



STRATEGIC WHITE PAPER

THE DESIGN OF CONTENT-CENTRIC MANETS

BY MATTEO VARVELLO, IVICA RIMAC, UICHIN LEE, LLOYD

© 2011 IEEE. Reprinted, with permission, from The 8th International Conference on Wireless On-Demand Network Systems and Services

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of LGS ' products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org.

By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

In this paper, we consider content-centric networking as a fundamental driver for mobile ad hoc network (MANET) protocol design. The content-centric networking paradigm shift of decoupling resources from the hosts they reside on permits a focus on data delivery, rather than end-to-end reach-ability. We systematically evaluate the suitability and effectiveness of existing approaches toward designing a content-centric MANET. We leverage the extensive prior work on both resource discovery and routing. To examine and compare the various existing designs, we identify a set of representative design alternatives. We develop analytical models for these designs that evaluate their efficiency for a content-centric MANET. Our models provide insights into the strengths and weaknesses of candidate design choices. Our analysis explores the performance boundaries of MANET designs and yields surprising results comparing unstructured flooding to more complex structured solutions. Based on our results, we derive a set of recommendations that are key to the successful design of a content-centric MANET.

TABLE OF CONTENTS



[1] INTRODUCTION	03
[2] DESIGN SPACE	05
[3] MODELING	06
[4] PERFORMANCE ANALYSIS	10
[5] CONCLUSIONS & FUTURE WORK	13
[6] REFERENCES	14

1. INTRODUCTION

A mobile ad hoc network (MANET) is a collection of wireless nodes that can communicate in the absence of a fixed infrastructure. Nodes in a MANET can change their locations and activity status (i.e., active/inactive) and can adapt to such network topology changes to ensure data delivery to any node in the network. Application scenarios include battlefield operations, emergency disaster relief, etc.

Content-centric networking (CCN) [3], [5], [8] is a networking paradigm whose goal is to consider access and delivery of resources (e.g., content and services) as a fundamental driver for network design. CCN uses hierarchical names as addresses instead of IP-based host addresses. This fundamental paradigm

shift “decouples” resources from the host that they reside on. It enables the network to effectively locate and deliver resources requested by an endpoint.

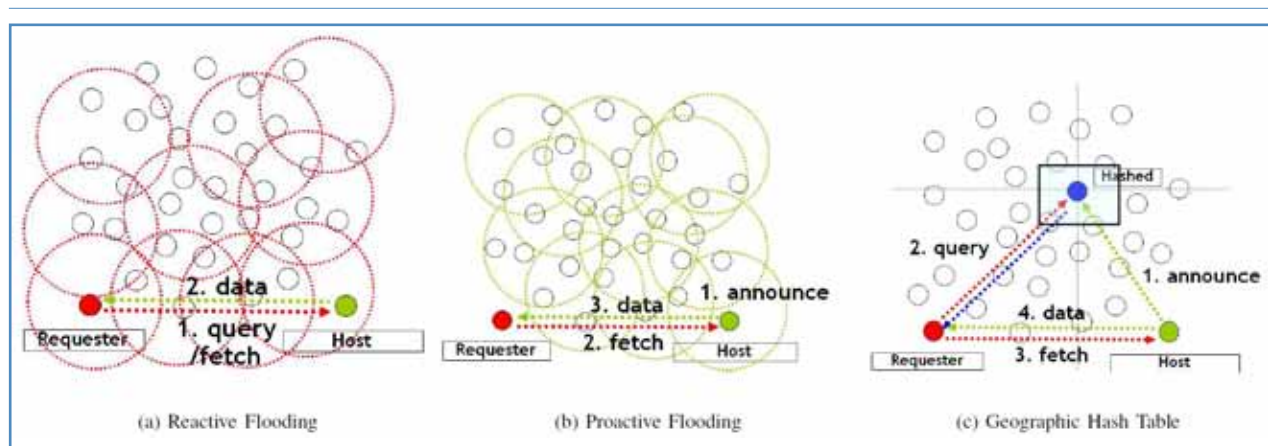
In this paper we consider CCN as a fundamental driver for MANET protocol design. We argue that the CCN paradigm is in fact well suited as a foundation for the design of a MANET. The purpose of routing protocols is to deliver “data” of interest to the target(s) in an efficient, reliable, secure way. Moreover, it is important to note that a large fraction of wireless communications in MANET scenarios is “data-centric” in nature, for instance, command and control, surveillance data, situation awareness information, and software updates.

Given this, we argue that, rather than designing complex topology-based routing protocols in order to achieve “end-to-end reachability”, an alternative “data centric” framework may be considered.

In our exploration of the protocol design space for a content-centric MANET (CCM), we find that key enablers of a CCM can arise from existing MANET protocols, including those designed for routing and resource discovery. Recent MANET research has focused on alternatives to topology-based routing (e.g., OLSR [1], AODV [12], DSR [6]). The motivation for this research is rooted in the challenges of routing in an environment of unpredictable topological changes due to node mobility and channel fading. An emerging alternative is geographic routing, which uses neighboring location information for packet forwarding (e.g., GPSR [7], GPCR [10], or geocasting [11]).

