Correction

Nodes do not have any idea about the network. The queries are spreading around the whole network, up to d hops from the originator. There is no way to stop forwarding queries, even though the query node has already received an answer. The cost for querying is counted as the cost for sending queries to the network, which is:

$$cost = \lambda(header_{MAC} + header_{IP} + header_{UDP} + header_{Q} + c)\left(1 + n\pi(d \cdot r)^2\right) \quad (15)$$

4 Experimental Results

The model described above has been implemented in Matlab 6.5. Using the model, four sets of experiments are done. The following tables and figures show the results of the experiments. In all the experiments below, we assume the update frequency $\mu = 0.1$ and each context information type can be represented in 32 bits, i.e., $c = 32$ bits. The sizes of headers are assumed as follows [15]: $header_{MAC}$ = 160 bits; $header_{IPv6}$ = 320 bits; $header_{UDP}$ = 64 bits; $header_{AD}$ = 32 bits [14]; $header_{Q}$ = 192 bits [14].

There are 4 sets of experiments done in this section. Experiment 1 is used to achieve the optimal cost by choosing the proper width, w, of the Bloom filter and the number of hash functions, b, with given depth, d, of the filter, query rate, λ, and number of services, s, per node. Experiment 2 shows the influence of the query rate, λ, on the network cost for given d and s. The influence of the query range, d, is evaluated in Experiment 3. In the final experiment, we show the impact of density of services, s, in the network.

4.1 Experiment 1

In this experiment, we assume query rate $\lambda = 0.1$, $s = 1$. For each given value of depth of filter, the experiment result shows there exists a certain value of w and b which leads to the minimum network cost. The result is shown in Table 2. It is also compared with the complete and non advertisement under similar situations.

As we see from Table 2, for each depth of the filter, the proper width and number of hash functions leads to a minimum network cost which is much lower than for the cases of a complete advertisement and a non-advertisement. The difference becomes larger as query range d increases. The final column shows the maximum number of services that are covered by one Bloom filter based on the related size of Bloom filter.

Table 2. Optimal BF cost for certain depth d compared with complete and non advertisment

d	w (bit)	b	BF cost (bit/s)	Complete Advertisement (bit/s)	Non Advertisement (bit/s)	Maximum number of services in BF
3	128	5	99	547	1459	18
5	256	5	204	2006	3917	50
7	512	6	452	4438	7603	98
10	1024	6	1199	9910	15437	200

Correction

4.2 Experiment 2

Using Bloom filters, we can reduce the packet size by using simple and efficient coding. However, false positives also create redundant traffic. We expect that there exists a point at which the traffic generated due to false positives is much more than the benefit of using Bloom filters. In contrast, if there are only few queries in the network, it does not pay to broadcast the context information to the entire network. A non-advertisement protocol can perform better in this case. This experiment is going to discuss the suitable range of using Bloom filters for context discovery to achieve the minimum network cost.

We set μ as a reference, and change the value of λ. Here we talk about λ/μ. The experiments show that the suitable range of λ/μ decreases when the depth of the filter, d, increases. When each node has only one service (s = 1), the Bloom filter context discovery algorithm performs better than the non-advertisement algorithm when λ is at least 0.1 times μ. The Bloom filter algorithm performs better than the complete advertisement algorithm even if λ is 10^8 times μ. Fig. 3a shows the situation when $d = 5$.

When each node has 4 services (s = 4), the network requires larger Bloom filters to contain more information. The results show that for d = 3, the proper range of λ/μ is (0.1, 1000); for d = 10, the proper range of λ/μ is (01, 100). Fig. 3b shows the result when $d = 5$.

We found that in practical situations the Bloom filter algorithm has a better performance. Therefore, it is a promising algorithm for mobile ad hoc networks. Note that the axes in Fig. 3 are represented in log scale.

