Scheme in Large-Scale Mobile Networks*, has been approved by this committee as satisfactory as one of the requirements towards the completion of the degree of MS in Computer Science.

(1) Chair, Thesis Committee
Dr. Wei Yu

7/27/2017
Date

(2) Committee Member
Dr. Chao Lu

7/27/2017
Date

(3) Committee Member
Dr. Alexander Wijesinha

7/27/2017
Date

MS in CS Program Director

8/1/2017
Date

Chair, Department of COSC

7/28/2017
Date

Janet V DeLaney

8-21-17

Abstract

The efficiency of inter-domain routing in large-scale mobile network environments is a critical issue, as traditional prefixed addresses may fail due to the dynamic topologies of mobile networks, where devices would be required to maintain massive routing tables. To address this issue, in this paper we leverage the principles of probabilistic data structures in the inter-domain routing scheme, and propose a novel Bloom Filter-based dual-layer inter-domain routing scheme. In particular, we first compare several representative structures and develop a strategy to integrate bloom filters. We then propose our novel Bloom Filter-based dual-layer inter-domain routing scheme. In the design of the routing scheme, we address issues related to the overall space cost and routing loop prevention, and present the corresponding solutions. We also present detailed descriptions of the structures and algorithms in our routing scheme. Finally, we conduct a performance evaluation to validate the effectiveness of our proposed scheme. Our experimental results demonstrate the effectiveness and efficiency of our proposed scheme.

Index Terms

Bloom Filter, Inter-Domain Routing, Mobile Networks, Internet-of-Things (IoT)

List of Figures:

List of Tables:

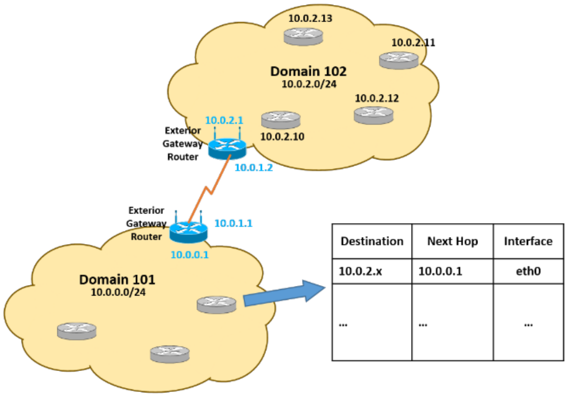
Table of Contents

I. INTRODUCTION

The Internet of Things (IoT) has developed rapidly in recent years, and is considered to be the next frontier in industrial evolution [1], [2], [3], [4]. It facilitates the massive distribution of physical objects to be controlled and observed remotely through the network. In practice, there have been a number of application domains for IoT. For example, the smart transportation system provides efficient management of transport traffic through both in-vehicle embedded sensors and deployed roadside sensors [5]. In addition, many IoT scenarios envision groups of objects connected through mobile ad-hoc networks (MANET) and next generation wireless network infrastructures [6].

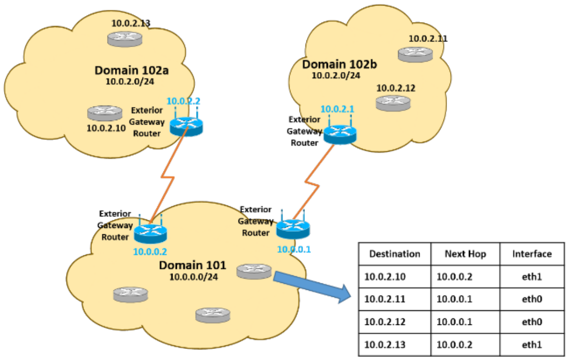
In many IoT applications, machine-to-machine (M2M) communication is required, where the end-point objects are able to remotely communicate through the network for the purpose of monitoring or controlling [7]. Typically, an IoT application domain would contain a large number of addressable physical objects and devices, and the potential scale is increasing all the time with the advancement of IoT. To ensure that every object is reachable, large routing tables must be maintained in each device. Nonetheless, single end-point objects, such as sensors, are typically constrained by power, memory, storage, and computation, and are thus unable to support such large routing tables. Meanwhile, it takes more processing time to search through the long routing table to match the destination addresses. Thus, a high-efficiency routing scheme is critical for IoT applications. In the routing design of large static networks, a traditional solution is to keep the prefixed addresses of objects in the routing table, which are in the same domains. The Border Gateway Protocol (BGP) [8] is one of the typical protocols that uses the prefix method. Figure 1 illustrates the prefixed inter-domain routing scheme, in which all objects in

Domain 102 can be grouped to one prefixed address in the routing tables of an object in Domain 101.



Destination	Next Hop	Interface
10.0.2.x	10.0.0.1	eth0
...	...	...

Fig. 1 Prefixed Inter-Domain Routing



Destination	Next Hop	Interface
10.0.2.10	10.0.0.2	eth1
10.0.2.11	10.0.0.1	eth0
10.0.2.12	10.0.0.1	eth0
10.0.2.13	10.0.0.2	eth1

Fig. 2 Partitioned Domain

Nonetheless, in mobile networks, such as MANET and other networks for IoT applications, the prefix approach can be challenging. As shown in Figure 2, the MANET Domain 102 is partitioned into two sub-domains 102a and 102b. Once the domain is partitioned with random grouped objects, there may be no appropriate prefixed address that can include one partition of addresses without involving any object in the other partition. In that circumstance, all addresses should be listed in the routing table, and the prefix cannot be used. As a result, it is obvious that when the scale of domain (the number of objects) is large, it is inefficient and costly to keep large routing tables, due to both processing time and storage cost.

There have been a number of research efforts devoted to inter-domain routing with dynamic topologies [9], [10], [11], [12], [13], [14]. For example, Dressler et al. in [14] proposed an ant colony optimization-based framework to obtain routing between multiple mobile network domains. Rekha et al. in [9] developed a Scalable Cluster-based Inter-Domain Routing protocol (SCIDR) for heterogeneous MANETs, which provides improved scalability by leveraging information such as CSI-channel state and channel correlation factor. Nonetheless, these mechanisms still require maintaining very large routing tables, even with a small network size, due to the mobility property of MANETs. Particularly, in the context of IoT, how to design an effective routing scheme to connect a massive number of objects, which can be either static or mobile, remains a significantly challenging issue.