Many MANET protocols exist for “data delivery”, ranging from unicast/multicast routing, to resource discovery, to content distribution. Resource discovery is a non-trivial operation in a MANET. Resources in a MANET include nodes, content and services. Initially, researchers proposed to use a centralized directory where resource information could be stored. Flooding-based approaches were then introduced in order to solve the reliability issue of the centralized approaches. Due to scalability concerns,

Figure 1: Candidate designs for CCM



more complex designs based on structural overlays (e.g., geographic distributed hash table [13]) emerged. In this paper, we systematically evaluate the suitability and effectiveness of the existing approaches. We identify a set of fundamental approaches that represent key points in the design space. We then develop analytical models for the identified designs and evaluate their efficiency for a content-centric MANET. Our models provide insights into the strengths and weaknesses of each candidate design as well as their dependence on network parameters, e.g., node churn and loss probability. Based on our results, we derive a set of recommendations for content-centric MANETs and sketch possible designs in this space. This work is a critical step towards a full specification of a content-centric network, which is part of our future work.

In addition to providing support for resource location and access, CCN-based networks aim to effectively deliver the identified resources to the requestor. This is achieved by enabling nodes to cache content they have forwarded. Caching is feasible as content is addressed by name and not by the host it resides on. Any node that has the requested content can respond with a copy. The benefits of ubiquitous caching are in particular pronounced in MANETs in the presence of unstable and unpredictable paths. For the remainder of this paper, we focus on aspects of resource discovery and routing as it is an enabler for caching. We believe that many of today's (a) Reactive Flooding (b) Proactive Flooding (c) Geographic Hash Table caching algorithms can directly be applied to MANETs and leave a detailed evaluation of caching techniques for further study.

The major contributions of this paper are threefold. First, we define a spectrum of candidate solutions for CCN in MANET. Second, we derive analytical models for each design and compare them in terms of cost (e.g., number of transmissions), reliability and latency under different scenarios. Third, we discuss recommendations for the design of content-centric MANETs. The remainder of this paper is as follows. Section II present the spectrum of candidate solutions for the design of content-centric MANET. Section III derives analytical models for each design that Section IV compares. Finally, Section V concludes the paper by discussing recommendations for the design of content-centric MANETs.

2. DESIGN SPACE

This Section presents a selected set of candidate designs for CCM. We chose flooding as an extreme design and geographic hash table as one of the most sophisticated approaches which is also considered one of the more suitable approaches for MANETs. For each design, we assume GPSR [7] as the underlying routing protocol. GPSR is a geographic routing protocol for multi-hop wireless networks. In GPSR, each node is aware of its location (e.g., via the global position system) and of the location of the nodes in its surroundings.

Packets are marked with the positions of their destinations and are routed using the local knowledge of each node. A node in MANET can be either a content host (it has a copy of a given content item), or a content requester (it seeks for a content item). Each design leverages three major operations:

- *announce*—used by a content host to advertise content availability
- *query*—used by a content requester to locate the MANET node where the desired content is located
- *fetch*— used by a content requester to actively retrieve a content item.

In the remainder of this Section, we describe the functioning of each design. A summary and visual representation of each design can be found in Figure .

A. Flooding

The rationale of flooding-based designs is to leverage the broadcast friendly nature of the wireless medium. Flooding-based designs can be divided into two sub-classes: *reactive* and *proactive*.

a) *Reactive*: This class of designs leverages network flooding as a requester seeks for a resource item, where a resource can either be content or the location of a node in the MANET. Thus, no announce operation is required in reactive flooding.

As Figure 1(a) shows, a requester initiates a resource discovery by means of a query message that is flooded in the entire MANET. As the request reaches the resource host, data flows back to the requester by mean of a unicast operation.

b) *Proactive*: This class of design leverages periodic network flooding for the announcement of resource availability from resource hosts (Figure 1(b)). It follows that a requester is always aware of the node at which a resource is located and can directly access it by means of a unicast operation. Thus, in proactive flooding, no query operation is required, and the content retrieval is accomplished by means of a fetch operation. The fetch operation consists of a unicast message.

B. Geographic Hash Table

In the Geographic Hash Table (GHT) [13], each resource is assigned a key through a hash operation or distributed hash table [2]. A key is in the form of geographic coordinates, e.g., the result of a hash operation is a pair of coordinates in a two dimensional space. Figure 1(c) shows the set of operations used in the GHT. First, the host announces the pair $\langle \text{resource}, \text{host} \rangle$ to the K nodes whose locations are the closest to the resource key. When a requester aims to retrieve a given resource, it first computes the hash for the resource. The result of the hash operation is the pair of coordinates where the underlying GPSR protocol needs to route in order to reach one of the K nodes that holds the pair $\langle \text{resource}, \text{host} \rangle$. This operation is accomplished by the query operation by means of a unicast message. Once it has retrieved the information about the resource host, the fetch operation directly reaches the resource host by mean of a unicast message.

