Queuing Theoretic Model for Service Discovery in Ad-hoc Networks

Dipanjan Chakraborty, Avinash Shenoi, Yelena Yesha, Yacov Yesha, Anupam Joshi
Computer Science Department
University of Maryland Baltimore County

Mukesh Singhal
Computer Science Department
University of Kentucky

Abstract

This paper describes an analytical model for service discovery protocols in ad-hoc networks. We propose a bottom-up approach towards modeling various essential features of distributed service discovery protocols. More specifically this paper proposes a method to analytically model the service cache (SC) on an ad-hoc node. We employ the use of stochastic process models and fundamental queuing theory to model the data present in the cache. We show using simulations how our model can be used to predict the service cache usage at the node level. Our model is based on the assumption of non-preemption of service descriptions from the service cache and that the system is at steady state. We assume that there are infinite number of services in the system. We also experimentally verify that our model predicts the service system behavior reasonably well even when the assumption of infinite services does not hold.

Keywords: Service Discovery, Queueing Theory, M/G/-c/c, Ad-hoc Networks

1 INTRODUCTION AND MOTIVATION

An Ad-hoc network (also refered to as Mobile Ad-hoc Network or MANET) is a wireless network formed on-the-fly by multiple mobile nodes like laptops, PDAs etc. Applications of top of MANETS often times need to utilize resources or services that are present on other mobile nodes in its neighborhood. This requires the need for a seamless service/information discovery infrastructure. Even though there is work in developing service discovery infrastructures, there is very little work in trying to analytically model ad-hoc service discovery protocols.

Apart from solutions in the wired-network [6, 8, 9] or Internet domain, there has been work towards development of service discovery architectures for MANETS [2, 5, 10, 11, 12]. Most service discovery protocols for ad-hoc networks have certain features in common. Some of these features are *advertising of a local service, caching of the received advertisements, broadcasting (or selective forwarding) of service discovery requests.*

In this paper, we have focused on a service discovery protocol [2] that essentially has variants of all the above-mentioned features. The discovery protocol we have focused on is as described:

Each node advertise their local service(s) to the nodes in their nearby vicinity (specified by number of hops). Each node caches a received advertisement in its service cache (SC) (provided it has space available). The advertisement timeouts and is expunged from the cache after a certain timeout period. When a node wants to discover a certain service, then it first checks its SC to see if it has information about the service. The service is considered "discovered" if an entry of the service advertisement is there in the SC. The node sends a service discovery request in the network in case of a cache miss by specifying the number of hops. Every node receiving this request checks its SC and replies with the information of the service if an entry corresponding to that service is present.

In this paper, we present a queueing theoretic formalism to model the service discovery at the service node level by trying to model the ways in which a node is affacted by the discovery protocol. The service cache is the key component that influences the discovery process at the node level. This is primarily because :(1) Number of entries in a service cache (SC) determines the probability of a service request matching a given service description; (2) Number of entries in a SC offers a reasonable parameter to judge the density of services/frequency of advertisement in the environment (assuming some relaxed assumptions on the maximum memory of the node); (3) The information present in the SC is used to determine the routing policy of service requests. This paper presents a M/G/c/c queuing theory-based model to predict the behavior of the SC and hence model the service discovery at the node level for ad-hoc networks.

Our motivation behind coming up with an analytical model is driven by the following factors: (1) *Protocol Behavior Prediction and Optimizations*: We accomplish various tasks like analytically predicting memory requirements for the Service Cache, optimizing advertisement frequencies of nodes etc. Analytical prediction of probability of a service being discovered for a given service density and network bandwidth is also possible. (2) *Theoretical Performance Bounds Calculation*: It empowers us with the capability to estimate upper and lower bounds on the memory and network capacity usages for discovery protocols having variations of the features mentioned above.

To the best of our knowledge, there has been no work towards analytically modeling service discovery protocols. However, there has been considerable work in modeling services [15, 16, 17] and service interactions [20, 21] using descriptive constructs and logic. There has been some work in analyzing self-healing strategies [18] that enable service discovery systems to mantain consistency [19] during network failure. However, this work tries to use a event-based specification model to specify protocol interactions. They do not use any mathematical model to study and predict the behavior. We are not providing a separate section for related work since we have covered most of the concepts and existing theory in this section.