In our research, we utilize and integrate the concepts of probabilistic data structures to address the member list issue, and make efficient inter-domain routing in dynamic topologies possible. To be specific, we first compare several space-efficient probabilistic data structures, including the Bloom Filter [15], one of its variants (Counting Bloom Filter) [16], and the Cuckoo Filter [17]. Based on the requirement of features respect to time efficiency, space efficiency, group aggregation, and insertion rejection, we then propose a new structure, which is a combination of the Bloom Filter and Counting Bloom Filter, and is able to insert, query, delete, and transmit the target objects (i.e., IP addresses) efficiently.

Based on our proposed data structure, we then design a new scheme, named the dual-layer inter-domain routing scheme for large scale mobile environments. To design an effective and efficient routing scheme, we consider how to address issues related to overall space cost and routing loops, and propose corresponding solutions. We also define the structure and procedure of the routing scheme, and design an experiment with multiple scenarios to evaluate the performance of the newly proposed routing scheme. The experimental results show that the routing scheme is both effective, producing 100 % packet delivery, and efficient, requiring low overhead.

The remainder of this paper is organized as follows: In Section II, we briefly review several representative probabilistic structures (the Bloom Filter, the Counting Bloom Filter, and the Cuckoo Filter), characterize their features, and systematically compare their performance. In Section III, we propose a novel dual-layer inter-domain routing scheme for large-scale mobile environments. In Section IV, we evaluate the effectiveness of our proposed scheme. In Section V, we conduct a literature review of existing works. Finally, we conclude the paper in Section VI.

II. Probabilistic Structures and Comparison

In this section, we review and compare several representative probabilistic structures, and introduce our proposed approach to achieve the requirement of inter-domain routing.

A. Probabilistic Structures

1) Bloom Filter: The Bloom Filter is a probabilistic data structure, which enables space efficiency. One of widely used applications of the Bloom Filter is to test whether an element is in a set of elements. Generally speaking, the body of the bloom filter is a bit array of m bits, in which every bit is set to 0, indicating an empty set. Also, k different hash functions are defined with the hash value ranges in m. To insert an element, the element is hashed by these k different hash functions, and each hash value determines a position in the bit array. The bits of those positions are set to 1. To query an element from the bloom filter, the element is hashed by each of these k hash functions, and a positive result returns if all corresponding bits are set to 1.

There are two important issues with the Bloom Filter. One is the false positives, where the probabilistic structure cannot guarantee a 100 % accuracy when querying the elements.

The other is deletion of elements, in which the potential overlapping positions make it difficult to delete an element without affecting others.

2) Counting Bloom Filter: To overcome the issue of deletion, the Counting Bloom Filter was introduced in 2000 [16]. It implements the operation of deletion on the original bloom filter. The idea is to replace the original bit array with a counter array, where each position is now a counter that stores the number of elements that use this position. As a result, more space is required compared to the original Bloom Filter. Meanwhile, the size of the counter determines the maximum number of overlaps allowed. As a consequence, inserting an element may be rejected if any corresponding counter has reached the limit.

3) Cuckoo Filter: Another candidate that supports deletion is the Cuckoo Filter, which was initially introduced in 2014 [17]. Using two hash functions with more compact space utilization, the Cuckoo Filter is able to achieve similar space cost and time complexity in query compared to the original bloom filter. Generally speaking, a Cuckoo Filter contains m buckets, each bucket has b entries with length of f bits. To insert an element, the element x is hashed using the first hash function H1 to an f-bit fingerprint f. Then, x and f are both hashed to the range of m using the second hash function H2. The two hash values H2(x) and H2(f) will determine two candidate positions at H2(x) and H2(x)$\oplus$H2(f, the fingerprint f is then inserted to either empty candidate position. In the case where both positions are full, it relocates an existing element to its other candidate location. The relocation repeats if there is still conflict, until it reaches the maximum number of attempts, indicating the failure of insertion. Notice that the time complexity of insertion is variable, and the worst case is more likely to occur when the filter is approaching the fully utilization. Querying or deletion requires constant time to check or delete from the bucket at candidate positions.

False positives also occur in the Cuckoo Filter, as it uses only two hash functions for the fingerprint and positions.

B. Design Rational of Combining Bloom Filters

To meet the requirements of inter-domain routing in large scale networks, we consider the features of Deletion Support, Space Cost, Time Complexity, Insertion Rejection, and Group Aggregation as the 5 most important criteria for the candidate structures. We compare and generalize the feature comparison in Table I. It can be observed that the original Bloom Filter has the advantages of both space and time efficiency, insertion rejection avoidance, and group aggregation. Nonetheless, it lacks support for deletion. The Counting Bloom Filter is able to support deletion, but at the cost of more space as a trade-off, in comparison with the original Bloom Filter. The Cuckoo Filter, although supporting deletion without the cost of additional space, has significant drawbacks of insertion time, insert rejection, and aggregation.

Considering the requirements of inter-domain routing, we thus propose a new structure, which is the combination of the Counting Bloom Filter and the original Bloom Filter. Our idea is based on the assumption that the gateways of the network domains are not constrained for resources in the same manner as the endpoint devices. We let each gateway of the domain maintain a Counting Bloom Filter for intra-domain objects, and a Bloom Filter for inter-domain routing. This would allow the gateway to add and delete objects within the domain, such that it can keep track of the dynamic topology. For each incoming data packet in the gateways, the only operation that is needed is query, while insertion and deletion are not necessary. Thus, we use the original Bloom Filter to represent each domain, and the gateways send their own Bloom Filters to each other for routing purposes. For each individual object that has limited capabilities, there is no need to maintain any bloom filter-based routing tables.

In our new approach of a combined Bloom Filter scheme, we address the issue of deletion, enable the additional space that is only needed in the gateway, allow the constrained objects maintain a low profile, and provide for updating the bloom filter in the routing table through flooding. In addition, it combines the benefits of both the original Bloom Filter, and the Counting Bloom Filter, making group aggregation easy for the potential use of multicast. Using the approach of combined Bloom Filters, we propose a dual-layer inter-domain routing scheme in Section III.