Table 1: List of Parameters

Name	Description
A	Availability
C	Average cost
L	Average latency of successful retrieval
h	Average hop distance between two end points
K	Retransmission limit
n	Total number of nodes
p	Average link-level loss probability
r	Radio range
T	Retransmission timeout
t_b, t_u	Average link-layer broadcast and unicast delay
$\mathcal{H}()$	Average number of hops traversed in case of loss
$\mathcal{P}()$	Multi-hop loss probability for a unicast message

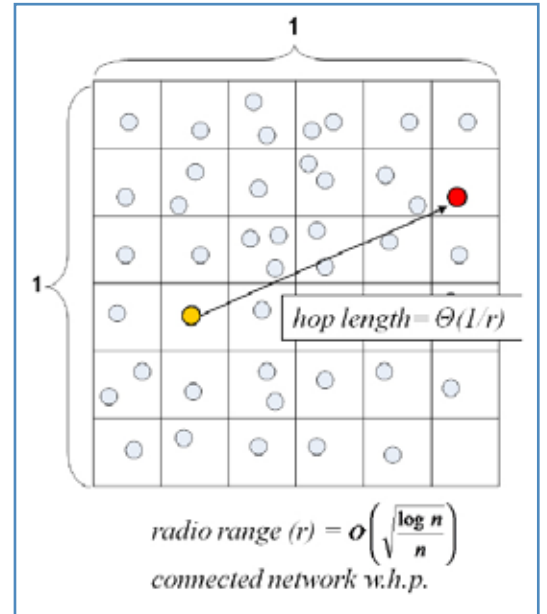
3. MODELING

In this section we perform an analytical study of the candidate schemes for resource discovery in MANETs. We start with a brief overview of the network model. Subsequently, we introduce the metrics used for comparison and derive analytical expressions for each metric and design. Table I summarizes the list of metrics, parameters and functions used in the remainder of this paper.

A. Network Model

In the network model as depicted in Figure 2 we assume that n mobile nodes are uniformly distributed in a square of unit area. Nodes have a common transmission range of $r = \Omega(\sqrt{\log n/n})$, which ensures connectivity in the network with high probability [9]. The average hop distance between two communicating nodes is then $d = O(1/r) = O(\sqrt{n / \log n})$ [4]. We introduce t_b and t_u to denote the average delays of broadcast and unicast messages between adjacent nodes, respectively. We assume that flooded messages are eventually delivered to all nodes even in the presence of packet loss; this holds when node density is high enough to ensure sufficient duplication of packets [14]. Under the same assumption the number of link-level messages of a flooding operation is $O(n)$.

Figure 2: Network Model



For unicast messages let p denote the average link-level loss rate. The loss probability for a unicast message sent over h hops can then be expressed by:

$$\mathcal{P}(h) := 1 - (1 - p)^h, \quad (1)$$

and the expected number of hops the message traverses in the case of loss follows:

$$\mathcal{H}(h) := \sum_{i=1}^d \frac{(1-p)^{i-1} p}{\mathcal{P}(h)} \cdot i \quad (2)$$

B. Evaluation Metrics

The general objective of all designs is to make resources on one node available to other nodes in the MANET. Availability manifests in success of retrieval in the event of another node querying and fetching a resource. In a lossy environment, higher availability can be achieved by increasing reliability of message delivery, which comes at the cost of increasing messaging overhead and increasing access latency. Thus, in our analysis we compare the different designs based on these metrics:

- 1) Availability A in terms of successful access.
- 2) Cost C in terms of average number of messages.
- 3) Average latency L for resource access.

The performance of a scheme depends in general on the performance and complexity of three operations it may involve: *announce*, *query* and *fetch* (cf. Section II). Each operation is accomplished through a sequence of messages; we use subscripts A ; Q ; F to refer to announce, query, and fetch operations, respectively. Note that some designs may also involve maintenance operations; we briefly discuss the *maintenance* operation when applicable.

As a consequence of the finite loss probability, we assume that any operation $OP = \{A, Q, F\}$ involving unicast implements timeouts and retransmissions to increase reliability. The number of trials until success is then governed by a geometric distribution and has an expected value of $1/(1 - P(h))$ in the unlimited case. Practical schemes, however, can be assumed to limit the number

of attempts to K_{op} before considering a request/response operation failed. Let $X_{op} = \{1,0\}$ describe the outcome of an operation as successful and failed, respectively. For an operation that is independent of the outcome of any other operation, X_{op} is distributed according to:

$$\Pr(X_{OP} = 1) = 1 - \Pr(X_{OP} = 0) = 1 - \mathcal{P}^{K_{OP}}(d), \quad (3)$$

where d is the hop distance between end points on which losses may occur.