The rest of the paper is organized as follows: In Section 2 we present the mathematical formalism to model the service discovery starting from a single ad-hoc node. Section 3 describes our assumptions and our node-level model in greater detail. Section 4 describes our experimental evaluation of the model. We conclude in section 5.

2 NODE-LEVEL INTERACTION MODEL

Our model is developed following a bottom-up modeling paradigm where we represent the Service Cache as the *atomic unit*. It is considered the atomic unit because it is the most basic unit that participates in the various interactions. All higher level interactions between nodes (e.g. forwarding of service advertisements, matching of service discovery requests, selective forwarding of service requests) either affect or are affected by the service descriptions stored in the SC.

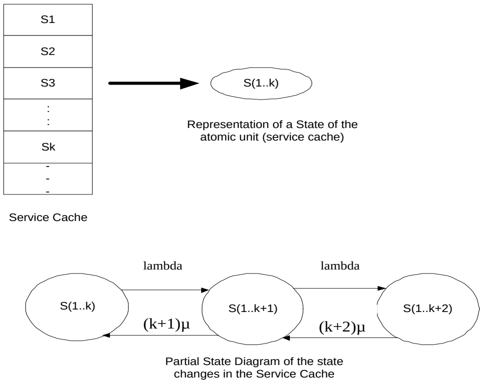

Figure 1: Service Cache in a node and corresponding markov model

State Definition:
We define the state E_n of the SC in the following manner: *The SC in a node is in state E_n at time t if and only if it contains n remote service descriptions at time t.* Thus, formally:

$$\begin{aligned} E_n(t) &= \text{service descriptions present in an SC} \\ &\text{at time t.} \\ &= n(\text{according to our definition}) \end{aligned}$$

We consider the presence of one service description as one unit in the SC. The physical memory occupied by one service advertisement may be different from another (due to their varying sizes).

The state of an SC changes when a new advertisement arrives in the system. An advertisement is considered new if it contains a new service description in it. Let c= maximum number of advertisement units that can be stored in a cache. λ= mean of the arrival process of service advertisements. Thus, when the SC is in state E_n, then it may change to state E_{n+1} with the arrival of a new advertisement with mean rate λ. Similarly, the state of a SC also changes when an advertisement times out and is expunged from the cache. Let μ= mean of the timeout values (refered as timeout distribution or service time distribution in the paper). The state of SC changes from E_{n+1} to E_n when a service description is removed from the SC. We follow a *no preemption* policy when the SC is full. Thus, a new advertisement is discarded if the SC is full.

2.1 M/G/c/c Queueing System

We use the M/G/c1/c2 queuing system [3] to model our service cache. Due to lack of space, we do not present details of the basic system. However, details can be obtained from our technical report [7]. In brief, an M/G/c1/c2 queuing system represents a system where (1) M denotes the distribution followed by the arrival process of jobs in the system. A process denoted by M indicates that the interarrival time between two jobs follow an exponential distribution. Thus the arrival process follows a poisson distribution [1]. (2) G denotes the distribution followed by the service time of jobs. A process denoted by G indicates that it can follow any distribution (poisson, gaussian etc). (3) c1 denotes the number of servers in the system (4) c2 denotes the maximum number of jobs that can be in this system at a given time *t*. For the purposes of this paper, we focus on a system where c1=c2. We explain the reason behind such model in section 3. Thus, there can be atmost c jobs (c=c1=c2) in the system.

2.1.1 Formulae

This system is modeled as a *Birth-and-Death* process with the coefficients:

$$\lambda_n = \lambda \quad n = 0, 1, 2, ..., \tag{1}$$

$$\mu_n = n\mu, \quad n = 1, 2, ...c \tag{2}$$

λ_n and μ_n denotes the arrival rate and the service time rate when the state of the system is E_n. We denote a=λ/μ. a is defined as the traffic intensity of the system. Let p_n = probability of system having n jobs
Thus p_n= probability of the system being in state E_n. It can be shown [3] that

$$p_n = \frac{\frac{a^n}{n!}}{1 + a + \frac{a^2}{2!} + \ldots + \frac{a^c}{c!}} \text{ for } n = 0, 1, ...c \tag{3}$$

Thus, we can get the probability of the system containing n jobs given that we know the arrival rae (λ) and the service time rate (μ) and the size of the system (c).