Table I Structures Comparison

Structure	BF	CBF	Cuckoo	CombinedBF
Deletion Support	No	Yes	Yes	Yes
Space cost	1	3x-4x	about 1x	1x / 3x-4x
Time Insertion	O(1)	O(1)	O(n)	O(1)
Time Deletion	N/A	O(1)	O(1)	O(1)
Time Query	O(1)	O(1)	O(1)	O(1)
Insertion Rejection	No	Rare	Yes	Rare
Group Aggregation	Boolean	Boolean	Query/Insert	Boolean

III. Our Approach: A New Dual-Layer Routing Scheme

In this section, we introduce a scheme called dual-layer routing scheme, which applies the structure of combined bloom filters that we presented in subsection II-B.

A. Overview

In a typical large-scale mobile network that supports both IoT applications, and the communication of multiple MANETs dedicated to particular missions, a large number of objects are distributed within several domains with either fixed or dynamic topologies. The basic idea of the dual-layer inter-domain routing scheme is to have the gateways in each domain maintain two layers of routing tables (internal and external), to which the Combined Bloom Filter proposed in subsection II-B is applied.

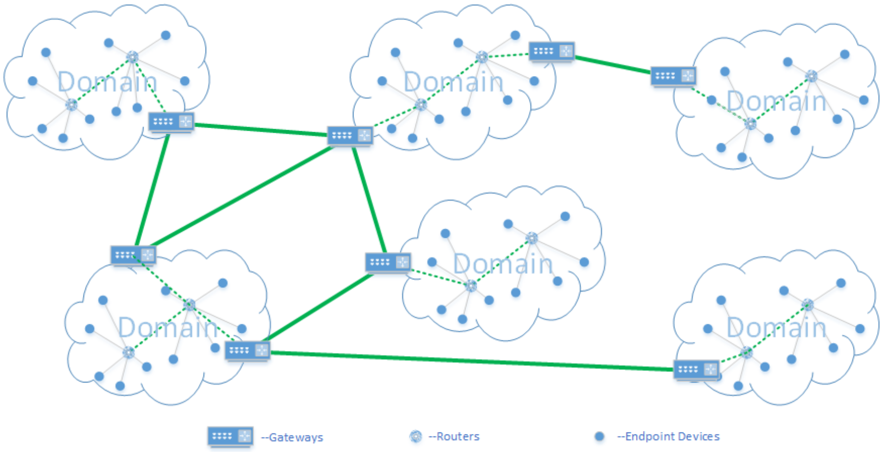


Fig. 3 Network Model for Inter-Domain Routing

To be specific, the internal layer includes a traditional routing table with all the objects in the domain, and a Counting Bloom Filter that contains these objects. The external layer includes the routing table of only gateways in the network, the original Bloom Filter in each gateway, and the aggregation of Bloom Filters corresponding to each interface. This design insures that inter-domain routing is handled only by the gateways (which are not resource constrained) in an efficient manner, and precludes the routers and endpoint objects inside each domain from needing to maintain a long list of routing tables. In the meantime, any change in the topology of the domain (on and off of devices, location changes, etc.) is handled by the corresponding gateway, and does not need to inform every other object in the entire network.

Figure 3 illustrates the network model where the proposed routing scheme operates. In this model, the network consists of multiple domains, where each domain consists of groups of objects, including endpoint sensors, mobile and fixed devices, routers, etc. The gateways of each domain are responsible for inter-domain routing, which enables the communication between devices in different domains. Notice that the connection between two gateways is defined as “direct" when all intermediate hops are gateways, while the connection is “indirect" when the crossing a domain through intra-domain objects as well as gateways. For the intra-domain structure, this can be composed of fixed, mobile, or both kinds of routers, where each may be connected with endpoint devices.

In our proposed routing scheme, we consider the following three types of objects in the network, based on their capabilities: (i) Type I Objects: Objects in this category are the end- point devices that do not need routing capabilities. These devices only connect to one router at a time so that the routing table can be very small, which minimizes the resource consumption in the most constrained objects (sensors, mobile devices, etc.). Type I objects may be turned on and off at frequently, and may move among the domains if they are mobile devices. (ii) Type II Objects: Objects in this category are the devices with intra-domain routing capabilities, but are not able to route to inter-domain devices without gateways. These devices can be either fixed or mobile, where each typically maintains the routing table of all objects within the domain. In our network model, the Type II objects are connected to many Type I objects to provide access to the network, as well as at least one Type II or Type III object to reach other domains. (iii) Type III Objects: Objects in this category are gateways that provide the inter-domain routing among the domains in the network. Inside each domain, the gateways additionally act as routers, connecting to Type II and Type I objects. In our network model, the gateways are considered non-constrained devices, meaning they have adequate resource capabilities. They may have some mobility, but most are deployed in fixed locations.

B. Issues and Solutions

Because Bloom Filters are used in our proposed dual-layer inter-domain routing scheme, this will reduce the available space if a bloom filter contains a majority of the objects in the network. Nonetheless, there would be multiple bloom filters used in a network, and multiple hits may occur in the query due to the accuracy (i.e. false positives). As consequences of the aforementioned characteristics, the following two issues need to be addressed: (i) the overall space cost, and (ii) the routing loop prevention. We describe the details of these problems, and propose corresponding solutions, in the following.

1) Space Cost: One issue of applying bloom filters is the space cost of the data structures. Our goal is to have lean routing tables in the routers. Recall the scenario of dynamic topologies, mentioned in Section I, where typical intra-domain routers (defined as Type II objects) need to maintain the routing table consisting of all the objects of the network. If we replace the list with bloom filters, the actual space cost will depend on the number of bloom filters and the size of each bloom filter. According to the structure of the Bloom Filter, the optimal size of a bloom filter is determined by the total number of elements when the target false positive rate is determined (fixed). For instance, with the target false positive rate to be 0.0156, each element would take 8.66 bits in average. If we use the 128-bit IPv6 address as the element, each address would take about

1/15 of its original length. That means the router should maintain no more than 15 bloom filters. Otherwise, it does not save the space.