Let f_A and f_Q denote the frequency of announce and query operations, respectively. Fetch operations are assumed to always follow a corresponding query, i.e., with the same frequency f_Q . The average cost $C_{tot}(\Delta t)$ of a design accumulated over a time period Δt is then expressed by:

$$C_{tot}(\Delta t) = f_A C_A \Delta t + f_Q (C_Q + C_F) \Delta t. \quad (4)$$

The expected latency L_{tot} of a retrieval attempt is the sum of the latencies for querying and fetching the resource:

$$L_{tot} = L_Q + L_F. \quad (5)$$

Due to the limited number of retries and infinite probability of failure of any operation involving unicast messaging, it is necessary to estimate their cost and latency over the failure and success cases. We therefore define the following two helper functions, which follow from the law of total expectations:

$$\mathcal{C}(d, k) := \mathcal{P}^k(d) k \mathcal{H}(d) + \sum_{i=1}^k \left\{ \mathcal{P}^{i-1}(d) (1 - \mathcal{P}(d)) \cdot ((i-1)\mathcal{H}(d) + d) \right\}, \quad (6)$$

$$\mathcal{L}(d, k, t) := \mathcal{P}^k(d) k t + \sum_{i=1}^k \left\{ \mathcal{P}^{i-1}(d) (1 - \mathcal{P}(d)) \cdot ((i-1)t + d\tau_u) \right\}, \quad (7)$$

where k denotes the limited of retries and t for the timeout. We next derive expressions for the availability and for the components of the cost and latency for each of the schemes.

C. Reactive Flooding

Reactive flooding schemes do not announce the hosted objects; hence, $\Pr(X_A = 1) = 1$ and $C_A = 0$. The retrieval process is a blended query-and-fetch operation, which we subscript with F : the request is flooded and eventually reaches a node hosting the desired object; the response consists of a unicast message. To reduce the impact of losses, the above request/response sequence is repeated up to K_F times with each round adding T_F of delay. Since only the unicast response can fail, the availability A is determined by:

$$A = \Pr(X_F = 1) = 1 - \mathcal{P}^{K_F}(h). \quad (8)$$

In the case of success, cost and latency can be expressed with a geometric distribution. If successful on the i th try, the number of packets sent is the sum of i flooding attempts, $(i-1)$ lost unicast responses, and one delivered unicast response. The cost of a query-and-fetch operation is given by:

$$C_F = \mathcal{P}^{K_F}(h) K_F (\mathcal{H}(h) + n) + \sum_{i=1}^{K_F} \left\{ \mathcal{P}^{i-1}(h) (1 - \mathcal{P}(h)) \cdot ((i-1)(\mathcal{H}(h) + n) + h + n) \right\} \quad (9)$$

The latency in presence of success in the i th round is the sum of $(i - 1)$ timeouts T_F and one time interval from sending the request via broadcast to receiving the response via unicast. This yields:

$$L_F = \mathcal{P}^{K_F}(h)K_FT_F + \sum_{i=1}^{K_F} \left\{ \mathcal{P}^{i-1}(h)(1 - \mathcal{P}(h)) \cdot ((i - 1)T_F + h(\tau_b + \tau_u)) \right\}. \quad (10)$$

D. Proactive Flooding

The announcement operation in this approach consists of a single flooding message, which eventually reaches all nodes. Thus, all nodes can address the source node directly and the query consists of only a local lookup, i.e., $C_Q = 0$ and $L_Q = 0$. We neglect the error introduced by node arrivals and assume that a local lookup is always successful, from which it follows that $\text{sPr}(X_A = 1) = 1$.

The retrieval process is reduced to a fetch operation consisting of unicast request/response sequences. The total success rate of a retrieval attempt is equivalent to the success probability of a fetch, and for the availability it follows:

$$A = \Pr(X_Q = 1) \Pr(X_F = 1) = 1 - \mathcal{P}^{K_F}(2h). \quad (11)$$

In case a fetch operation terminates unsuccessfully, it has triggered K_F rounds of request/response sequences each adding $H(2h)$ of messaging overhead and T_F of delay. When successful in the i th attempt, the cost sums up to $(i - 1)$ times the average number of hops during failure and an average round-trip for the successful request-response pair. The cost of a fetch operation C_F and its latency L_F , respectively, are then given by:

$$C_F = \mathcal{C}(2h, K_F), \quad (12)$$

$$L_F = \mathcal{L}(2h, K_F, T_F). \quad (13)$$

E. GHT

The GHT scheme involves all three operations—announce, query and fetch—, each of which consists of pairs of unicast request and reply messages. The retrieval process fails in case of the following three events: (1) the query operation terminates without delivery of a response; (2) the query response indicates that the announcement for the requested object was not correctly delivered; (3) the fetch operation terminates without a delivered response to the requester. Both events (1) and (2) lead to immediate abortion of the retrieval process and still suppress the fetch operation. For the availability it follows:

$$A = \Pr(X_A = 1) \Pr(X_Q = 1) \Pr(X_F = 1). \quad (14)$$

We now derive the error probabilities and average cost for all three operations, as well as the expected latency components for the retrieval.