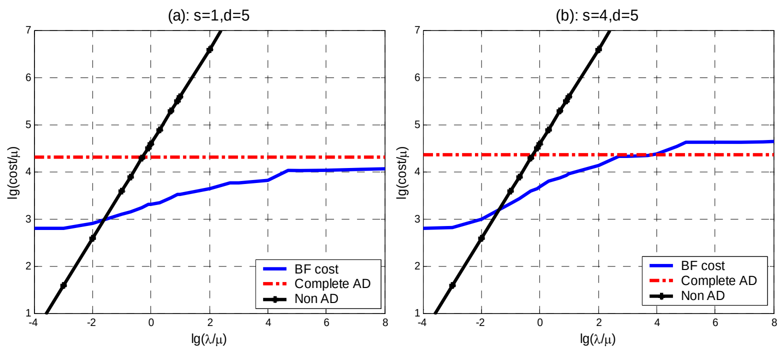

Fig. 3. Performance results for λ/μ when s = 1 (a) and s = 4 (b)

4.3 Experiment 3

With a larger search range (larger d), there are more context information types available within the range. On the other hand, a larger d also leads to larger Bloom filter. In this set of experiments, we would like to see the impact of d.

We set the depth of the Bloom filter, d, from 3 to 10, and compare the performance with different values of s and λ (fixed μ = 0.1). The results show that, in general, the

Correction

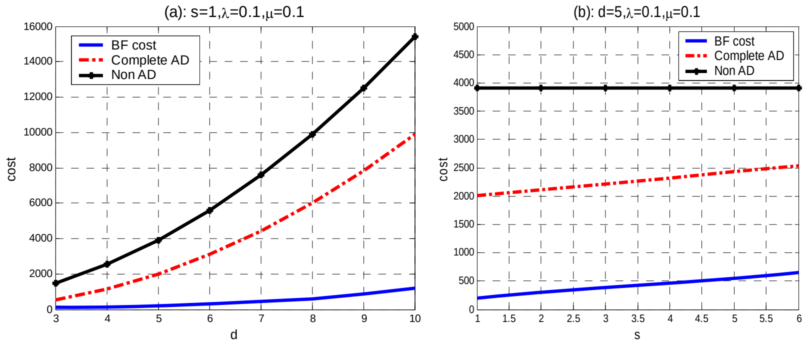

Fig. 4. when $\lambda = 0.1$: (a) impact from change of d ($s = 1$); (b) impact from change of s ($d = 5$)

Bloom filter algorithm has better performance than complete and non-advertisement algorithms. There is a limit to the number of services and the query rate for which the algorithm has the best performance. When exceeding that limit, the performance of Bloom filter becomes worse. When the number of services within the range (this depends on both d and s) and the query rate is quite high, the cost of using the Bloom filter algorithm increases significantly. For instance, this happens when $s = 4$, $\lambda = 20$, and $d > 9$. Fig. 4a shows the curve when $s = 1$, $\lambda = 0.1$.

4.4 Experiment 4

From the experiment above, we find that the number of services per node also has some influence on the network cost. In this set of experiments, we would like to investigate it in detail. We do this for fixed d and λ. The results show that s has some influence from s, but not much. When s increases from 1 to 6, i.e., the number of context sources increases from 0.0007 to 0.0042 per m^2, a Bloom filter still gives the best result among three alternative algorithms. The network cost of using a Bloom filter increases only a little bit faster for than the complete advertisement algorithm. We can expect the Bloom filter algorithm to perform worse when s is really large, which will seldom happen in reality (for given d and λ). Fig. 4b shows the curve for three alternative algorithms when $d = 5$, $\lambda = 0.1$.

5 Conclusions and Further Work

The use of attenuated Bloom filters for advertising available context types in ad-hoc networks is very promising. Results obtained from the model presented in this paper reveal the combined cost of advertising and doing unsuccessful queries due to false positives. There exists a proper size of Bloom filters to achieve optimal network cost. The performance of Bloom filters also highly depends on the ratio of query and advertisement rates, and query range of nodes. Density of network context information sources also has some influences. For a fully distributed ad hoc network in practical