Typically, there are far more Type II objects (intra-domain routers) in a system where many applications are deployed. Thus, it is too costly to use bloom filters for each router with the connected Type I objects (sensors, endpoints, etc.), and list them all. In our approach, Bloom Filters and Counting Bloom Filters are used only in the Type III objects (gateways). Each gateway maintains a Counting Bloom Filter containing only the objects in the domain where the gateway is located, as well as maintaining a list of the Bloom Filters of the other gateways and interfaces. As a result, the intra-domain routers maintain routing tables of only objects inside the domain. They forward the packets with outside destinations to the default gateway of the domain, and let the gateways determine the next steps. This strategy works for domains with both single and multiple gateways, and we detail the procedure in subsection III-E.

2) Routing Loop Prevention: Routing loops are another critical issue that need to be addressed. Because of the false positive rate in bloom filter queries, there could be multiple hits when querying one address in a group of bloom filters. Of all these hits, at most one is the true target. Nonetheless, it is not easy for the router to determine which is correct at this stage. If the strategy is to forward packets to all possible candidate hops, there could be routing loops, where those hops forward the packets to each other, back and forth infinitely, leading to large overhead and low efficiency of packet delivery.

To avoid routing loops, we propose a repacking strategy, which would lock the target domain and deliver the packets to the specified gateway(s) for intra-domain routing.

This strategy not only prevents the routing loop, but also reduce the probability of false positive hits as well. The detailed procedure will be presented in subsection III-E.

C. Structures of Routing Tables

We define the structures of routing tables for each type of object.

1) Type I Objects: The Type I objects do not maintain any routing table. They will send and receive data packets using the default interface connected to Type II and Type III objects. Thus, any device that supports ipv6 addresses is able to be added to the network without additional setup.

2) Type II Objects: The Type II objects (routers) form the main structure of the domains, which can be of fixed, mobile, or mixed topologies. Two tables are maintained in each Type II object. One (T2-1, shown in Structure 1) is a typical routing table that contains destinations in IPv6 addresses, next hops in IPv6, interfaces, and costs. The other table (T2-2, shown in Structure 2) is a gateway table, which lists the IPv6 addresses of all gateways (Type III objects) belonging to the domain.

Structure. 1
T2-1: Routing Table in Type II Objects

Dest IPv6	Next Hop IPv6	Interface	Cost
...	...	...	...
Object i	Object j	Interface r	c
...	...	...	...

Structure. 2
T2-2: Gateway Table in Type II Objects

Gateway in IPv6
...
Gateway g
...

Notice that prefixed addresses are not used in the routing table since we assume every object has a fixed IPv6 address, and can move to another domain without a change in address. Compared with storing all objects in the network, the routers only need to store the objects in the domain, and no table of bloom filters is needed.

3) Type III Objects: The type III objects (gateways) have two layers of tables and a Counting Bloom Filter. As one of the objects that form the domain, it maintains the same table as the Type- II objects, representing all internal (intra-domain) interfaces. T3-I-1 (shown in Structure 3) is a typical routing table, which includes the destinations in IPv6 addresses, next hops in IPv6, interfaces, and costs. T3-I-2 (shown in Structure 4) is a gateway table, which lists the IPv6 addresses of all gateways (Type III objects) belonging to the domain.

Structure. 3
T3-I-1: Internal Routing Table in Type III Objects

Dest IPv6	Next Hop IPv6	Interface
...	...	...
Object i	Object **j**	Interface **r**
...	...	...

Structure. 4
T3-I-2: Internal Gateway Table in Type III Objects

Gateway in IPv6
...
Gateway g
...

A Counting Bloom Filter (CBF, shown in Structure 5) is maintained by each Type-III object, and contains all the objects in the domain. Recall that the Counting Bloom Filter allows objects to be inserted or deleted.

Structure. 5
Counting Bloom Filter of the domain in Type III Objects

CBF_g

As a gateway, three more tables representing the external (inter-domain) interfaces are maintained by the Type III objects as follows.

First, the External Gateway Table (T3-E-1, shown in Structure 6) lists all the reachable gateways in the network, next hops, Interfaces, and costs. Gateways in the same domain are also listed in this table. If the only route to connect to the next hop gateway is through the inter-domain routers, the interface is marked as internal for future internal procedures (details can be found in subsection III-D). This table is not large as the number of gateways is relatively small. Thus, query to this table can be fast. This table is used to establish the bloom filter related tables, and for fast routing between gateways.

Structure. 6
T3-E-1: External Gateway Table in Type III Objects

Dest Gateway (IPv6)	Next Hop Gateway (IPv6)	Interface	Cost
...	...	...	...
Gateway i	Gateway **j**	Interface **r**	c
...	...	...	...

Second, the External Gateway Bloom Filter Table (T3-E-2, shown in Structure 7) lists the reachable gateways in the network, and the Bloom Filters representing each gateway's domain (converted from the Counting Bloom Filters that each gateway maintains). The size of this table is determined by the number of gateways and the size of each bloom filter. The table may be large, but query through the table can be fast because it only needs to check a fixed number of positions in each bloom filter, equivalent to the number of hash functions. This table is used with the External Gateway Table to establish the Interface-Bloom Filter table, and performs as a backup table to handle false positive hits.

Structure. 7.
T3-E-2: External Gateway Bloom Filter Table in Type III Objects

Gateway in IPv6	Bloom Filter of the Domain
...	...
Gateway g	BF_g
...	...

Third, the Interface-BloomFilter Table (T3-E-3, shown in Structure 8) lists all interfaces of this object, and each associated Bloom Filter. The value of the Bloom Filter is the result of performing OR operations to all of the Bloom Filters of the gateways that route to this interface. Like the Gateway-BloomFilter table, this table may be large, but is also fast to query because only a fixed number of positions in each bloom filter need to be checked. This table is used as the primary routing table.

Structure. 8
T3-E-3: Interface-BloomFilter Table in Type III Objects

Interface	Aggregated Bloom Filter
...	...
Interface r	$BF_i + BF_j$+...
...	...

D. Establishment Routing Tables

In the following, we describe the procedure of establishing routing tables in sequence from inside to outside a domain. Generally speaking, the intra-domain and inter-domain routing tables are established in parallel. The latter is also updated based on the Bloom Filters maintained by each gateway, and tracks the updated intra-domain topology.