1) *Announce*: An announce operation is successful as soon as an announce message sent over an average of h hops successfully reaches the resolver node, even if a response acknowledging the registration of the object fails. The success probability equates then to:

$$\Pr(X_A = 1) = 1 - \mathcal{P}^{K_A}(h). \quad (15)$$

The announce operation is terminated either after the reception of a response message, or after K_A unsuccessful tries. Since an announce operation is independent of any other operation, its average cost is given by:

$$C_A = \mathcal{C}(2h, K_A). \quad (16)$$

2) *Query*: A query operation is considered successful and terminated with the delivery of a response in a maximum of K_Q tries. It is independent of whether the response contains the resource information or not, so that for the success probability it follows:

$$\Pr(X_Q = 1) = 1 - \mathcal{P}^{K_Q}(2h). \quad (17)$$

The average number of query messages C_Q and the average latency L_Q follow from Eqn. 6 and Eqn. 7, respectively:

$$C_Q = \mathcal{C}(2h, K_Q), \quad (18)$$

$$L_Q = \mathcal{L}(2h, K_Q, T_Q). \quad (19)$$

3) *Fetch*: On a failure of the query operation, the fetch is suppressed and retrieval fails. Otherwise, the requester starts the fetch operation, which succeeds with the first response retrieved in a maximum of K_F tries. The probability of a success can then be calculated using:

$$\Pr(X_F = 1) = 1 - \mathcal{P}^{K_F}(2h). \quad (20)$$

For calculating the average cost of a fetch operation, it is important to consider that the execution of the fetch is conditional on the success of both the announce and query operations. Thus:

$$C_F = (1 - \mathcal{P}^{K_A}(h))(1 - \mathcal{P}^{K_Q}(2h))\mathcal{C}(2h, K_F), \quad (21)$$

$$L_F = (1 - \mathcal{P}^{K_A}(h))(1 - \mathcal{P}^{K_Q}(2h))\mathcal{L}(2h, K_F, T_F) \quad (22)$$

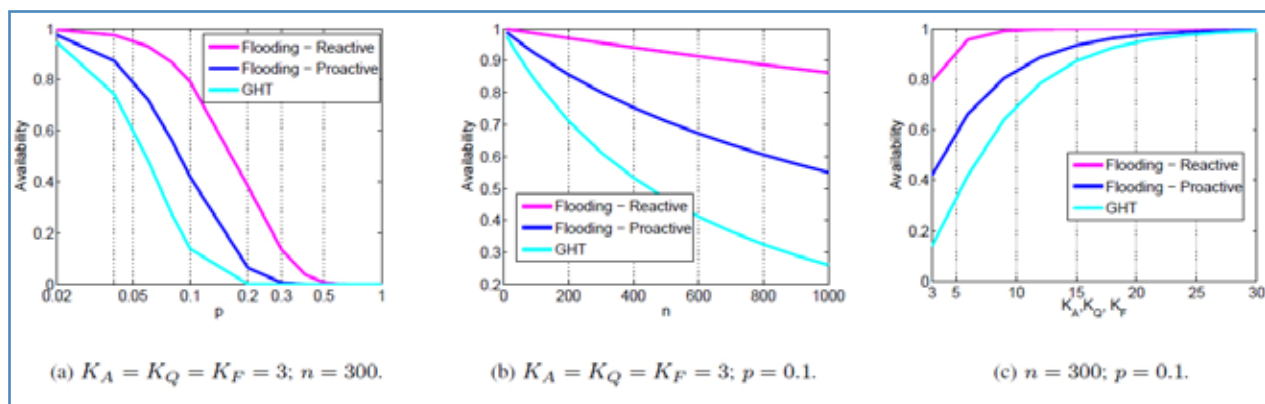
4. PERFORMANCE ANALYSIS

After having derived mathematical expressions capturing the availability, cost and latency based of each design, in this Section we evaluate them under different scenarios, e.g., small vs. large networks, reliable vs. challenging environment. We assume a IEEE 802.11 DCF MAC layer and present numerical results of all designs for the same large range of parameter settings to allow for a direct comparison.

A. Availability

We begin with an analysis of the resilience to losses. Figure 3(a) shows the availability as a function of the link-layer loss probability p . We set $n = 300$ nodes, which represents a medium to large MANET, and limit for the number of tries for all operations—announce, query, and fetch—to $K_A = K_Q = K_F = 3$. In absence of packet loss ($p = 0$), all designs are perfectly reliable and availability is $A = 1$. As p increases, the availability of each design decreases fast. Reactive flooding is the most reliable scheme with about 80% success of content retrieval attempts for $p = 0:1$, while GHT is the least reliable scheme. The high availability for reacting flooding is achieved through flooding operations, which are always successful but at the same incur a high cost (cf. Section III). Conversely, GHT leverages unicast-based messages, which have a high probability of failure in presence of larger values for p but are less expensive in terms of messaging overhead.