3 M/G/c/c MODEL FOR SERVICE CACHE

We model the service cache in our bottom-up approach using the M/G/c/c queuing model. We have a few simplifying assumptions for the SC.

1. **Infinite Services Assumption:** We assume that there are *infinite services* in the whole system. Thus, every service advertisement entering a service cache is considered unique.
2. **Steady State Assumption:** We assume *steady state* of the system. This paper does not deal with transient states. From the point of view of the service cache, the system is in steady state when the arrival rate and the service time rate does not vary with time. Transient states are very difficult to model and moreover, most of the systems reach a steady state after a short time.
3. **No Preemption Assumption:** We assume that an incoming advertisement is discarded if the SC is full.

We present an analogical proof showing that the various components of our service discovery protocol can be related using a one-to-one cardinality to the various components of the M/G/c/c queuing system.

1. **Service Advertisements and Jobs**: We map a service advertisement in our system to a job in M/G/c/c. However, the inter-arrival time between jobs in an M/G/c/c system follow exponential distribution. In section 4, we provide experimental verification that the inter-arrival time between advertisements follow exponential distribution.
2. **Service Cache Entries and Servers**: Each free entry unit in the service cache is considered equivalent to a free server in the M/G/c/c system. The action of storing an advertisement in one free entry in the service cache is mapped to the action of a job being processed by a server in M/G/c/c. Service time distribution in M/G/c/c is mapped to the distribution of the timeout values of the individual entries in the service cache. Note that the service time can follow any distribution (property of M/G/c/c as explained in section 2.1).
3. **Maximum Cache Size** c: The maximum size of the cache (in number of units) can be considered equivalent to the number of servers c in the queuing model.
4. **Maximum number of entries** c: From the first three equivalence relationships, we deduce that the maximum number of jobs in the queuing model is analogous to the maximum number of advertisements present in a service cache at any time *t*. The service cache behaves exactly like the M/G/c/c queuing model in this regard.

Thus, with the above analogical mappings, our service cache is in state E_n when there are n service descriptions in it. An arrival of a new advertisement is a birth (that changes the state) and the timeout of an advertisement can be considered to be a death. Given an arrival rate λ

and the timeout rate μ we can find the traffic intensity a (i.e. the ratio λ/μ) and the probability p_n that the system will contain n entries at steady state.

4 EXPERIMENTAL EVALUATION

We implemented our test bed on the well-known network simulator Glomosim [14] under various mobility conditions and different node topologies. Simulation environment consisted of node topologies ranging from a topology with 25 nodes to a topology with 100 nodes. We used a random way-point mobility pattern for all the nodes. Each node advertises its services across either one or more hops. For the purpose of the experiments, we considered services to be advertised across one hop only. Each experiment was ran for a simulation time of 1 hour over a space of (190m X 190m). We considered an uniform advertisement interval of 10s and uniform advertisement timeout of 5s. Nodes followed a random waypoint mobility pattern with 3m/s speed with a 5s stoppage time.

We carried out experiments to observe the distribution of the inter-arrival time of service advertisements. We also carried out probabilistic estimation of the service cache being in state E_n (n=0,1,2…c)(for a given λ and μ) and compared it to the values predicted by M/G/c/c model (using equation 3). Finally, we performed experiments to determine how well M/G/c/c model approximates the behavior of the service cache when the **Infinite Services Assumption** does not hold. We discuss the results in the following subsections. Our simulation takes into consideration message losses, disconnections, delays and node mobility. Thus, this by far approximates a real-life ad hoc network.

4.1 Determination of the Arrival Process of Advertisements

We observed that if the transmission of advertisements from individual nodes follow an uniform distribution (with a certain upper bound), the arrival process of advertisements follow a poisson distribution. Logically, if there are two nodes in the system, a uniform process of advertisement generation will result in an uniform arrival process (under ideal situations, i.e. no message loss, disconnections etc). However, this is clearly not the case when there are multiple nodes in the vicinity of a device and when the ideal situations do not hold true. We assumed that the transmission of service advertisements followed an uniform distribution. This is a most general assumption since it does not impose any restriction on the way in which the advertisement is generated. The inter-arrival time distribution of advertisements is plotted in figure 2. We carried out the above experiment for nodes ranging from 25 to 100 with different mobility patterns and different advertisement diameters. We carried out least square approximation on the experiment data and observed that the curve is very near to exponential. Thus, we conclude that for an uniform advertisement transmission distribution, the advertisement arrival process follows a poisson distribution for a real-life ad hoc network environment. This is an important result in the field of ad-hoc service discovery advertising. This result further substantiates the assumption about the distribution followed by the arrival process of jobs in a M/G/c/c queuing system.