1) Intra-Domain Routing: The intra-domain routing algorithm follows traditional link-state routing algorithms. The routers (Type II objects) and gateways (Type III objects) of the domain construct their maps of connectivity. By sharing the link information with their neighbors, each objects updates its map, forming the Routing Table (T2-1 and T3-I-1). In addition, each gateway initializes its Internal Gateway Table (T3-I-2) with its own identity and shares the table to the neighbors. The recipients then update the Gateway Tables (T2-2 and T3-I-2) and share them to other neighbors. Once the establishment is finished, each Type II and III object should have these two tables with the full information of the domain.

2) Counting Bloom Filter: When the gateway gets an update on its intra-domain routing information (Table T3-I-1), it updates its Counting Bloom Filter. New objects are inserted, and objects that no longer exist are deleted. The updated Counting Bloom Filter is further converted to original Bloom Filters for sending out to the other gateways, in order to update the inter-domain routing tables.

3) Inter-Domain Routing: Three tables need to be established for the inter-domain routing in gateways. The External Gateway Table (T3-E-1) is updated by using the link-state routing algorithms for directly connected gateways. Meanwhile, the gateway checks its Internal Gateway Table (T3-I-2) for the gateways in the same domain, where the cost for the indirect connection is assigned with a larger number.

Once the External Gateway Table is established or updated, the gateway will use this routing table to send its domain Bloom Filter to each item listed in the table. Upon the receipt of the Bloom Filter, the External Gateway-Bloom Filter Table (T3-E-2) is updated. Then, the update of the Interface-Bloom Filter Table (T3-E-3) is triggered by the update of either T3-E-1 or T3-E-2. It then re-calculates the Aggregated Bloom Filters for the involved interfaces.

E. Routing Procedures

In the following, we describe the routing procedures in detail. We first introduce the concept of repacking, which is an important mechanism used in the routing procedure to prevent routing loops. We then describe detailed flow for each type of object to process data packets.

1) Repacking: Before introducing the routing procedures, we first present the concept of repacking. Repacking is the process of putting an IPv6 packet into a new packet, which has a new destination address and its payload is the original packet. In our proposed routing scheme, the repacking destinations are Type III objects (gateways). Repacking is used for packets to cross domains, and for external gateways to handle false positive hits. The former is when a data packet needs to cross a domain where its original destination is not in the intra-domain routing table, it is repacked with the gateway on the other side as the destination, so that the routers in the domain know where it should be forwarded. The latter case, occurs when there are multiple hits in external routing, and the original packet is repacked and forwarded to the gateway of individual candidate domains. These gateways would unpack and check the original packet only in the internal routing tables, preventing routing loops among the candidate gateways.

2) Type I Objects: Type I objects do not maintain routing tables. Thus, the outgoing packets will go directly to the Type II or Type III object that it is connected with. It is also the destination of all incoming packets.

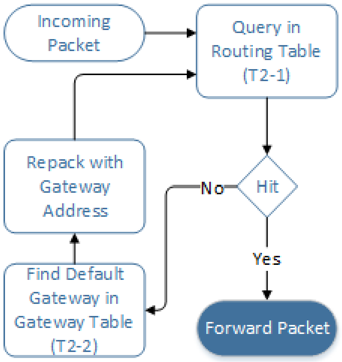

Incoming Packet
Query in Routing Table (T2-1)
Repack with Gateway Address
No
Hit
Yes
Find Default Gateway in Gateway Table (T2-2)
Forward Packet

Fig. 4 Processing Incoming Packets by Type II Objects

3) Type II Objects: An incoming packet to Type II objects would fit one of the following scenarios: (i) This is the destination itself: In our scheme, if the destination of a packet is a non-gateway object, it must be an original packet. Thus, no further forwarding is needed. (ii) Destination is outside the domain: The packet must come from its directly connected Type I objects. Since the Type II objects only maintain routing tables for intra-domain objects, the packet would be repacked with the first object in the Gateway Table (T2-2) as the new destination, and forwarded according to the Routing Table (T2-1). (iii) Destination inside the domain: The packet could be a repacked packet to the gateways or an original packet to one of the objects inside the domain. Whichever it is, it would match one item in the Routing Table (say T2-1) and be forwarded to the next hop.

Figure 4 illustrates the workflow of Type II objects in processing incoming packets. The incoming packet is searched in the Routing Table (T2-1) and forwarded according to the hit. A miss indicates an outside destination, where the packet is repacked using the default gateway address found in Gateway Table (T2-2). There must be a hit after repacking so that the repacked packet will be forwarded accordingly.

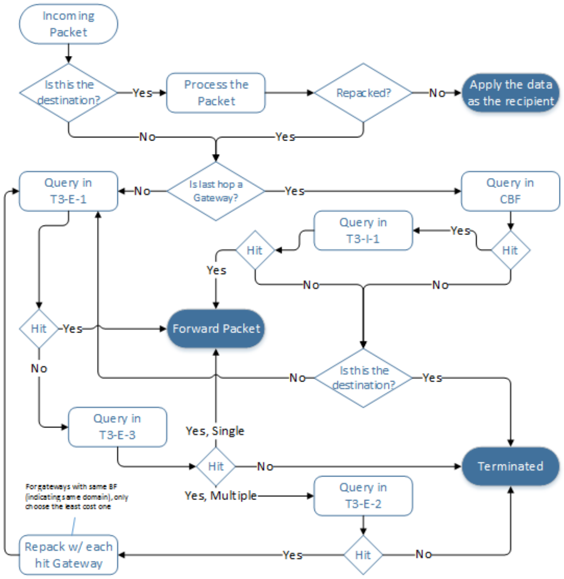


Fig. 5 Processing Incoming Packets by Type III Objects

4) Type III Objects: Figure 5 illustrates the workflow for Type III objects to process incoming packets. An incoming packet to the Type III objects could be from either internal interfaces or external interfaces. From internal interfaces, we have (i) Destination is inside the domain: This is a packet from one inside object to another inside object. The gateway is only used as an inter-domain router. The packet would match one item in the Internal Routing Table (T3-I-1) and be forwarded according to it. (ii) Destination is outside the domain: The packet must come from its directly connected Type I objects. Otherwise, the outgoing packets would be repacked by the routers or another gateway of the domain (for crossing the domain). Since Type III objects maintain the tables for inter-domain routing, the packet would be queried in the external tables and forwarded accordingly. (iii) This is the destination itself: The packet could be original, and then it needs no further actions. In most cases, it is a repacked packet, either from an internal router or another gateway of the domain. In this case, the packet should be unpacked for the original destination, which can be queried in the external tables.