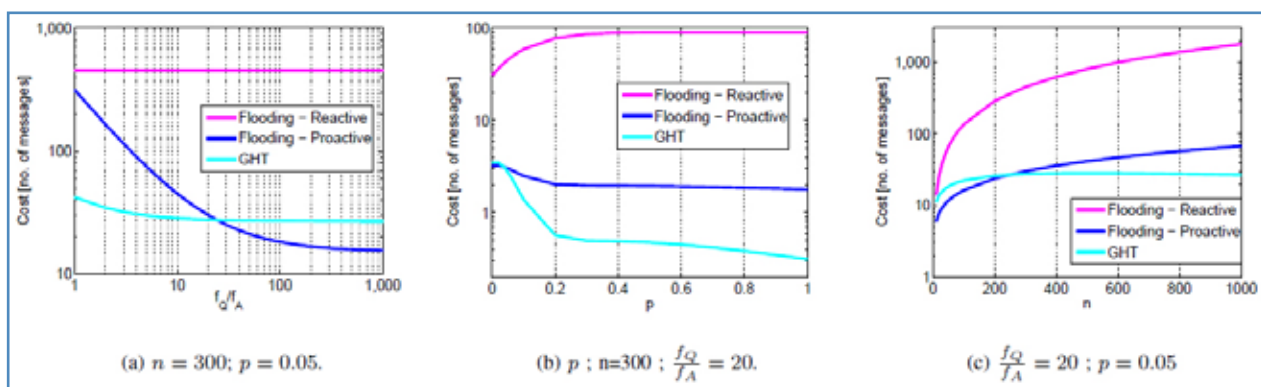
Figure 3: Availability Analysis



We now investigate the impact of the network size on availability. Figure 3(b) shows the availability as a function of the number of nodes n . We set $p = 0:05$ and $K_A = K_Q = K_F = 3$. Figure 3(b) confirms the trend observed in Figure 3(a): as the network size grows, reactive flooding provides for high availability; e.g., about 85% successes for $n = 1; 000$. In contrast, availability for proactive flooding and GHT drops below 80% as the network grows over two hundred and four hundred nodes, respectively.

In a operational MANET, loss probability p and network size n are not tunable. The only means to improve availability is by increasing $K_A, K_Q,$ and K_F , i.e., the number of retrials in case of failure for an announce, query and fetch operation, respectively. Though each of these parameters can be set independently, for simplicity we set $K_A = K_Q = K_F$. Figure 3(c) shows the evolution of the availability as a function of $K_A = K_Q = K_F$ in a medium/large MANET (i.e., $n = 300$) with reasonable loss probability (i.e, $p = 0:05$). Intuitively, as K_A, K_Q, K_F increase availability in each of the different designs approaches 1. However, while triplicating K_A, K_Q, K_F increases the availability of flooding-based designs by only 20 - 35%, it increases the availability of GHT by about 55%. This can be explained by the fact that increasing K_q, K_q, K_f reduces the gap between unicast-based operations and flooding-based operations in terms of reliability.

Figure 4: Cost Analysis ; $K_A, K_Q, K_F = 3$



B. Cost

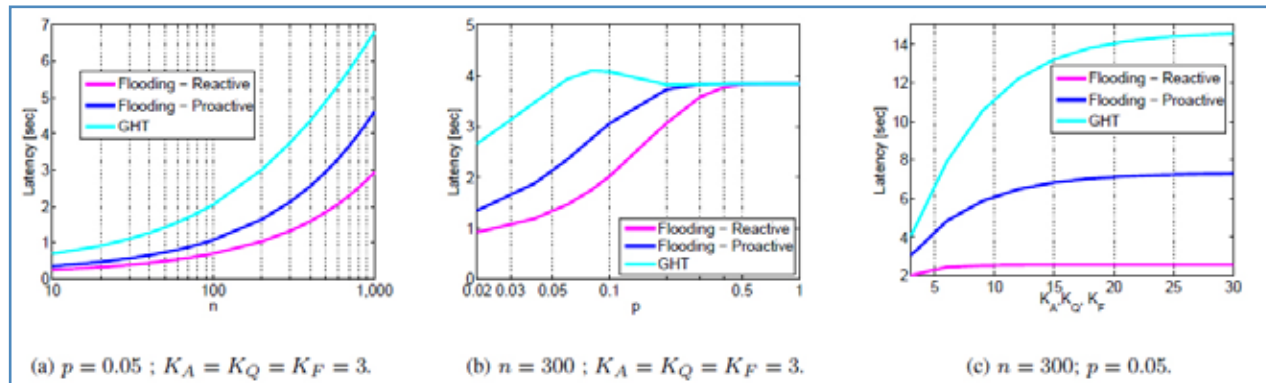
In this subsection, we compare the cost (i.e., number of messages sent over the MANET) of each design under different scenarios. This metric has high importance in particular in the case for bandwidth-limited and energy-limited devices. In the remainder of this Section, we discuss the numerical results for the cost of each content design under a large spectrum of parameters (Figure 4). Throughout the analysis, we consider $K_A = K_Q = K_F = 3$.