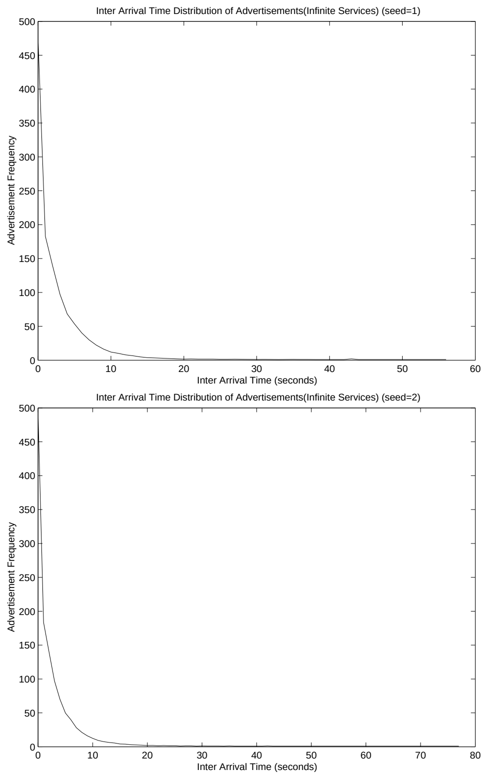

Figure 2: Distribution of Inter-arrival times of service advertisements for Uniform advertisement transmission time distribution

4.2 Comparison of Cache Usage Probability

For a given λ and μ, equation 3 provides a probabilistic measure of the system being in state E_n. Cache Usage is defined as the actual number of cache entries being used. We calculate the probabilistic distribution of the service cache being in different states (state being given

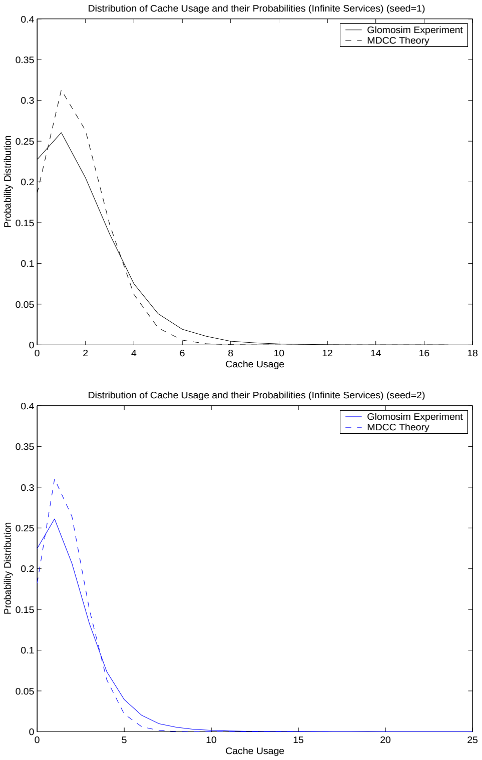

Figure 3: Cache Usage Probabilistic Measure compared against prediction made using M/D/c/c model

by the cache having certain entries) for a given λ and μ. We calculate λ from the arrival process distribution. We assume a constant service time distribution for the purpose of the experiments. Thus in our experiments the general distribution (G) is replaced by the deterministic distribution (D). We carried out these experiments assuming that all the incoming advertisements were unique. Thus we respected the **Infinite Services Assumption** in these experiments. We compared the probabilistic measures as obtained from experiments against the ones predicted by M/G/c/c theory. The results are shown in figure 3.