From external interfaces, we have (i) Destination is inside the domain: This is a packet from one outside object to one inside. It does not have false positive matches during the previous forwarding activities. The packet would match one item in the Internal Routing Table (T3-I-1), and be forwarded according to it. (ii) Destination is outside the domain: The packet is on its way to the destination domain, it would be queried in the external tables and forwarded accordingly. (iii) This is the destination itself: The packet could be original, and then it needs no further actions. In most cases, it is a repacked packet by one of its previous gateways due to a false positive hit. In this case, the packet should be unpacked for the original destination, and then be queried only in the internal tables.

IV. Performance Evaluation

To evaluate the performance of our proposed dual-layer routing scheme, we have implemented the combined bloom filter in Java (modified from the source code in Apache Hadoop 2.7.3).

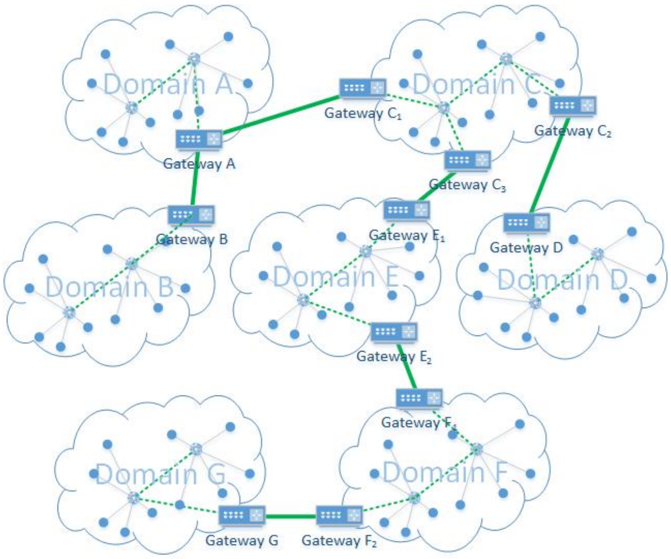


Fig. 6 A Typical Network with the Combination of Different Domains

We have designed simulation scenarios based on the types of domains and types of connections. Figure 6 illustrates a typical network in our evaluation. As it can be observed, the objects are located in two types of domains, those with a single gateway (Domain A, B, D, and G), and those with multiple gateways (Domain C, E, and F). There are also two types of connections between two domains, those that are neighbors (such as Domain A and B), and those that must cross a third domain (such as Domain A and D, have to cross C). Combining these two dimensions, and considering the number of gateways in both the source and destination, we have a total of eight scenarios that are listed in Table II.

Table II. Evaluation Scenarios

Src and Dst Type	Neighbour	Across
Single to Single	A→B (S1)	A→D (S5)
Single to Multiple	A→C (S2)	A→E (S6)
Multiple to Single	C→A (S3)	E→A (S7)
Multiple to Multiple	C→E (S4)	C→F (S8)

In our evaluation, we set the total number of objects to 70000, evenly distributed to 7 domains, as shown in Figure 6. Of the 10000 objects in each domain, the number of Type III objects (gateways) in total is eleven, as shown in Figure 6, the number of Type II objects (routers) are set to 50, and the remainder are Type I objects. Each intra-domain topology is randomly generated, and we ensure all the routers and the gateways are reachable by each other. The bloom filter is configured with the number of hash functions set to 7, where the optimal false positive rate is 0.0078, and the filter size is 706920 bits. In evaluating each scenario, one Type I object randomly selected from the source domain will send data packets to every object in the destination domain, which is a total of 10000 packets.

The key performance metrics for the evaluation are delivery rate and overhead. The delivery rate is defined as the ratio of packets being successfully forwarded to the destination, which represents the reliability of the routing scheme. Overhead is defined as the ratio of packets that need repacking and multiple forwarding due to the false positive hits in querying the bloom filters, and it represents some degree of efficiency. To be specific, we count the number of packets repacked due to multiple hits, the number of packets repacked due to crossing domains, and the number of multiple forwards.

Table III. Data Delivery

Scenario	Packet Sent	Packet Delivered	Delivery Rate
S1	10000	10000	100%
S2	10000	10000	100%
S3	10000	10000	100%
S4	10000	10000	100%
S5	10000	10000	100%
S6	10000	10000	100%
S7	10000	10000	100%
S8	10000	10000	100%

Packet Delivery: Table III shows the delivery rate of data packets for each scenario that we tested. As it can be observed, 100 % of all data packets are delivered to the target destinations.

Overhead: Table IV represents the overhead when data packets are transmitted in eight scenarios that we have designed. According to the results, we find that repack from multiple hits is rare, and may only occur when the majority of the objects are on the opposite direction. For instance, in Scenario S_1, 50000 objects out of 70000 are on the other interface in Gateway A when trying to send packets to Domain B, which causes 14 false positive hits out of 10000 packets. Thus, these 14 packets are repacked and delivered to Gateway B by following the algorithm that we have designed. Since the bloom filter in each individual gateway has only 1/7 of the total objects, it is very rare to have false positive hits in this level. In our experiment, no

additional packets would be needed. Also, notice that the majority of overhead is from repacking to cross a domain. For all communications that need to cross a domain, such as S_5, S_6, S_7, and S_8, repacking is required.

In another case where the source objects are in the domain with multiple gateways, if the default gateway is not the correct one, repacking is required to deliver the packet to the correct gateway. For instance, cases S_4 and S_8 are the two scenarios that apply this kind of repacking. Although the repacking may not be avoided in real-world deployment, it would not be large as only one IPv6 header is added. To conclude, the overall overhead of applying our proposed routing scheme is maintained in a low level.