We start by analyzing the impact of the content popularity on the average cost per content retrieval attempt. To do so, we vary the ratio f_Q/f_A , where f_Q denotes the frequency of query operations and f_A denotes the frequency of announce operations. The ratio f_R/f_A is the number of content retrieval attempted between two consequent announce operations, thus, representing content popularity. Figure 4(a) shows the evolution of the cost of a retrieval attempt as a function of f_R/f_A for $n = 300$ and $p = 0.05$. Reactive flooding is by far the most expensive scheme independently of content popularity since each content retrieval attempt generates a flooding operation, independently of how popular the requested content item is. Thus, the cost per content retrieval attempt is constant. Conversely, in proactive flooding a flooding operation is used only for content announcement, i.e., each f_R/f_A content retrieval attempts. As f_R/f_A increases, i.e., content is requested more frequently, the cost of the flooding operation for the announcement is amortized and the overall cost decreases. A similar behavior is observable also for GHT; however, the cost decrease is less evident as the cost of an announce operation in GHT is much smaller than in proactive flooding. When the ratio f_R/f_A becomes larger than 20, proactive flooding shows a smaller cost per content retrieval attempt than GHT. This indicates that despite the high cost of flooding-based operations, they are efficient in content-centric MANETs in presence of highly popular content items.

We now focus on the impact of the loss probability p on the cost per content retrieval attempt (Figure 4(b)). We set $n = 300$ nodes and $f_R/f_A = 20$, i.e., the value derived from Figure 4(a) at which GHT and proactive flooding have a comparable cost per content retrieval attempt. We vary the value of p in the range between 0 - 0.1. We refrain from testing values of $p > 0.1$ because of their negative impact on availability, as depicted in Figure 3(b). When p tends to 0, i.e., each scheme has a high and comparable availability (cf. Figure 3(b)), proactive flooding and GHT have also comparable cost, while reactive flooding confirms its very expensive nature. When p approaches 0.1, the curves for GHT and proactive flooding depart, with GHT the cost getting close to zero. This can be explained by GHT's availability decreasing much faster than proactive flooding, which results in an apparent cost reduction.

Finally, we evaluate the impact of the network size n on the cost per content retrieval attempt (Figure 4(c)). We set $p = 0.05$ and $f_R/f_A = 20$. The network size has a bigger impact on the cost for flooding-based approaches than for unicast based approaches such as GHT. However, when the network size is smaller than 200-300 nodes proactive flooding has a smaller/comparable cost than GHT. This value of n derives from the ratio $f_R/f_A = 20$ that allows to amortize the cost of the flooding-based announcement in proactive flooding. Then, as n increases GHT shows a smaller cost compared to proactive flooding. However, the difference between the cost of the two designs is not very pronounced, e.g., for $n = 1,000$ about 65 message are sent per content retrieval attempt for proactive flooding versus 25 messages for GHT.

Figure 5: Latency Analysis



C. Latency

We now compare the *latency* (i.e., required time to accomplish a content retrieval attempt) of the designs design under different scenarios. This is important for some applications running over MANET, e.g., interactive applications that might require to locate and retrieve content quickly. In the remainder we discuss the numerical results for the latency of each content design under a large spectrum of parameters (Figure 5). Throughout the analysis, we set $T_a = T_q = T_f = 1.5 * rtt$, where the round-trip-time rtt depends on the network size and t_u , the average unicast delay between adjacent nodes. This choice is justified by the need to ensure that a message retransmission is not triggered before the possible reception of its acknowledgment message. We also set the average unicast and broadcast delay between adjacent nodes to $t_u = 50ms$ and $t_b = 30ms$, respectively [].

We first investigate the impact of the network size on the latency. Figure 5(a) shows the evolution of the latency as a function of the number of nodes n . We set $p = 0:05$ and $K_A = K_Q = K_F = 3$. When the network is small (minimum 10 nodes), all schemes show average small latency values in a similar range; e.g., 200ms for reactive flooding and 800ms for GHT. GHT has the longest latency values due to the query step: requesters first need to contact the resolver node, which knows at which node in the MANET the content is located, and only then can start the fetch operation. Flooding-based approaches have smaller latency values as they do not need this additional query step. While flooding-based approaches become very costly as the network size grows (cf. Figure 4(b)), they can deliver content with much shorter latencies than GHT as depicted in Figure 5(a). For example, even in a network composed by 1; 000 nodes, reactive flooding shows a latency of 3sec compared to 6sec calculated for GHT.

Figure 5(b) shows the impact of the loss probability p on the latency derived for each design. We consider a MANET composed by $n = 300$ nodes and $K_A = K_Q = K_F = 3$. Reactive-flooding is overall the design resulting in the smallest average latency. In fact, reducing the number of unicast operations is beneficial to increase the availability (Figure 3), which in turn reduces the chances of timeouts, and therefore long latency values. For the same reason, the proactive flooding has shorter latency values compared to GHT. As p increases, all designs converge to a similar latency value which is the product $T_{OP} * K_{OP}$, where $OP = A, Q, F$ ¹; in fact, when the loss probability is very high, all the unicast operations fail causing to retransmit up to K_{OP} times after expiration of a timeout of duration T_{OP} . If we focus on the GHT curve, we notice that the curve has a maximum in $p = 0:8$, and then it converges to $T_{OP} * K_{OP}$.