We observe that the probability estimates follow each other very closely. In general, the M/D/c/c theory underestimates the probability right at the beginning, overestimates in the middle and underestimates at the end with marginal error bounds. We observe that the mean error ranges from 6% to 13%. Thus, we can conclude that M/G/c/c queuing model can be used to model the service cache of nodes in our bottom-up model towards service discovery. However, the predictions made by this model are subject to marginal errors but overall obtains good results in modeling the behavior of the service cache.

4.3 Comparison of Cache Usage Probability without the *Infinite Services Assumption*

The **Infinite Services Assumption** does not hold for practical ad-hoc networks. This is because there are finite number of nodes in an ad-hoc environment and it is infeasible to think of infinite services being present on a finite number of nodes in a real-life environment. The results of the M/G/c/c model are only valid when each job is unique. We carried out experiments to determine how well the model estimates the behavior of the service cache in presence of a finite number of services. Two advertisements are duplicate if they contain the same service descriptions. We observed that the arrival process of advertisements follow a poisson process anyway. Our experiments showed that M/G/c/c is still a good model to apply even when the environment contains finite services. We carried out experiments with a service/node ratio ranging from 1 to 0.125. Figure 4 displays the results. Thus, we can apply the model towards modeling service discovery even when there are finite number of services in the environment. However, the service/node ratio is important to determine how well the model will represent the actual working of the system. We have shown that the model is appropriate even with a service/node ratio of 0.125 which is a reasonable value for a real-life ad hoc network.

5 CONCLUSIONS

In conclusions, we have analogically proved that M/G/c/c model can be used to predict the cache usage of a service cache (SC) in a mobile device participating in ad-hoc service discovery. We observe that arrival rate of advertisements (given uniform sending rate) follows exponential distribution very closely in an ad hoc network. The cache usage can be predicted reasonably well using M/G/c/c queuing model. We have experimentally showed that even for finite services M/G/c/c provides a good approximation of the predicted usage. This would enable practical ad-hoc environments to apply our theory as a good "approximation model".

Our bottom-up queueing theoretic model has got far-fetched applications in modeling service discovery protocols. We can determine the average timeout of a service description for a given the arrival rate, the size of the cache and some threshold probability of the required number of entries in the cache. We can also determine the optimal timeout rate for the system for a given advertisement arrival rate so as to achieve equilibrium. Equilibrium is defined as that state when the average number of entries in the SC do not fluctuate.

In future work, we aim to model higher level interactions between these Service Cache units and eventually model protocol level components of ad-hoc service discovery applications.

References

[1] A. Populis. *Probability, Random Variables and Stochastic Processes.*

[2] Dipanjan Chakraborty, Anupam Joshi. *GSD: A Novel Group-based Service Discovery Protocol for MANETS.* IEEE MWCN. Stockholm, Sweden. 2002.

[3] Arnold O. Allen. *Probability, Statistics, and Queuing Theory with Computer Science Applications, Second Edition.*

[4] Donald Gross and Carl M. Harris. *Fundamentals of Queuing Theory, Second Edition.*

[5] Dipanjan Chakraborty and Anupam Joshi. *Dynamic Service Composition: State-of-the-Art and Research Directions.* Technical Report TR-CS-01-19. University of Maryland, Baltimore County. 2001.

[6] Salutation Architecture Specification (Part 1), version 2.1 edition. `http://www.salutation.org`. 1999.

[7] Dipanjan Chakraborty and Avinash Shenoi and Anupam Joshi and Yelena Yesha. *A Queueing Theoretic Model for Service Discovery in Ad hoc Networks.* Technical Report. TR-CS-03-24. CSEE. University of Maryland, Baltimore County.

[8] Rekesh John. UPnP, *Jini and Salutation: look at some popular coordination framework for future network devices.* California Software Labs. 1999.

[9] Ken Arnold and Ann Wollrath and Bryan O'Sullivan and Robert Scheifler and Jim Waldo. *The Jini Specification.* Addison-Wesley. 1999.

[10] Arturo Crespo and Hector Garcia-Molina. *Routing Indices for Peer-to-Peer Systems.* ICDCS. 2002.

[11] Sumi Helal and Nitin Desai and Choonhwa Lee. *Konark- A Service Discovery and Delivery Protocol for Ad-hoc Networks.* WCNC. 2003.

[12] R.H Katz et.al. *The Ninja Architecture for Robust Internet-scale Systems and Services.* Special Issue of Computer Networks on Pervasive Computing.