Table IV. Overhead of 10000 Packets

Scenario	Repack for Crossing	Repack for Multiple Hits	Additional Packets
S1	0	14	0
S2	0	0	0
S3	0	0	0
S4	10000	0	0
S5	10000	0	0
S6	10000	0	0
S7	10000	0	0
S8	20000	0	0

V. RELATED WORK

With the advancement of modern computer and communication technologies, various hetero- geneous networks (MANET, etc.) have been highly integrated into a large-scale mobile network to support diverse mobile applications. When a massive number of mobile objects are deployed in the network, significant challenges have arisen in designing appropriate routing mechanisms to support effective and reliable routing among objects and network management protocols [18], [19], [20], [21], [22], [23], [24]. Relevant to the routing of mobile objects, a number of MANET routing protocols [25], [26], [27], [28] have been developed. Using MANET as an example, the existing routing protocols can be categorized into two groups: inter-MANET routing, and intra-MANET routing.

The Border Gateway Protocol (BGP) has been considered an effective inter-domain routing protocol for static networks, but it is not suitable for large-scale mobile networks due to the dynamic network topology caused by node mobility. A number of efforts [25], [26], [27], [28] have been devoted to inter-MANET routing. Instead of targeting mobile network scalability like inter-MANET routing, intra-MANET routing [29], [30], [31], [32] focuses on optimal route discovery to improve overall network performance.

Bloom Filters have been applied to many domains and applications, including network routing [33], [34], [35], [36]. For example, Oigawa and Sato in [33] proposed the use of Bloom Filters to control redundant packets, which were sent by the Zone Routing Protocol (ZRP) in MANETs. Sasaki and Makido in [34] proposed to use a distributed Bloom Filter Table (DBFT) to substantially reduce overhead for highly-dynamic peer-to-peer (P2P) distributed storage systems. Trindade et al. in [36] proposed the use of the Time Aware Bloom Filter (TAB) to remove the values of out-of-date topology information. Notice that, although Trindade's work offered a solution for removal of an entry in a Bloom Filter, the protocol only addressed a flat MANET, rather than a multi-domain MANET.

VI. Final Remarks

In this paper, we addressed the issue of inter-domain routing in large-scale mobile networks that connect a massive number of objectives (e.g., IoT and multiple MANETs) by using the concept of probabilistic data structures. Particularly, we first list and compare three probabilistic structures. Based on the requirement of inter-domain routing, we then conceived of the combined bloom filter scheme, and proposed a new dual-layer inter-domain routing scheme. The issues related to the overall space cost and the routing loop prevention were defined and corresponding solutions were presented. We also presented the detailed description of structures and algorithms implemented in our routing scheme. In addition, we designed an experiment to apply the defined structures and algorithms with multiple data transmission scenarios, and evaluated the performance of our proposed scheme. The experimental results show that our proposed approach is capable of ensuring a 100 % delivery of data packets with very little additional overhead. Meanwhile, our scheme is 100 % compatible with the current IPv6 data packets, and can be used within a subnet that connects to the traditional backbone network.

REFERENCES

[1] John A. Stankovic. Research directions for the internet of things. IEEE Internet of Things Journal, 1(1):3–9, 2014.

[2] Guobin Xu, Wei Yu, David W. Griffith, Nada Golmie, and Paul Moulema. Toward integrating distributed energy resources and storage devices in smart grid. IEEE Internet of Things Journal, 4(1):192–204, 2017.

[3] Jie Lin, Wei Yu, Nan Zhang, Xinyu Yang, Hanlin Zhang, and Wei Zhao. A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. IEEE Internet of Things Journal, 2017.

[4] Nnanna Ekedebe, Houbing Song, Wei Yu, Chao Lu, and Yan Wan. Securing transportation cyber-physical systems. Securing Cyber-Physical Systems, 2015.

[5] Nnanna Ekedebe, Chao Lu, and Wei Yu. On experimental evaluation of intelligent transportation system (its) safety and traffic efficiency. In Proc. of IEEE International Conference on Communication (ICC), 2015.

[6] Wei Yu, Hansong Xu, Hanlin Zhang, David Griffith, and Nada Golmie. Ultra-dense networks: Survey of state of the art and future directions. In Proc. of IEEE International Conference on Computer Communication and Networks (ICCCN), 2016.

[7] Amirshahram Hematian, Wei Yu, Chao Lu, David Griffith, and Nada Golmie. Clustering-based device-to-device communication to support diverse applications. In Proc. of ACM International Conference on Reliable & Convergent Systems (RACS), 2016.

[8] Y. Rekhter, T. Li, and S. Hares. A border gateway protocol 4 (bgp-4), Jan 2006. Internet Engineering Task Force, RFC4271.

[9] B Rekha and DV Ashoka. Scidr: A scalable cluster based inter-domain routing protocol for heterogeneous manet. International Journal of Computer Applications, 122(4), 2015.

[10] Izegbuwa Okundaye, Thomas Kunz, and Semra Gulder. Inter-domain routing for tactical mobile ad-hoc networks. In Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th, pages 1–6. IEEE, 2014.

[11] Ziane Sara, Mekki Rachida, and Thomas Kunz. A reactive inter-domain routing protocol for manets. MOBILITY 2015, page 46, 2015.

[12] Keisei Okano and Yoshiaki KAKUDA. An inter-domain routing protocol based on autonomous clustering for heterogeneous mobile ad hoc networks. IEICE Transactions on Communications, 98(9):1768–1776, 2015.

[13] Joy Na Wang, Joshua Van Hook, and Patricia Deutsch. Inter-domain routing for military mobile networks. In Military Communications Conference, MILCOM 2015-2015 IEEE, pages 407–412. IEEE, 2015.

[14] Falko Dressler and Mario Gerla. A framework for inter-domain routing in virtual coordinate based mobile networks. Wireless networks, 19(7):1611–1626, 2013.

[15] Burton H. Bloom. Space/time trade-offs in hash coding with allowable errors. Commun. ACM, 13(7):422–426, July 1970.

[16] Li Fan, Pei Cao, J. Almeida, and A. Z. Broder. Summary cache: a scalable wide-area web cache sharing protocol. IEEE/ACM Transactions on Networking, 8(3):281–293, Jun 2000.