Figure 3(c) shows that increasing the values of K_A, K_Q, K_F helps to meet a MANET requirement in term of availability. However, this increase in availability comes at the price of additional latency. We now estimate the impact of K_A, K_Q, K_F on the latency (Figure 5(c)). Figure 5(c) shows that as K_A, K_Q, K_F increase, the latency curve of each design stabilize. This happens because when the availability reaches 1 for a given set of values for K_a, K_q, K_f (cf. Figure 3(c)), adding additional retries is not beneficial as they are rarely executed. Given unicast-based schemes are the ones that more can benefit from increasing K_A, K_Q, K_F , they reach very high latency values, e.g., about 14sec when $K_A = K_Q = K_F = 30$. Conversely, given reactive flooding can reach an availability of 1 already when $K_A = K_Q = K_F = 10$, the latency grows to a maximum value of 3sec.

¹This holds because we assume that $K_A = K_Q = K_F$

5. CONCLUSIONS & FUTURE WORK



This paper investigates the design space for a content-centric MANET. We design a set of analytical models from which we compare the performance of several candidate solutions, including reactive flooding, proactive flooding, and geographic hash tables (GHT). A surprising result of our analysis is the competitive performance of unstructured flooding compared to more sophisticated techniques. In fact, for a relatively small MANET (e.g., less than 200-300 nodes) the cost of maintaining routing information for unpopular content overwhelms the benefits of more complex structured solutions. In these networks it is much better to use reactive (on-demand) flooding. By contrast, structured solutions (as represented by GHT) demonstrate better performance both in term of cost (number of packets) and latency for popular content. Futhermore, GHT is optimal in the case of static networks and networks with small node churn.

The implications of these results are that, in MANET designs that take data delivery as a primary objective, such as content centric MANETs, data popularity should be a primary design criterion. The performance boundaries explored in this paper are very useful for MANET design solutions in which a designer can predict network conditions. For example, we recommend flooding for small networks with unpopular content and large host churn and recommend GHT in the absence of churn and with uniform content popularity.

Studying flooding and GHT solutions provides initial insight into the design space for a CCM. We have begun to study additional solutions, including hierarchical geo-location (HLS) and virtual ring routing (VRR). From these explorations we believe that GHT performance is representative of both HLS and VRR.

Based on the initial analyses of this paper, we can take a step back in the evolution of complex MANET designs and reconsider basic design choices. We observe that the complexity of network protocols is largely attributed to preserving end-to-end reachability; cross-layer optimization is then necessary to better utilize network resources for that purpose (e.g., mobility prediction, link level broadcasting, network coding). However, the main purpose of MANETs is to deliver data of interest to the target(s) in an efficient, reliable, secure way. This paper provides initial results in re-evaluating MANET design toward the design of a content-centric MANET.

In future work, we intend to extend our analysis to a full specification of a content-centric MANET. For example, caching and replication are important features of designing a “data centric” network that we are currently adding to our models and analysis.

REFERENCES



- [1] T. Clausen and P. Jacquet. Rfc-3626 optimized link state routing protocol (olsr), 2003.
- [2] F. Dabek, B. Zhao, P. Druschel, and I. S. Towards. A common api for structured peer-to-peer overlays. *Lecture Notes in Computer Science*, 2735:33–44, 2003.
- [3] M. Gritter and D. Cheriton. An Architecture for Content Routing Support in the Internet. In *USITS'01*, San Francisco, CA, USA, March 2001.
- [4] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):338–404, 2000.
- [5] V. Jacobson, D. K. Smetters, J. D. Thronton, M. F. Plass, N. H. Briggs, and R. L. Braynard. Network Named Content. In *CoNEXT 2009*, Rome, Italy, December 2009.
- [6] D. B. Johnson, D. A. Maltz, and Y. C. Hu. Rfc-4728 the dynamic source routing protocol for mobile ad hoc networks (dsr), 2007.
- [7] B. Karp. Gpsr: Greedy perimeter stateless routing for wireless networks. In *MobiCom00*, pages 243–254, Boston, MA, USA, Aug. 2000.
- [8] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. Kim, S. Shenker, and I. Stoica. A Data-Oriented (and Beyond) Network Architecture. In *SIGCOMM 2007*, Kyoto, Japan, August 2007.
- [9] S. R. Kulkarni and P. Viswanath. A deterministic approach to throughput scaling in wireless networks. *IEEE Transactions on Information Theory*, 50(6):1041V1049, 2004.
- [10] C. Lochert, M. Mauve, H. Fubler, and H. Hartenstein. Geographic routing in city scenarios. *Comput. Commun. Rev.*, 9:69–72, 2005.
- [11] J. Navas and T. Imielinski. Geocast - geographic addressing and routing. In *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pages 66–67, Budapest, HU, 1997.
- [12] C. Perkins, E. Royer, and S. Das. Rfc-3561 ad hoc on-demand distance vector (aodv), 2003.
- [13] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker. Ght: a geographic hash table for data-centric storage. In *WSNA'02*, pages 78–87, New York, NY, USA, 2002. ACM.
- [14] B. Williams and T. Camp. Comparison of broadcasting techniques for mobile ad hoc networks. In *MOBIHOC02*, pages 194–205, Lausanne, Switzerland, 2001.