[13] S. Ni, Y. Tseng, Y. Chen and J. Sheu. *A Mobility-transparent Deterministic Broadcast Mechanism for Ad-hoc Networks.* IEEE Transactions on Networking. 1999.

[14] Xiang Zeng, Rajive Bagrodia, Mario Gerla. *GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks.* Proc. 12th Workshop on Parallel and Distributed Simulations. 1998.

[15] DARPA Agent Markup Language. `http://www.daml.org`.

[16] Dipanjan Chakraborty and Filip Perich and Sasikanth Avancha and Anupam Joshi. *DReggie: A smart Service Discovery Technique for E-Commerce Applications.* 20th Symposium on Reliable Distributed Systems. October. 2001.

[17] Web Services Description Language 1.1. `http://www.w3.org/TR/wsdl12`

[18] C. Dabrowski and K. Mills. *Understanding Self-healing in Service Discovery Systems.* Proceedings of the First ACM SigSoft Workshop on Self-healing Systems (WOSS '02), November 18-19, 2002, Charleston, South Carolina, ACM Press, pp. 15-20.

[19] C. Dabrowski and K. Mills and J. Elder. *Understanding Consistency Maintenance in Service Discovery Architectures in Response to Message Loss.* Proceedings of the 4th International Workshop on Active Middleware Services, IEEE Computer Society, July 2002, pp. 51-60.

[20] Web Services Flow Language. `http://xml.coverpages.org/wsfl.html`.

[21] Business Process Execution Language. `http://xml.coverpages.org/bpel4ws.html`.

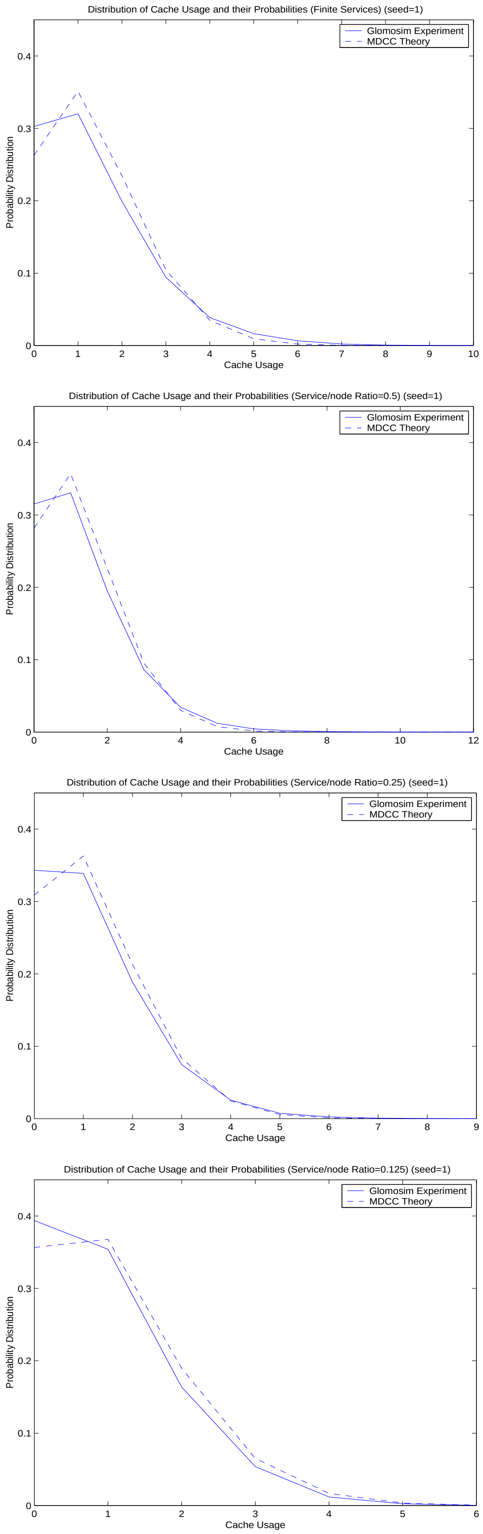

Figure 4: Cache Usage Probabilistic Measure compared against prediction made using M/D/c/c model for finite services. Service/node ratio ranges from 1 to 0.125