[17] Bin Fan, Dave G. Andersen, Michael Kaminsky, and Michael D. Mitzenmacher. Cuckoo filter: Practically better than bloom. In Proceedings of the 10th ACM International on Conference on Emerging Networking Experiments and Technologies, CoNEXT ’14, pages 75–88, New York, NY, USA, 2014. ACM.

[18] David Airehrour and Jairo Gutierrez. An analysis of secure manet routing features to maintain confidentiality and integrity in iot routing. In Ontario, Canada. CONF-IRM, volume 2015, 2015.

[19] G Vithya and B Vinayagasundaram. Qos by priority routing in internet of things. Research Journal of Applied Sciences, Engineering and Technology, 8(21):2154–2160, 2014.

[20] Zhikui Chen, Haozhe Wang, Yang Liu, Fanyu Bu, and Zhe Wei. A context-aware routing protocol on internet of things based on sea computing model. Journal of Computers, 7(1):96–105, 2012.

[21] Sharief MA Oteafy, Fadi M Al-Turjman, and Hossam S Hassanein. Pruned adaptive routing in the heterogeneous internet of things. In Global Communications Conference (GLOBECOM), 2012 IEEE, pages 214–219. IEEE, 2012.

[22] Weichao Gao, James Nguyen, Daniel Ku, Hanlin Zhang, and Wei Yu. Performance evaluation of netconf protocol in manet using emulation. Software Engineering Research, Management and Applications, volume 654 of the series Studies in Computational Intelligence, pages 11–32, 2016.

[23] Paul Loh Ruen Chze and Kan Siew Leong. A secure multi-hop routing for iot communication. In Internet of Things (WF-IoT), 2014 IEEE World Forum on, pages 428–432. IEEE, 2014.

[24] Weichao Gao, James Nguyen, Wei Yu, Chao Lu, and Daniel Ku. Assessing performance of constrained application protocol (coap) in manet using emulation. In Proc. of ACM International Conference on Reliable & Convergent Systems (RACS), 2016.

[25] Shohei Fujiwara, Tomoyuki Ohta, and Yoshiaki Kakuda. An inter-domain routing for heterogeneous mobile ad hoc networks using packet conversion and address sharing. In Distributed Computing Systems Workshops (ICDCSW), 2012, 32nd International Conference on, pages 349–355. IEEE, 2012.

[26] You Lu, Biao Zhou, Ian Ku, and Mario Gerla. Connectivity improvement for inter-domain routing in manets. In MILITARY COMMUNICATIONS CONFERENCE, 2010-MILCOM 2010, pages 617–622. IEEE, 2010.

[27] Seung-Hoon Lee, Starsky HY Wong, Chi-Kin Chau, Kang-Won Lee, Jon Crowcroft, and Mario Gerla. Intermr: Inter-manet routing in heterogeneous manets. In Mobile Adhoc and Sensor Systems (MASS), 2010 IEEE 7th International Conference on, pages 372–381. IEEE, 2010.

[28] Falko Dressler, Roman Koch, and Mario Gerla. Path heuristics using aco for inter-domain routing in mobile ad hoc and sensor networks. In International Conference on Bio-Inspired Models of Network, Information, and Computing Systems, pages 128–142. Springer, 2010.

[29] David Johnson, Yin-chun Hu, and David Maltz. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. Technical report, 2007.

[30] Wei Yu and Jangwon Lee. Dsr-based energy-aware routing protocols in ad hoc networks. In Proc. of the 2002 International Conference on Wireless Networks, 2002.

[31] Zygmunt J Haas, Marc R Pearlman, and Prince Samar. The zone routing protocol (zrp) for ad hoc networks. 2002.

[32] Luzi Anderegg and Stephan Eidenbenz. Ad hoc-vcg: a truthful and cost-efficient routing protocol for mobile ad hoc networks with selfish agents. In Proceedings of the 9th annual international conference on Mobile computing and networking, pages 245–259. ACM, 2003.

[33] Yuria Oigawa and Fumiaki Sato. An improvement in zone routing protocol using bloom filter. In Proc. of 2016 19th International Conference on Network-Based Information Systems, pages 107–113, 2016.

[34] Kengo Sasaki, Shinya Sugiura, Satoshi Makido, and Noriyoshi Suzuki. Bloom-filter aided two-layered structured overlay for highly-dynamic wireless distributed storage. IEEE Communications Letters, 17(4), April 2013.

[35] João Trindade and Teresa Vazão. Hran: Heat routing protocol for ad-hoc networks. In Proc. of the 10th IFIP Annual Mediterranean Ad Hoc Networking Workshop, pages 107–113, 2011.

[36] Joao Trindade, Ricardo Lopes Pereira, and Teresa Vazao. Scalability of bloom filter based routing for large scale mobile networks. In Proc. of the 2014 7th IFIP Wireless and Mobile Networking Conference (VMNC), 2014.

Curriculum Vita

NAME: Weichao Gao

[redacted]

PROGRAM OF STUDY: Computer Science

DEGREE AND DATE TO BE CONFERRED: Master of Science, 2017

Collegiate institutions attended	Dates	Degree	Date of Degree
Towson University	2014-2017	M.Sc	August 2017
University of Michigan	2009-2011	MBA	May 2011
Fudan University	2001-2005	B.Sc	July 2005

Major: Computer Science, Business Management, and Pharmacy

Minor(s), if applicable:

Professional publications:

W. Gao, J. Nguyen, D. Ku, H. Zhang, and W. Yu. Performance evaluation of netconf protocol in manet using emulation. *Software Engineering Research, Management and Applications*, volume 654 of the series Studies in Computational Intelligence, pages 11–32, 2016.

W. Gao, J. Nguyen, W. Yu, C. Lu, and D. Ku. Assessing performance of constrained application protocol (coap) in manet using emulation. *In Proc. of ACM International Conference on Reliable & Convergent Systems (RACS)*, 2016.

J. Nguyen, Y. Wu, W. Gao, W. Yu, C. Lu and D. Ku, "On Optimal Relay Nodes Position and Selection for Multi-Path Data Streaming,*" 2017 IEEE Wireless Communications and Networking Conference (WCNC)*, San Francisco, CA, 2017, pp. 1-6.